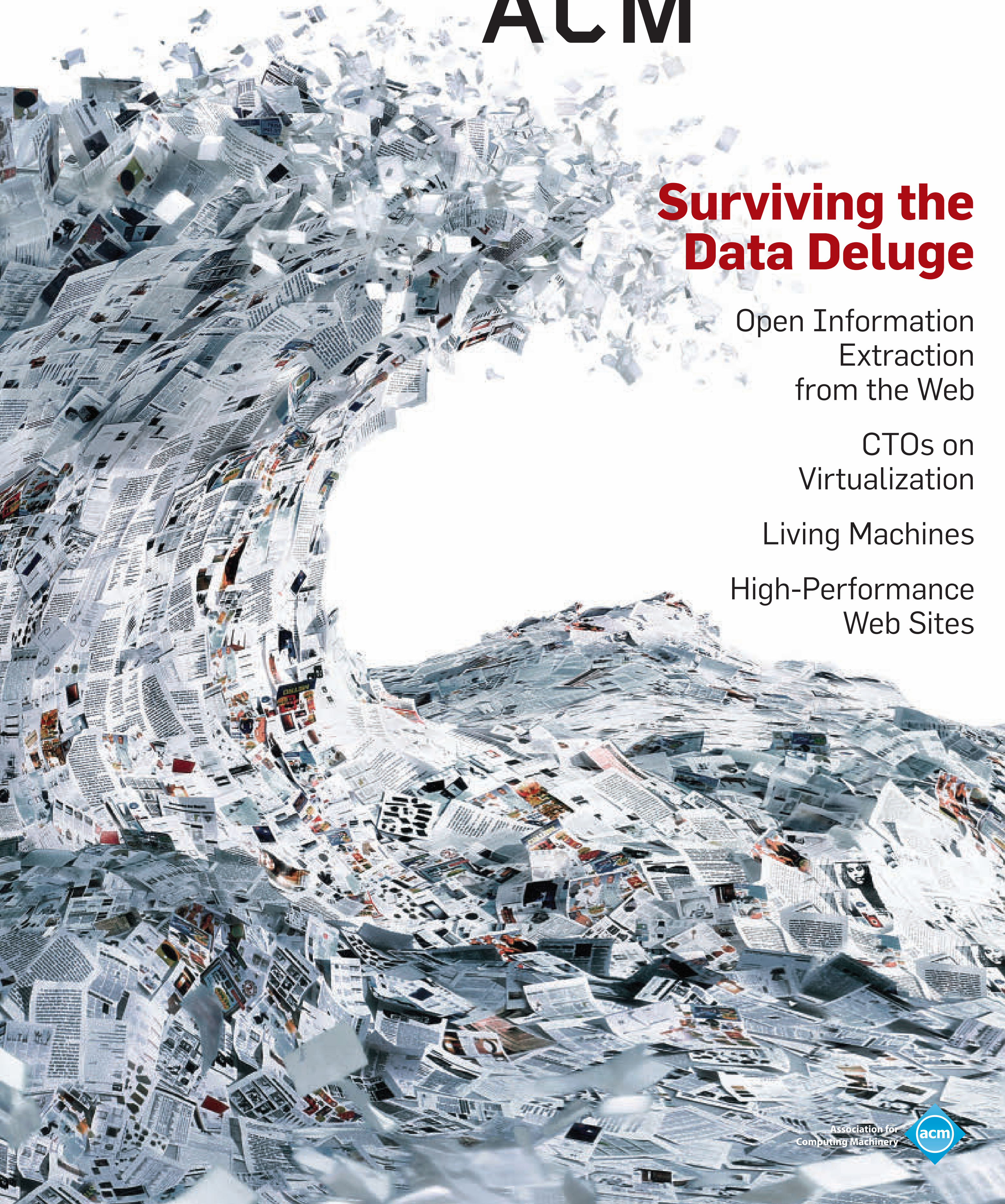
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

12/08 VOL.51 NO.12



Surviving the Data Deluge

Open Information Extraction from the Web

CTOs on Virtualization

Living Machines

High-Performance Web Sites

Association for Computing Machinery

ACM, Uniting the World's Computing Professionals, Researchers, Educators, and Students

Dear Colleague,

At a time when computing is at the center of the growing demand for technology jobs worldwide, ACM is continuing its work on initiatives to help computing professionals stay competitive in the global community. ACM's increasing involvement in initiatives aimed at ensuring the health of the computing discipline and profession serve to help ACM reach its full potential as a global and diverse society which continues to serve new and unique opportunities for its members.

As part of ACM's overall mission to advance computing as a science and a profession, our invaluable member benefits are designed to help you achieve success by providing you with the resources you need to advance your career and stay at the forefront of the latest technologies.

MEMBER BENEFITS INCLUDE:

- Access to ACM's **Career & Job Center** offering a host of exclusive career-enhancing benefits
- **Free e-mentoring services** provided by MentorNet®
- **Full access to over 3,000 online courses** from SkillSoft®
- **Full access to 600 online books** from Safari® Books Online, featuring leading publishers, including O'Reilly (Professional Members only)
- **Full access to 500 online books** from Books24x7®
- A subscription to ACM's flagship monthly magazine, ***Communications of the ACM***
- Full member access to the new ***ACM Queue*** website featuring blogs, online discussions and debates, plus video and audio content
- The option to subscribe to the full **ACM Digital Library**
- The **Guide to Computing Literature**, with over one million searchable bibliographic citations
- The option to connect with the **best thinkers in computing** by joining **34 Special Interest Groups** or **hundreds of local chapters**
- **ACM's 40+ journals and magazines** at special member-only rates
- ***TechNews***, ACM's tri-weekly email digest delivering stories on the latest IT news
- ***CareerNews***, ACM's bi-monthly email digest providing career-related topics
- ***MemberNet***, ACM's e-newsletter, covering ACM people and activities
- **Email forwarding service & filtering service**, providing members with a free acm.org email address and **Postini** spam filtering
- And much, much more

ACM's worldwide network of over 92,000 members range from students to seasoned professionals and includes many of the leaders in the field. ACM members get access to this network and the advantages that come from their expertise to keep you at the forefront of the technology world.

Please take a moment to consider the value of an ACM membership for your career and your future in the dynamic computing profession.

Sincerely,

Wendy Hall

President
Association for Computing Machinery

Association for Computing Machinery

Advancing Computing as a Science & Profession

Association for Computing Machinery

Advancing Computing as a Science & Profession

membership application & *digital library* order form

Priority Code: ACACM28

You can join ACM in several easy ways:

Online	Phone	Fax
http://www.acm.org/join	*+1-800-342-6626 (US & Canada)* *+1-212-626-0500 (Global)*	*+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

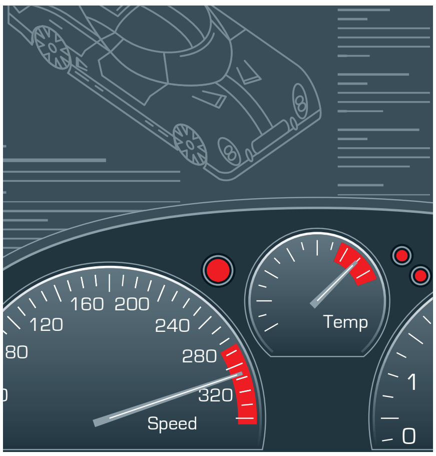
Signature

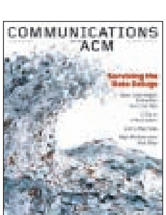
COMMUNICATIONS OF THE ACM

PHOTOGRAPH BY SACHI GAHAN

12/08
VOL. 51 NO. 12

Practice





About the Cover: To create an image of a wave of data, German artist Thomas Herbrich photographed over 1,000 printed pages then reduced them to 6cm through multiple photocopying. He then began the arduous task of adhering thousands of these miniature pages onto corrugated Styrofoam sheets to create and photograph this 3D scene. The effect of flying papers at the top of the wave was created by throwing hundreds of mini-pages out of a bucket and photographing them. He then blended the separate images together to form the wave. For more about his work, see http://www.herbrich.com/.

ILLUSTRATION BY NIK SCHULZ

Contributed Articles

Review Articles

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.

Association for Computing Machinery
Advancing Computing as a Science & Profession

COMMUNICATIONS OF THE ACM

A monthly publication of ACM Media

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.

ACM U.S. Public Policy Office
Cameron Wilson, Director
1100 Seventeenth St., NW, Suite 507
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
Chris Stephenson
Executive Director
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (800) 401-1799; F (541) 687-1840

Association for Computing Machinery (ACM)
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (212) 869-7440; F (212) 869-0481

CONTACT POINTS

Copyright permission
permissions@cacm.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmcoa@cacm.acm.org
Letters to the Editor
letters@cacm.acm.org

WEB SITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/guidelines

ADVERTISING

ACM ADVERTISING DEPARTMENT
2 Penn Plaza, Suite 701, New York, NY 10121-0701
T (212) 869-7440
F (212) 869-0481

Director of Media Sales
Jonathan M. Just
jonathan.just@acm.org

Media Kit acmmediasales@acm.org

BPA Audit Pending

Subscriptions
Annual subscription cost is included in the society member dues of $99.00 (for students, cost is included in $42.00 dues); the nonmember annual subscription rate is $100.00.

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current Advertising Rates can be found by visiting http://cacm.acm.org/advertising or by contacting ACM Media Sales at (212) 626-0654.

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

COMMUNICATIONS OF THE ACM

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

POSTMASTER

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

Association for Computing Machinery

Printed in the U.S.A.

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/1409360.1409361 Ronald F. Boisvert and Holly Rushmeier

ACM Publications: Access and Sustainability

The dissemination of research results and other technical information is one of the primary ways in which ACM carries out its mission as an educational and scientific society. As such, ACM seeks to make its publications accessible to as wide an audience as possible. However, in doing so ACM also has a responsibility to its members, as well as to the profession at large, to ensure that its publications program is sustainable far into the future. Balancing these sometimes competing goals has always been a challenge. It hasn't gotten easier in the digital age.

The centerpiece of the ACM publications program is the ACM Digital Library (DL) and its associated *Guide to Computing Literature*. Through these outlets, ACM provides free access to the metadata of over 240,000 articles that ACM publishes, as well as to more than one million bibliographic citations to the computing literature at large. The metadata ACM provides is extremely rich, containing forward citations, clickable references, unique author bibliographic pages, along with citation and download counts.

ACM also has a responsibility to its members, as well as to the profession at large, to ensure that its publications program is sustainable far into the future.

ACM supports the desires of individual authors to make their work universally available by posting the accepted (uncopyedited) versions of their articles on their own Web site or institutional site. Thanks to the very effective indexing provided by Web search engines, this, in effect, provides significant free and open access to the latest results of computing research.

A large modern professional publishing program does incur expenses, however. For example, ACM provides access to professionally developed tools for tracking submissions and reviews, paid administrative assistants for editors, professional copyediting of journal articles, as well as development and oversight of the publishing program itself, including investigation and redress of charges of plagiarism.

ACM's commitment to sustained, long-term access to its publications rests with its DL. Support of this archival library, including clean metadata, search mechanisms, high-performance data server development and maintenance, as well as preparation for migration to future data formats, also incurs significant cost.

To help recover these costs, access to the full text of the definitive versions of articles published in the ACM DL does require a subscription. As a non-profit, ACM is committed to hold the costs to the community at a very low level. This is accomplished by close attention to providing highly cost-effective internal operations, which are supported by the volunteer effort of ACM reviewers and editors.

ACM has thus far elected to use subscriptions rather than an "author pays" model to support the publishing program. In a discipline where much research is not supported by large grants and is not performed at large institutions, requiring authors to bear the expense of the publications program would shut out important segments of the community and severely impair research progress.

ACM's low prices for access to its DL have enabled wide availability (currently over 2,500 libraries worldwide). In addition, ACM facilitates access to its literature around the world through a pricing structure that is adjusted to individual's status (such as, professional vs. student) and geographic location.

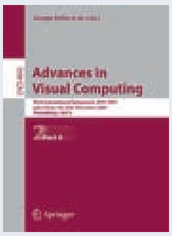
ACM continues its commitment to keeping the barriers to access of ACM's research publications as low as possible subject to its responsibility to sustain long-term accessibility and growth. We will continue to track new publishing models and developments in our quest to provide the best service to ACM members and the community at large.

Ronald F. Boisvert
Holly Rushmeier
CO-CHAIRS, ACM PUBLICATIONS BOARD

Springer
the language of science

springer.com

New and Noteworthy



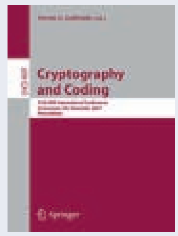
Advances in Visual Computing

Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings, Part II

G. Bebis, University of Nevada, Reno, NV, USA

The 77 revised full papers and 42 poster papers presented together with 32 full and 5 poster papers of 6 special tracks were carefully reviewed and selected from more than 270 submissions. The papers cover the four main areas of visual computing: vision, graphics, visualization, and virtual reality.

2007. XXXIII, 827 p. With online files/update. (Lecture Notes in Computer Science, Volume 4842)
Softcover
ISBN 978-3-540-76855-5 ► **$119.00**



Cryptography and Coding

11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings

S. Galbraith, Royal Holloway University of London, UK (Ed.)

The 22 revised full papers presented together with 2 invited contributions were carefully reviewed and selected from 48 submissions. The papers are organized in topical sections on signatures, boolean functions, block cipher cryptanalysis, side channels, linear complexity, public key encryption, curves, and RSA implementation.

2007. XI, 423 p. With online files/update. (Lecture Notes in Computer Science, Volume 4887)
Softcover
ISBN 978-3-540-77271-2 ► **$79.95**

Distributed Computing and Networking

9th International Conference, ICDCN 2008, Kolkata, India, January 5-8, 2008, Proceedings

S. Rao, International Institute of Information Technology, Bangalore, India

The 30 revised full papers and 27 revised short papers presented together with 3 keynote talks and 1 invited lecture were carefully reviewed and selected from 185 submissions. The papers are organized in topical sections on agreement protocols.

2008. XVIII, 588 p. With online files/update. (Lecture Notes in Computer Science, Volume 4904)
Softcover
ISBN 978-3-540-77443-3 ► **$99.00**

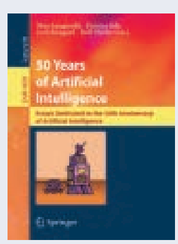
Formal Methods and Software Engineering

9th International Conference on Formal Engineering Methods, ICFEM 2007, Boca Raton, Florida, USA, November 14-15, 2007, Proceedings

M. Butler, University of Southampton, UK

The 19 revised full papers together with 2 invited talks presented were carefully reviewed and selected from 38 submissions. The papers address all current issues in formal methods and their applications in software engineering. They are organized in topical sections on security and knowledge, embedded systems, testing, automated analysis, hardware and concurrency.

2007. VIII, 387 p. With online files/update. (Lecture Notes in Computer Science, Volume 4789)
Softcover
ISBN 978-3-540-76648-3 ► **$74.95**



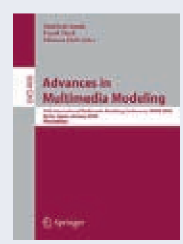
50 Years of Artificial Intelligence

Essays Dedicated to the 50th Anniversary of Artificial Intelligence

M. Lungarella, University of Zurich, Switzerland

The selected papers reflect the breadth of the topics presented and discussed at the summit, covering subjects ranging from the history and prospects of AI, to speech recognition and processing, linguistics, bionics, and consciousness. The papers are organized in topical sections on Historical and Philosophical Issue.

2007. X, 399 p. With online files/update. (Lecture Notes in Computer Science, Volume 4850)
Softcover
ISBN 978-3-540-77295-8 ► **$79.95**



Advances in Multimedia Modeling

14th International Multimedia Modeling Conference, MMM 2008, Kyoto, Japan, January 9-11, 2008, Proceedings

S. Satoh, National Institute of Informatics, Tokyo, Japan

This book constitutes the refereed proceedings of the 14th International Multimedia Modeling Conference, MMM 2007, held in Kyoto, Japan, in January 2007. The 23 revised full papers and 24 revised poster papers were carefully reviewed and selected from 133 submissions. The papers are organized in topical sections on media understanding, creative media, visual content representation, video codecs, media retrieval, as well as audio and music.

2008. XIX, 510 p. With online files/update. (Lecture Notes in Computer Science, Volume 4903)
Softcover
ISBN 978-3-540-77407-5 ► **$89.95**

Easy Ways to Order for the Americas ► **Write:** Springer Order Department, PO Box 2485, Secaucus, NJ 07096-2485, USA ► **Call: (toll free)** 1-800-SPRINGER ► **Fax:** 1-201-348-4505 ► **Email:** orders-ny@springer.com or **for outside the Americas** ► **Write:** Springer Distribution Center GmbH, Haberstrasse 7, 69126 Heidelberg, Germany ► **Call:** +49 (0) 6221-345-4301 ► **Fax :** +49 (0) 6221-345-4229 ► **Email:** SDC-bookorder@springer.com
► Prices are subject to change without notice. All prices are net prices.

013841x

DOI:10.1145/1409360.1409362 Scott E. Delman

Looking Back and Forward

2008 has been a year of incredible change for *Communications of the ACM*. Starting with the 50th anniversary issue in January in which it was first announced that the magazine would undergo a major revitalization to the realization of this goal with the debut of the July issue with a dramatically expanded editorial scope and completely new look. Behind the scenes there has been a team of the most dedicated and talented staff and volunteers any publisher could ever hope to work with and a renewed sense of optimism about what is possible for the Association's flagship publication.

Well within our sights now is a magazine that can truly be considered the most important scholarly publication in the field of computing that is fast attracting the interest of practitioners and industry technologists as well as researchers and educators, and a source of high-quality in-depth news for the entire computing and IT industries. Now more than ever in the recent past, *Communications* is attracting readers new to the magazine and longtime readers whose interest had lapsed but have now returned and are enjoying each new issue.

Well within our sights now is a magazine that can truly be considered the most important scholarly publication in the field of computing.

Getting to this point has not been easy. The past year has been at times tumultuous and filled with uncertainty, as roles and responsibilities have changed, new members have been added to the team, and several have left for other pursuits. Change is never easy and it takes a special organization to manage it well, starting with a clear vision for what needs to be accomplished and a drive to keep working toward that vision. The publication that exists today is not perfect and our job is not done, but we are on the right path and what we hear on a daily basis from you, our readership, helps to keep us focused and moving in the right direction.

The next year will be filled with even more challenges and opportunities. For the first time, ACM will launch a true destination Web site for its flagship publication. This site will among many other features include daily news for the computing community, Web 2.0 functionality such as the ability to comment on all articles published in the magazine, smart technology that will serve up related content based on user preferences, and an improved user experience for job seekers and employers looking to connect with each other. The site will serve as an important complement to the print magazine, but will in many ways exceed the magazine's print limitations. Ultimately, it is our goal to provide you with a complete experience that serves your information needs and gives you even more value for your hard-earned membership dollars than ever before.

Thank you for all of the feedback you have given in 2008 to make *Communications of the ACM* what it is today and please continue to offer your ideas and contributions to help make the magazine even better in 2009.

Scott E. Delman, GROUP PUBLISHER

DOI:10.1145/1409360.1409363

Apple Builds Great Platforms, Too, Not Just Products

WHY IS MICHAEL Cusumano puzzled about Apple? The main point in his Viewpoint column "Technology Strategy and Management" (Sept. 2008) was that you can either build products or you can build platforms. He then claimed that Microsoft is successful because it builds platforms and said he was puzzled by Apple's success since it creates only products that, no matter how great, are still not the basis of a computing platform. Isn't Unix, on which OS X is based, a platform? OS X is the user-interface platform on top of Unix, and Cocoa is the object-oriented development platform.

Cusumano might say they are not industrywide platforms, but that is rather the fault of an industry that fawns too much on incumbent monopolies. Indeed, Apple has done well to survive in such an environment.

The old VHS vs. Beta chestnut is irrelevant. A Google search finds the real reasons for Beta's failure: not technically superior to VHS in all respects and certainly not, for consumers, in the most important one—tape play time. The VHS vs. Beta war involved hardware formats; we deal in software, a much different world.

Cusumano then said that Apple copied its GUI from Xerox and by implication Microsoft then did the same to Apple. However, Xerox rather invited Apple to create such systems. Microsoft took some of Apple's code without Apple's permission, resulting in a legal case that was finally settled in 1997.

The Viewpoint's pull quote hailed that "Despite faster recent growth than Microsoft, Apple relies too much on the fleeting nature of 'hit' products," completely ignoring the design and technical excellence of Apple's products and platforms.

Apple indeed has an excellent platform—OS X, Cocoa, and Unix—that is the basis of Macintosh, the iPhone, and other products. Apple developed its own business model, while Microsoft relied on the old IBM business model of "Someone else has a good idea, copy it, and crush them." Cusumano ignored this aspect of Microsoft's "success."

Ian Joyner, Sydney, Australia

Author's Response:

The subject of Apple vs. Microsoft always stimulates emotional and even religious-like responses from the faithful, sometime impolite diatribes as well. I simply have been looking at what these companies have done for the past 20+ years and invest no emotion in them or their products. In my interpretation, Apple is primarily a product company and Microsoft primarily a platform company.

Michael Cusumano, Cambridge, MA

Should Manufacturers Fear RFID Tags?

I see few of the advantages cited by Brian L. Dos Santos and Lars S. Smith in their article "RFID in the Supply Chain: Panacea or Pandora's Box?" (Oct. 2008, Virtual Extension), which claimed that wide deployment of RFID could increase espionage among companies and limit the technology's potential as a competitive edge because they could all read the content stored in one another's RFID tags.

The article further said that new laws might be required to protect the investment companies make in their supply chains. But laws introduced too early that fail to fully consider the interests of all stakeholders, or of society in general, could stifle research, in the same way the U.S. Digital Millennium Copyright Act of 1998 made research in encryption and security more difficult.

Though I agree that rewritable tags that store a product's historical data might pose some risk to user companies, any company's ability to read RFID tags might also be improved, ultimately representing the killer app of RFID tags.

If consumers were able to check tags and see where, say, a particular item was manufactured, the result could be a more informed choice among RFID-using manufacturers and their products by, say, making it easier to avoid sweatshops or factories that do not comply with environmental regulations; embedded tags that store product histories might also allow for easier recycling or disposal. Rewritable tags could further help consumers check when an item was manufactured, helping them avoid items marketed as the latest and greatest but that actually spent months on a shelf in storage, even as suppliers were counting on consumer ignorance.

Empowerment is great for consumers but would require more honesty from manufacturers and suppliers, so some companies might resist. This is where I would want to see new laws regarding the types of data that could be stored on RFID chips.

Tomasz Rybak, Bialystok, Poland

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:10.1145/1409360.1409364 David Roman

Previewing the New CACM Web Site

Searching and Browsing

Readers use *Communications'* content—both print and digital—in different ways. Some take their time and leisurely read through articles from top to bottom, while others take a hurried approach, often in search of their personal buzzwords. Studies show that 79% of Web site users skip past carefully constructed prose to scan pages for a specific word, phrase, or image (Jakob Nielsen, *Alertbox*, Oct. 1, 1997). If they don't quickly find what they are looking for, they will do a formal search. These habits have significantly impacted the design of the soon-to-be-launched *Communications* Web site, where we plan to facilitate both types of usage patterns. For those interested primarily in browsing the site for content relevant to their specific technical interests, we are building a robust "Browse by Subject" capability that will enable the site to serve up content on specific topics in computing. For those interested in quickly identifying specific objects, both regular and advanced search options will be available utilizing the Endeca and Google search engines.

"Browse by Subject" navigation was refined by early usability tests of *Communications*' site. The tests showed some purposeful readers jumping past headlines to find specific articles. "Browse by Subject" topics do the jumping for them. They give readers a way to figuratively bypass rows of bookshelves to land squarely in front of titles devoted to software, or education, and so forth. (The placeholder topics shown on this screenshot will give way to topics defined through the collaborative effort of IT practitioners, academics, *CACM* staff, and volunteers.) They provide a way to locate articles before turning to traditional search.

The topics will also help scan recent ACM Digital Library articles, as shown in the purple-bannered Portal widget shown here. "Browse by Subject" navigation will be so simple in practice as to require little or no explanation.

ACM Member News

SOUVAINE APPOINTED TO NATIONAL SCIENCE BOARD

Diane L. Souvaine, chair of the department of computer science at Tufts University, has been appointed to the National Science Board. Souvaine was nominated by President George W. Bush in September and confirmed by the U.S. Senate in early October. As one of eight board newcomers, Souvaine will serve a six-year term.

The 24-member National Science Board acts as an independent advisor to the president and Congress on national policy issues involving science and engineering research and education. The board is also the oversight and policy-making agency for the National Science Foundation.

Souvaine's specialty is computational geometry, which focuses on the design and analysis of algorithms for solving geometric problems.

SIGMM AWARD

Ralf Steinmetz, head of the Multimedia Communications Lab at the Department of Electrical Engineering and Information Technology at Technical University Darmstadt in Germany, is the 2008 winner of the SIGMM Award for Outstanding Technical Contributions to Multimedia Computing, Communications, and Applications. Steinmetz was recognized for "pioneering work in multimedia communications and the fundamentals of multimedia synchronization."

ACM GERMAN CHAPTER

The German Chapter of ACM celebrated its 40th anniversary with a symposium on "IT In Abundance" at the IBM Lab in Boeblingen. The chapter includes more than 400 members and has local groups, which meet monthly, in 35 locations across Germany. The chapter wants to better communicate with other ACM chapters, especially European ones, and increase its membership. For more information, visit www.informatik.org or contact Gerhard Schimpf at gerhard.schimpf@smfteam.de.

news

Science | DOI:10.1145/1409360.1409365 Kirk L. Kroeker

Living Machines

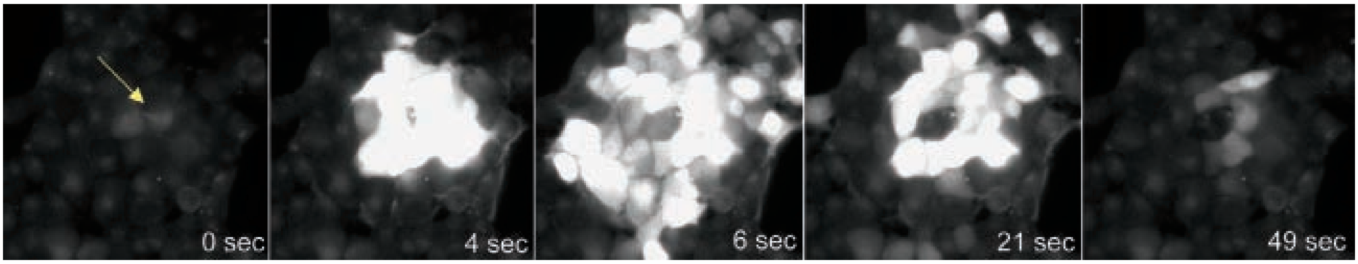
Researchers of molecular computing and communication are focusing on the type of breakthroughs needed to make the vision of ultrasmall, biocompatible computers a reality.

PHYSICISTS HAVE LONG postulated the idea that machines would become so sophisticated one day that scientists would be able to build increasingly smaller and more sophisticated devices until, at an advanced stage, entire computational systems would be able to operate inside the boundaries of a device no larger than a single cell. One early example of this type of speculation was a landmark 1959 lecture titled "Plenty of Room at the Bottom." In the lecture, delivered at the California Institute of Technology (Caltech), Nobel laureate Richard Feynman talked about engineering circuits at the molecular level, with the idea being to build a tiny set of tools that would be able to build an even smaller set of tools, and so on, until scientists reach the point at which they can create circuits consisting of a mere seven atoms.

Feynman's lecture has been credited many times for inspiring researchers working in nanotech and quantum computing. "The principles of physics, as far as I can see, do not speak against the possibility of maneuvering things atom by atom," said Feynman. "It is not an attempt to violate any laws; it is something, in principle, that can be done; but in practice, it has not been done because we are too big." Science hasn't yet realized Feynman's vision of an atomic- or even a molecular-scale computer, but it has been steadily moving in that direction for the last 50 years. Much research has focused on moving beyond the speed limitations of traditional semiconductors with quantum computing, using bulky machines that rely on atoms themselves as bits and bytes, but another branch of research, molecular computing and communication, has focused on the type of breakthroughs needed to make the fantastic vision of ultrasmall computers a reality.

Researchers working in molecular computing and communication—the inspiration for which can be traced, in part, to John von Neumann's theory of cellular automata and Alan Turing's work in autonomous self-structuring—seek to provide fundamentally new methods of solving challenging computational problems at microscale sizes. Currently, nanomachines created from biological materials are capable only of simple functions, such as detecting molecules, performing chemical reactions under certain conditions, and generating motion. While simple, these functions translate into sensing, logic, and actuation, respectively, each of which is a key element in any computing or communication system. But as with any advanced science, several major challenges in molecular computing must be overcome for the technology to make its way from lab to industry.

One of the challenges facing researchers working in this area, which requires advanced expertise in multiple disciplines, is to develop new languages



In an experiment on cell-to-cell communication conducted by Tadashi Nakano and colleagues at Caltech, a mechanically induced calcium wave propagates through several cells. The networked cells, behaving much like nodes on a LAN, propagate signals in all directions.

and methodologies so members of a team understand each other. "What a protocol means to computer scientists is communication procedures, while it means to biologists experimental procedures," says Tadashi Nakano, a professor of computer science at the University of California, Irvine (UCI). "This is a trivial example, but in order to communicate with biologists, I first needed to learn a set of vocabularies that they use in daily conversation; and I needed to explain computer science vocabularies to them." Nakano's research in the Molecular Communication Group at UCI in-volves cell biology, nanotechnology, and communications engineering, with the ultimate goal of the work being to integrate these disciplines and establish molecular communication as a science.

Currently, Nakano and colleagues are focused on engineering cell-to-cell communication through calcium signaling. The photograph on page 11 shows a series of images captured as a mechanically induced calcium wave propagates through several cells. To monitor the waves, Nakano's team loaded cells with calcium-sensitive fluorescent dyes, then used a micropipette to mechanically stimulate a cell and a fluorescence microscope to capture the images. Nakano says the project, which is in an early stage, is focused on designing the key components, such as amplifiers and switches, that are necessary to build a cell-based network. To realize some of the promises of molecular communication, says Nakano, the field's understanding of cellular communication must be expanded and engineering techniques for modifying cell functions must be advanced. "Our current attempt is like designing systems using black boxes or components whose behavior is not completely predictable," he says. "Our hope here is being able to reveal what's inside the black boxes—that is, answering unknown questions in cell biology."

DNA Walkers

Most molecular communication projects—such as Nakano's work and projects under way at other research labs around the world—share a focus on sender nanomachines, receiver nanomachines, carrier molecules, and the environment in which these tiny objects

A DNA walker created by Caltech chemists Jong-Shik Shin and Niles A. Pierce. The vertical strands form the walker's body and legs, which walks on the horizontal track.

operate. Senders and receivers include biological and biologically derived nanomachines that are capable of emitting and capturing carrier molecules, such as proteins, ions, or even DNA. Several research teams, for example, have built DNA walkers that operate much like kinesins, which are motor proteins that use ATP hydrolysis to move along microtubules. The goal of these projects is to construct a synthetic transport device that mimics the linear movement of motor proteins and can be used not only to carry a signal but also to carry out nanoscale computations.

Niles Pierce, a Caltech professor of applied and computational mathematics and bioengineering, and his colleagues have created a walker that can move along a DNA track. The first DNA walker that Pierce created several years ago was not autonomous—it required external control over the fuel strands—but his colleagues and him have built on the initial work and have created a microscale system that powers itself. "The key innovation in moving from nonautonomous to autonomous motors (both powered by the formation of DNA base pairs) was to develop conformation-changing molecules that could bind to one molecule and then change structure to deliver energy to the system," says Pierce.

Currently, Pierce and his colleagues are working on the algorithms needed to create what he calls "a compiler for molecular computing" that will take as input a high-level abstraction of the desired function for a molecular system and produce as output molecular sequences that can be synthesized to execute the function in a test tube or cell. Pierce is also working to develop both nanomechanical instruments that use molecules to detect and regulate signals in living cells and nanomechanical drugs designed to kill diseased cells while leaving healthy cells untouched. "It is a huge scientific challenge to model the cellular environment in which our synthetic molecular systems must operate in living systems," he says. "It remains to be seen how significant the resulting uncertainties are in thwarting our engineering efforts."

At present, molecular programming is a research topic, and Pierce says science is far from creating general solutions to these design challenges. Pierce estimates that it will take a minimum of three to five years to achieve practical nanomechanical instrumentation and 10 to 15 years before there are nanomechanical drugs. "But there are already some systems," he says, citing Caltech researcher Paul Rothemund's "DNA origami" method for constructing shapes and patterns, "where high school students can program molecules using a simple CAD interface to specify nanoscale details of the self-assembling structures." Other notable research includes John Reif's work at Duke University, which is focused on self-assembling nanostructures; Tom Knight's work at MIT, which is oriented toward standardizing DNA components for synthetic biology; and the work of Caltech's Erik

An example of Caltech researcher Paul Rothemund's "DNA origami" method for constructing shapes and patterns.

IMAGES BY NILES A. PIERCE (TOP) AND PAUL ROTHEMUND, CALIFORNIA INSTITUTE OF TECHNOLOGY

Winfree and New York University's Ned Seeman, which is focused on DNA computing and nanotech.

While these DNA projects have received a great deal of attention in scientific journals and even in the mainstream press, other, less well-known approaches to molecular computing might lead to entirely new computing paradigms, say researchers. One example is the work of Andrew Adamatzky, professor of unconventional computing in the department of computer science at the University of the West of England. Adamatzky's research focuses on reaction-diffusion computing, in particular on a chemical reaction called the Belousov-Zhabotinsky (BZ) reaction, which causes waves of ions to propagate through an environment. By controlling the BZ propagation pattern, Adamatzky has shown it is possible to produce biological logic gates.

Given the proper environment, the BZ reaction can operate much like a parallel processor in which each point on the wave front, mapped to a particular grid, can serve as a point of calculation. To create a Boolean logic gate, for example, Adamatzky represents True and False by the presence or absence of a wave fragment. When two or more wave fragments collide, they fuse, dissipate, generate new wave fragments, or change their trajectory or velocity, representing the Boolean variables and implementing the computation. The trajectories of the traveling wave fragments can be changed dynamically, and adjusted and tuned by colliding other wave fragments against them, making for complex interactions. "The medium can implement such sophisticated tasks as computation of the shortest collision-free path, approximation of Voronoi diagrams of arbitrary geometrical objects, and development of a skeleton of a planar shape," he says. "We proved the computational universality of the BZ medium by constructing a set of functionally complete logical gates in laboratory experiments. Our results indicate that the BZ system is a general-purpose parallel computer."

But as with other work in molecular computing, several problems must be addressed in reaction-diffusion computing. Sensitivity of the BZ reaction is one of the major issues facing Adamatzky. "We design logical circuits according to principles of collision-based computing, where information is represented by traveling wave fragments," he says. "Unfortunately, these localized excitations are unstable; they collapse or expand after some period of time." But even with unsolved problems and unanswered questions, the experiments are proving to be useful for other fields. The core ideas of reaction-diffusion computing already have spread to those working in massively parallel computing; and even some conventional silicon processors execute wave-based algorithms.

Future applications include environmentally friendly systems that can automatically decompose and drug delivery systems that can be embedded in human bodies.

These projects represent a small cross section of the ongoing developments in molecular computing. Unlike quantum-computing projects that require sizeable machinery and clean-room-style environments, the promise of molecular computing is that it can operate in natural environments, without electrical power. Whether scientists are able to achieve Feynman's vision of nanoscale computers consisting of circuits and switches made of a handful of atoms remains to be seen, but researchers today are hopeful that paradigms developed through work in molecular computing will lead to entirely new applications, such as environmentally friendly systems that automatically decompose, energy-efficient machines that generate minimal amounts of heat, and biocompatible communication systems that can be embedded in human bodies.

Based in Los Angeles, **Kirk L. Kroeker** is a freelance editor and writer specializing in science and technology. Tatsuya Suda, National Science Foundation, contributed to the development of this article.

Medicine

Detecting Breast Cancer

A large-scale British study has found that one radiologist aided by a computer is as accurate as two radiologists when it comes to detecting breast cancer in a mammogram, according to an article published online in the *New England Journal of Medicine*.

Mammograms are used to screen women for early signs of breast cancer, but the tests are not perfect. In the U.S. most mammograms are read by a single radiologist, while in Britain, they are read by two radiologists or technicians. The British researchers, led by radiologist Fiona J. Gilbert of the University of Aberdeen, analyzed the results from a randomized study of 31,000 British women. It found that a radiologist aided by a computer detected 198 cancers out of 227, while a pair of radiologists detected 199 cancers.

Computer-aided detection (CAD) systems use computer algorithms to analyze digital mammogram images and to pick out and mark suspicious areas. CAD systems are used in approximately 25% of mammogram readings in the U.S., and this percentage should increase as more medical centers switch from film x-rays to digital images.

"In the United States, it's just not practical in most practices to do double readings by physicians," said Carol H. Lee, a radiologist at Memorial Sloan-Kettering Cancer Center in New York and head of the American College of Radiology's Breast Imaging Commission. "These results are reassuring to me that a single reading with CAD can achieve that same sensitivity."

Double reading is the standard practice in at least 12 European countries. "Up to now double reading has been the gold standard of mammography," said Dr. Marco Rosselli del Turco, president of the European Society of Breast Cancer Specialists. "We have been waiting for a well-designed, prospective, randomized study to establish the role of CAD. This study provides a definitive answer about the value of adding CAD to single reading, and is likely to lead to a change in European guidelines."

Technology | DOI:10.1145/1409360.1409366

Ted Selker

Touching the Future

In combination with finger and hand gestures, multitouch input is enabling users to manipulate and display information in innovative and unprecedented ways.

The iPhone looks, feels, and acts like no other phone, thanks to its device-sized screen and multitouch responsiveness to one and two-fingered gestures. Sweep a finger to the right to unlock its touch screen. Another finger sweep scrolls through all of its features, from applications to photos. Spread two fingers apart or draw them together on a list, photo, or newspaper article to zoom in or out. Press on the screen for a few seconds and the application icons start shaking as though they have become unglued, allowing a user to rearrange them. The judicious combination of simple finger gestures and screen animations, representing physical metaphors, make the iPhone's interface feel like a physical object that one can intuitively manipulate.

Jeff Han, right, employs a simple, two-fingered gesture to manipulate an image on one of Perceptive Pixel's wall-size multitouch monitors.

With the iPhone, Apple successfully brought together decades of research and dreams. However, one could imagine the iPhone, upon its introduction, faring poorly relative to the Nokia N95 smartphone. The iPhone features an inferior camera, a slower processor, a worse keyboard, no Flash player, and not even a way to attach a lanyard. Yet, the iPhone has been a huge hit. It boasts simple integrated function that introduced millions of consumers to a gestural, multitouch interface. Unlike previous technology deployments, the iPhone did not ask consumers what it can be useful for, but presents a suite of scenarios that a person typically uses several times a day, and its designers carefully matched use scenarios to user actions. So, while one must wrestle with different ways to use the myriad of functions on the Nokia N95, the iPhone user enlarges a roadmap by simply spreading two fingers apart.

Buoyed by the iPhone's success, companies are competing to show compelling multitouch scenarios that enable users to input, process, and display information in innovative ways, often involving finger and hand gestures. Jeff Han's company, Perceptive Pixel, has created a seven-and-a-half foot diagonal multitouch monitor, most prominently seen on CNN and known as its "Magic Wall." (Perceptive Pixel has also sold its multitouch monitors to a number of unspecified U.S. government agencies.) Han's wall-size monitors use a variation of the typical camera and projector mounted behind the display surface, with the camera mapping "frustrated" light in the projection surface to detect touch inputs.

In Perceptive Pixel's videos of its wall-size multitouch monitors, Han and others use finger and hand gestures to type on a virtual keyboard, call up a document, and scroll through its content; change the composition of a human face; and manipulate 3D objects by turning them any way a user desires. In one sequence, a user pulls up a distant image from Google Earth and zooms in on the image until it's revealed to be a street-level view of midtown Manhattan, which the user pans and tilts; the user can then transform the Google Earth image into a computerized scale model of buildings and other street-level features.

Microsoft Surface uses five cameras and projector mounted beneath a table display surface, aided by a Windows Vista PC, to produce up to 52 points of touch on its 30-inch diagonal tabletop screen. As well as recognizing gestural input, Surface recognizes objects with RFID tags placed on them, allowing a user to transfer music files between an MP3 player and a smartphone by placing each device on the computer's tabletop surface. Surface uses the presence of the object and the RFID to identify the type of device and its capabilities. The music files appear on the tabletop surface near each device, enabling a user to drag material from one device to the other.

The UnMouse Pad is a multitouch pad, similar to a mouse pad, created by Ken Perlin, a computer scientist at New York University, and several colleagues. One can use any object, such

PHOTOGRAPH BY SACHI GAHAN

as a small block of wood, like a mouse on the UnMouse Pad, but more interestingly one can use multiple fingers to write and draw on the pad, creating the corresponding content on a computer screen.

An undergraduate student of Perlin's, for example, has used the UnMouse Pad for animation; one of the student's hands moved the "paper" around on a screen while his other hand drew on it. "You rethink how you interact with information," says Perlin. "It's much more human friendly."

The UnMouse Pad is part of Perlin's focus on creating innovative but low-cost technology. "Rather than use a large number of wires and expensive circuitry, we use a sparse set of force-sensing wires on the surface—one wire every quarter of an inch—and that's sufficient," he explains. "Our approach allows us to measure continuous position in the spaces between the wires, using simple and low-cost electronics."

Fortunately for Perlin and other researchers, there is now a large arsenal of relatively inexpensive motion-sensing equipment for gestural and multitouch input.

The most ubiquitous touch screens, such as ATMs and airline kiosks, measure a finger's position as it presses a layer of transparent indium tin oxide that is charged from the horizontal edges of the display to a layer that is charged from the vertical edges. Except for light pens that use the onscreen image to find where the tethered stylus is, most touch sensing today requires calibration. One can imagine a calibration-free technology that could be integrated into flat-panel displays. The approach could use the thin film circuitry in the display as an array of antennas. Modified driver chips could be used to measure the changes to the electrical field in patterns at different pixel locations in the display. And software could sort the changes in a high-frequency electrical environment of the local pixel circuits to locate where and how many fingers are touching the screen.

Commercial Success

The iPhone's commercial success—approximately 10 million units sold and counting—has been a giant step forward for the mainstreaming of multitouch technology. "The iPhone is important," Perlin notes, "because it tells people that using multiple fingers to interact with computers via hand gestures is a natural, wonderful thing."

Apple, Microsoft, and Perceptive Pixel, among others, are investigating how to best use multiple fingers and hands for multitouch input.

Bill Buxton, a senior researcher for Microsoft Research and a pioneer in human-computer interaction and computer graphics, envisions a future of multitouch devices with their own specially designed operating systems and applications. "One solution I see," Buxton says, "is that we will start building new classes of computational devices that are not constrained by the legacy applications that were designed for a very different style of interaction."

Buxton believes future technology will create new relationships between typical consumer devices and multitouch screens. "What is really fascinating to me is when you combine the ability of not just the sense of touch of my fingers, but when different objects—a phone or a camera—makes a relationship with the use of my hands and gestures," he says. "This will lead to a convergence of multitouch surfaces and what is known as tangible computing."

New multitouch surfaces also mean new finger and hand gestures, and along with the development of multitouch operating systems and applications, Apple, Microsoft, and Perceptive Pixel are investigating how to best use multiple fingers and hands for multitouch input. "A lot of our research is coming up with gestures and manipulation metaphors," according to Han, such as how a CAD designer could manipulate multiple parts of an

Computer Security

Attacking Keyboards

Swiss security researchers have developed four attacks that can detect what a person is typing on a keyboard by analyzing the signals produced by keystrokes. According to the researchers, doctoral students Marti Vuagnoux and Sylvian Pasini of the Security and Cryptography Laboratory at the Swiss Ecole Polytechnique Fédérale de Lausanne, keyboards are "not safe to transmit sensitive information."

Vuagnoux and Pasini tested 11 keyboard models connected to a computer via a PS/2 or USB slot and found that each of the keyboards was vulnerable to at least one of the four attacks they devised. (The attacks also worked with laptop keyboards.) The researchers used a radio antenna to "fully or partially recover keystrokes" by spotting the electromagnetic radiation emitted when the keys were pressed.

Vuagnoux and Pasini have stated they believe "our attacks can be significantly improved, since we used relatively inexpensive [equipment]."

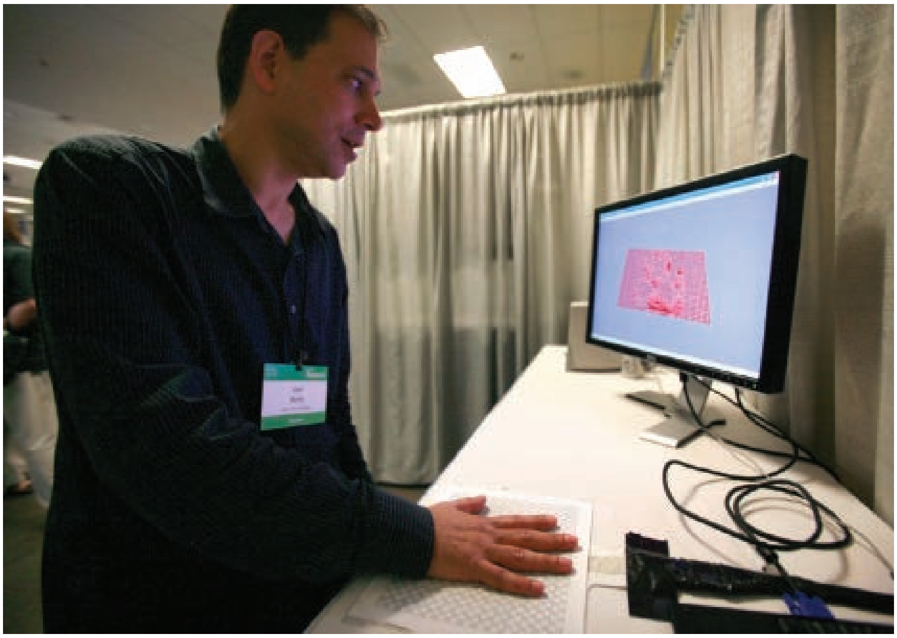
Artificial Intelligence

Human or Not?

"Am I conversing with a human or a computer?" is a question that a dozen judges grappled with at the annual Loebner Prize competition held at the University of Reading.

In 1950, British mathematician Alan M. Turing suggested that a computer could be said to be thinking if, in a text-based conversation, its responses are indistinguishable from a human's. Turing predicted that by the end of the 20th century, computers would have a 30% chance of being mistaken for a human in a five-minute text-based conversation.

In the 18th Loebner Prize competition, Elbot, one of six programs, nearly passed the Turing test, tricking 25% of judges into believing it was human. Each of the six programs fooled at least one judge.



Ken Perlin demonstrates the UnMouse Pad during the ninth annual Microsoft Research Faculty Summit at the Microsoft headquarters in Remond, WA.

engine with only his or her hands.

As humans get their hands and fingers into the act of manipulating everything, they must be careful in their expectations of how novel the actions are as new expressive multitouch, gestural, and bimanual interfaces are developed. And researchers must learn from Apple that it chose a set of multifinger tracking techniques for the iPhone; its interface makes the act of showing a photo to another person into a visual spectacle. Researchers must develop scenarios of use that people want to perform, and scenarios can be better than simple, ergonomic, and productive; they can be memorable, socially positive, and robust for both novice and expert use.

"The way it will end up is we will get better, as a species, for having our computer interfaces match the richness that is already built into our brains and body," Perlin says. "You watch people use sign language, gesturing, manipulating tools, or playing guitar in the real world. We have the potential of writing software that captures the richness of human interaction that has evolved over millions of years. When it becomes successful, it won't seem exotic. It will be as astonishing to people that they used to rely on a mouse and keyboard as it was to people when they used to program through punch cards."

Hands and eyes are the special connections that the human brain has to the physical world. In the next few years, many great advances are possible, such as 3D multitouch interfaces. Three dimensions can make a huge difference. For instance, George Miller's famous paper about short-term memory holding seven, plus or minus two things demonstrated that people remember many more items in a random list of numbers, letters, and words in 3D than in 2D.

Although coordinated multitouch input, such as a two-handed, chordic stenographer keyboard, might be great for words—and words are powerful—a picture is truly worth a thousand words. Multitouch interfaces offer more exciting ways to search for and display images. They might also drive faster use of information. These interfaces are demonstrating natural 3D metaphors, allowing coordinated manipulation to replace what would be multiple actions with a cursor control. In the near future humans can look forward to merging simple hand gestures with rich feedback in a 3D interface to create display and control surfaces that are simple to use, increase productivity, and produce more socially positive experiences.

Formerly an associate professor at the MIT Media Laboratory, **Ted Selker** develops and tests new user experiences in Palo Alto, CA. Saratoga, CA-based author and technology writer Wylie Wong provided additional reporting.

Quantum Cryptography

Quantum-Encrypted Network Debuts in Vienna

A team of international researchers and Siemens Austria unveiled the first commercial network protected by quantum encryption at a Development of a Global Network for Secure Communication based on Quantum Cryptography (SECOQC) conference in Vienna, BBC Radio reports. The network connected six different locations in Vienna and one in the neighboring town of St. Poelten, and involved nearly 125 miles of fiber-optic cable.

While standard network security encryption is based on complex mathematical equations that are extremely difficult to crack, it is vulnerable to parties who possess sufficient computing resources and time. The promise of quantum encryption is its reliance on the laws of quantum theory, which has been shown to be inherently unbreakable.

"All quantum security schemes are based on the Heisenberg Uncertainty Principle, on the fact that you cannot measure quantum information without disturbing it," Gilles Brassard, a computer scientist at Montreal University and a pioneer of quantum cryptography, told BBC Radio. "Because of that, one can have a communications channel between two users on which it's impossible to eavesdrop without creating a disturbance. An eavesdropper would create a mark on it. That was the key idea."

The Vienna network used extremely faint beams of light, equal to single photons being fired a million times a second, which raced between the nodes. When an intruder tried to eavesdrop on the quantum exchange, the photons scrambled, and the rise in the error rate at the node detectors announced the attack. The network shut down without being compromised, and the network connections were rerouted via other nodes without any interruption in connectivity.

The SECOQC researchers believe a business-viable network is possible in three years.

Society | DOI:10.1145/1409360.1409367

Samuel Greengard

Upwardly Mobile

Mobile phones are bridging the digital divide and transforming many economic, social, and medical realities, particularly in developing nations.

Few technologies have impacted the world as significantly as the mobile phone. The ability to connect with others *anywhere* and *anytime* has changed the way people think and behave. Yet, beyond phone calls, messaging, Internet access, the ability to snap photographs, and share data, these wireless devices have ushered in profound social changes that ripple into commerce, banking, healthcare, and beyond. "It is the first time in the history of technology that social class and geography are largely irrelevant," says Jhonatan Rotberg, a lecturer at MIT.

Although mobile phones have already transformed more affluent nations, they are ringing up some of the most profound changes—and biggest dividends—in developing countries, where new and innovative ideas, services, and methods of interaction are rapidly emerging. Today, people are using mobile phones to track crop prices in Kenya and manage micropayments in the Philippines. They are tapping into these devices to handle healthcare information in Nicaragua and oversee bakery orders in Nigeria.

In fact, with an estimated three billion-plus mobile phones in use worldwide and approximately 80% of the world's population within the reach of a cell tower, almost no corner of the globe remains untouched. "As the number of mobile phones has grown, an accompanying explosion in innovative approaches to using mobile technologies has taken place," says Jonathan Donner, a social scientist and researcher in the Technology for Emerging Markets Group at Microsoft Research in Bangalore, India.

Mobile technology, Donner says, "is creating broader economic and social development opportunities. Yet, at the same time, it raises questions about how we use the technology and the types of norms and expectations that should exist."

Dialing Into a Better Life

Although the idea for a mobile phone dates back to 1915 and wireless radio devices have been used for much of the 20th century, it wasn't until the 1990s that the technology took off in a significant way. As the size of mobile phones shrunk and the devices became simpler and more powerful, consumers began to snap them up in order to stay connected at work and in their personal life. Meanwhile, developing nations, which often had little wired infrastructure, began embracing mobile technology as a way to leapfrog expensive telecommunications investments and put phones in the hands of citizens.

The ability to skip straight to mobility has created remarkable economic opportunities. Jonathan Ledlie, a researcher for Nokia in Cambridge, MA, points out that without an existing infrastructure and legacy systems for handling financial transactions and human interactions, entrepreneurs in developing countries have become astute at inventing applications and processes that tap into the needs of these societies. "In some instances, entrepreneurs are deploying capabilities that are more advanced or innovative than those of developed nations," Ledlie observes.

"For the first time in history, information is no longer the exclusive domain of the powerful and the rich," says MIT lecturer Jhonatan Rotberg. "The ubiquity of mobile devices is changing the political and economic dynamics around the world."

The need for a person-to-person payment system prompted U.K.-based Vodafone to roll out a mobile banking system called M-pesa in July 2007. Vodafone predicted that it would sign up 200,000 subscribers in Kenya during the first year. Instead, it quickly exceeded the projections and acquired 1.6 million customers. Vodafone is now set to introduce mobile banking in neighboring Tanzania as well as India.

Meanwhile, firms such as Wizzit in South Africa and GCASH in the Philippines have introduced systems that allow customers to make purchases, payments, and withdrawals through the post office and kiosks. Those holding accounts can also exchange money and credits directly with others using these systems. And in places where no formal micropayment system exists, people have turned to their mobile phones as a way to exchange pre-paid phone card minutes via short message service (SMS).

Mobile technology is also changing how people farm. Sugarcane growers in Warana, India use their phones to check on water levels, fertilizer stock, and other supplies needed to run a cooperative, Donner says. With 70,000 farmers spread across 75 villages, buying and managing PCs is too expensive and impractical. Instead, an application using SMS disperses information to subscribers across the network. At the same time, some microentrepreneurs use mobile phones to expand their customer base. Donner tells the story of a Nigerian baker who started taking orders for cakes via SMS and quickly expanded his presence beyond his immediate neighborhood. He experienced a 30% increase in sales.

In fact, microfinance may represent the most significant aspect of mobile phone use around the world. So-called "inclusive capitalism" is making waves and changing the nature of some societies. In Bangladesh, one of the world's poorest countries, Grameen Bank uses microcredits to put mobile phones equipped with long-lasting batteries into the hands of women. They become a village phone provider and collect small commissions from their customers. Already, more than 250,000 "phone ladies" exist and Grameen Bank has grown into Bangladesh's largest telecom provider, with annual revenues approaching $1 billion. Similar programs have popped up in other countries, including Indonesia, Rwanda, and Uganda.

Healthcare is another area attracting attention. A program piloted in Nicaragua monitors tuberculosis patients via their mobile phones. Because compliance is critical and any break in treatment can result in a relapse or others becoming infected, patients must urinate on a reactive strip every day to reveal a code. However, the process of monitoring patients and sending a healthcare worker to collect results on a daily basis is both costly and time-intensive. Instead, officials now ask patients to send data via SMS and then reward them with free cellular minutes.

In some cases, the technology is bridging the gap between the Internet and phone messaging, and improving education. In South Africa, for example, a program allows students to query Wikipedia via SMS and receive audio text that they can record on their handsets and play back anytime they desire. The hybrid nature of the application achieves something relatively rare, Donner notes. "It breaks down the walls between Web content and SMS content. In doing so, it demonstrates a way in which rich, dynamic Internet content can be made accessible to—and can be created by—communities using relatively affordable and common basic mobile handsets."

Ledlie says that enterprising minds have created a slew of other mobile solutions, incorporating ideas as diverse as M-journalism and classified advertisements. The latter includes apartment listings and "available for work" postings that serve as a simplified form of Craigslist. "For people and societies without access to computers, these types of phone-based systems offer revolutionary capabilities," Ledlie explains. "They are likely to improve lives in significant ways."

Designs on the Future

Mobile phones have followed a predictable evolutionary path, says Minoru Etoh, a researcher at NTT DoCoMo's laboratory in Tokyo. The first stage of development was speech communication, the second stage was data communication (Web and email), and the third stage is life assistance. "SMS-based money transfers in places like India represent real life assistance," Etoh says. More importantly, "mobile phones may fill the digital divide between PC owners, who are typically more affluent, and non-PC owners."

For minorities and the disabled, mobile phones can provide critical capabilities as well as social networking opportunities that haven't previously existed. The technology also breaks down social structures and class divisions. "For the first time in history," Rotberg observes, "information is no longer the exclusive domain of the powerful and the rich. The ubiquity of mobile devices is changing the political and economic dynamics around the world. The technology is empowering people that have in the past been disenfranchised."

Those in more affluent regions are also altering the way they view the world. Social networking—through text and photo messaging, games, and sites such as Facebook and Twitter—are creating opportunities to interact in entirely different ways. "Social networking allows people to create distinct networks of friends, family, and colleagues and to broadcast what they are doing along with short updates about their lives," Ledlie says. In the coming years, he notes, phones are also likely to replace subway cards, parking passes, and credit and debit cards.

The challenge for engineers and designers is to build devices and interfaces that meet the needs of diverse populations. So far, most mobile phones have undergone a "trickle down" process of moving from more demanding and affluent users in developed nations to individuals in poorer countries. However, the situation is beginning to change, as Nokia and other manufacturers introduce phones that are sand-proof, incorporate flashlights, and use easily replaceable parts that better fit the needs of those who live in places where phones cannot be easily repaired.

Designers such as Etoh and Ledlie say that, in the future, it is vital to adapt user interfaces to better match the needs and requirements of specific

Artificial Intelligence

MIT Develops Autonomous Wheelchair

MIT researchers have created an autonomous wheelchair that has the ability to learn about locations inside a building and take its occupant to a specified place in response to a verbal command.

A wheelchair user need only say "Go to my room" or "Take me to the cafeteria" and the wheelchair, based on a map stored in its memory, will take its occupant to the desired location. The robotic wheelchair learns about an environment similar to how a recently hired employee learns about a new work environment—by being taken on a tour of the space and notified about the most important locales.

The robotic wheelchair was developed by MIT assistant professor of aeronautics and astronautics Nicholas Ray; Bryan Reimer, an MIT AgeLab research scientist; and Seth Teller, a professor of computer science and engineering and head of the Robotics, Vision, and Sensor Networks (RVSN) group at MIT's Computer Science and Artificial Intelligence Laboratory. Teller says the RVSN group is developing other machines with situational awareness, ranging from a mobile phone to an industrial forklift.

Beauty salon owner Josephine Macaladad, left, shares a lighthearted moment with Merlita Werlan, who converts cash into mobile phone-usable GCASH at the Balayan Public Market in Batangas, Philippines.

PHOTOGRAPH COURTESY OF MICROENTERPRISE ACCESS TO BANKING SERVICES (MABS) PROGRAM

groups and user segments. It's also necessary to design phones that support customized applications, such as those running on Java and Ajax, and develop better mobile Web browsers, improved speech recognition, and more streamlined user interfaces that blend voice, data, and Web features into a single, seamless package.

Researchers are also exploring ways to bring the Web closer to mobile phones, through audio wikis and audio "anthro" features that would allow people to record stories and share them. And while text messaging is inexpensive and easy to use, it lacks the advanced features of custom-designed applications, which come at a steep cost in terms of development and adoption. Finally, there's the social and psychological element. It's crucial to confront the implications of "individual addressability" and the changing personal boundaries created by mobile communications.

Yet, research and development marches on. MIT's Next Billion Network is now working to promote mobile technologies that affect change from the bottom up. The initiative aims to explore the use of mobile phones as more than mere talking and texting devices and to push viable technology solutions from the lab to real life. "Tremendous barriers to widespread adoption still exist," Rotberg observes. "But the opportunities for change and economic empowerment are enormous. We have only begun to unlock the power of mobile technology."

Promoting change is also at the center of David M. Reich's universe. The former telecommunications analyst has his sights set on developing a mobile marketplace to connect low-wage workers and employers around the globe. The company, Assured Labor, will roll out its service later this year in an attempt to address the 280 million individuals who are unemployed, underemployed, or toiling in subpar work conditions. "We want to make it simple for employers to find good and trustworthy workers quickly, anywhere, and at anytime," Reich explains.

Make no mistake, the future of mobile technology looks bright. "What makes mobile phones so remarkable," Donner says, "is that they connect people to people, whereas landlines connect places to places." By unwiring the world, people are ultimately able to rewire their minds—and change things for the better. "Mobile technology," he concludes, "has the potential to help enterprises, villages, cities, and regions connect and conduct matters of daily life more efficiently than ever before."

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

Human Rights

A Global Internet Policy

Google, Microsoft, and Yahoo! have joined forces to create the Global Network Initiative, a worldwide set of guidelines to help companies cope with requests from governments that might conflict with individuals' online privacy and freedom of expression.

Nearly two years in the making, the Global Network Initiative contains three core documents: a set of principles that its members endorse; guidelines for companies to implement internally that will detail how they handle information requests from governments; and a mechanism that will give independent auditors an opportunity to review a company's practices and judge whether they are in compliance with the program.

In addition to helping businesses protect human rights on the Internet, the guidelines are meant to stymie the private sector or government from imposing their own standards upon companies.

Response to the initiative has been mixed, but generally positive. Mike Posner, president of Human Rights First, which was involved with creating the program, called it "an important first step in providing standards for free expression and privacy that obligate companies to do more to challenge government restrictions." On the other hand, Marc Rotenberg, director of the Electronic Privacy Information Center, argued that, being self-regulatory, the proposals let companies "interpret them as they wish or back out of them when they choose."

A coalition of tech companies, human rights groups, academics, and others helped devise the program, and founding members Google, Microsoft, and Yahoo! have pledged to contribute $100,000 apiece for two years to support the Global Network Initiative, but several leading telecom companies are apparently content standing on the sidelines. AT&T, Verizon Communications, and Sprint Nextel in the U.S. have not joined the organization, nor has France Télécom and Vodafone in Europe.

News | DOI:10.1145/1409360.1409368

Leah Hoffmann

Making a Difference

The Grace Hopper Celebration featured technical talks, workshops, networking events, and lively discussions about increasing the number of women in computer science.

"We're trying to reflect how women think about technology," says Deanna Kosaraju, vice president of programs at the Anita Borg Institute, of the 2008 Grace Hopper Celebration of Women in Computing. The conference, co-founded by Anita Borg and Telle Whitney in 1994 and inspired by the legacy of Rear Admiral Grace Murray Hopper, balances a broad range of technical talks with professional and personal forums in an attempt to address the interests of women in computing. This year's theme, "We Build a Better World," took a holistic approach, emphasizing cooperation, teamwork, and the ability to make a difference. Nearly 1,450 women from 22 countries came together this October for the four-day meeting in Keystone, CO.

Anne Condon introduces Fran Allen, the first woman to win the A.M. Turing Award, to the audience at the Grace Hopper Celebration of Women in Computing conference.

This year, in order to help attendees get to know each other, organizers coordinated a variety of networking events during the first two days of the conference. A workshop with leadership coach Jo Miller explored how networking can help women recession-proof their careers; the event was filled to capacity with more than 400 attendees. Both structured and unstructured networking events followed, including a women of color luncheon and a special meeting for first-time attendees, enabling women to put their newly learned skills to the test. Meanwhile, a résumé clinic gave 160 women an opportunity to chat with 40 industry HR representatives and get advice on their résumés.

"It was a great way to break the ice," explains Anne Condon, conference chair and professor of computer science at the University of British Columbia.

To help women manage their networking goals, the CONNECT project enabled interested attendees to keep track of who they'd met with special scannable barcodes that automatically logged meetings between faculty, students, and industry representatives. Daily email messages summarized the connections each participant had made and included strategies to help them make the best use of their time; more than 16,000 connections were logged over the course of the conference. Social networking technologies like blogs, tweets, and Facebook also helped attendees keep track of one another, and sometimes plan collaborations before they'd even reached the conference.

"People have various ways of communicating, and we try to leverage that," says Kosaraju.

Another new event this year was the CTO Forum, a series of panels, lunches, and roundtable discussions that gave the female chief technical officers from Xerox, Intel, and other leading technology companies an opportunity to talk about topics of interest to the computing community. Among the subjects they explored were strategies to increase the number of women in computer science, the future of technology, and their own personal experiences. Other technical talks included presentations on Hewlett-Packard's "skinware" drug delivery technology, emerging energy technologies, data mining, and multi-robot intelligence. Mary Lou Jepson, CTO of Pixel Qi, also discussed her experiences with One Laptop Per Child, a global project to provide underprivileged children with new educational opportunities.

"From a technical perspective, it's a great place to get a broad view of what's going on," says Condon. Yet as difficult as it is to convey the breadth of the conference program, it's even harder, attendees and organizers agree, to describe just how exciting the atmosphere is.

"Very successful women are willing to talk to people who are just starting out," says Emily Fortuna, a senior at Rice University. "It's a very uniting experience."

Based in Brooklyn, NY, **Leah Hoffmann** writes about science and technology.

DOI:10.1145/1409360.1409369

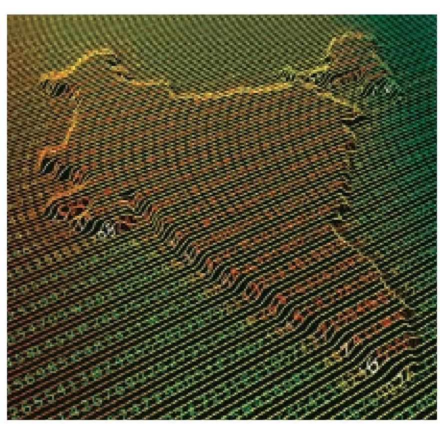
Alok Aggarwal

Emerging Markets
Labor Supply in the Indian IT Industry

Exploring the evolving dynamics and interconnectedness of India's educational system and its IT work force.

INDIA IS A country of extremes: for everything that exists in its socioeconomic structure, there exists an opposite somewhere close by. On one hand, the wealthiest family in India is in the process of building a billion-dollar mansion-house; on the other, approximately 10 million families live on government-owned land in shanties that each cost less than $100.[a] Similarly, although the seven Indian Institutes of Technology (IITs) produce approximately 4,000 graduates in engineering and computer science that are among the best in the world, approximately one-fourth of all high school graduates who want to pursue engineering or computer science are unable to do so, primarily because of cost.

According to research conducted by global research and analytics firm Evalueserve (www.evalueserve.com) approximately 85% of the labor supply for the IT industry (both exports and domestic) in India is composed of engineers (with four-year degree courses) and Master of



Computer Applications (MCA) graduates (with five-year degree courses). The remaining 15% comprises graduates (both five-year master's and three-year bachelor's) in mathematics, commerce, and sciences. During 1998 and 2008, after including inflation, the total amount spent on technical and IT education for such graduates grew by 17% every year. Whereas the class of 2004 had approximately 273,300 engineering and MCA degree graduates, the class of 2008 had approximately 454,200. During 1998 and 2008, the number of engineering and MCA graduates grew at an annual rate of 13.5%. Even if this growth rate can be maintained, the demand for such professionals is likely to increase at an annual rate of 15.5% until 2016 (see my July column), and hence there would be a cumulative shortage of 100,000 to 150,000 such graduates by 2016 (where no such shortage exists today).

The expenditure by the Indian government on technical and IT education increased from $375 million in 1996 to $830 million in 2006. Although the Indian government's overall expenditure grew at an annual rate of 8.4%, this rate is only half the overall growth rate of 17% mentioned earlier. More disconcerting is the fact that the Indian government's technical education expenditure as a percentage of total education expenditure decreased from 4.1% in 1996 to 3.9% in 2006. Furthermore, according to the National Knowledge Commission (www.knowledgecommission.gov.in) that advises the prime minister of India, approximately 75% of the current funds go toward salaries and benefits, 15% toward rent and utilities, and the remaining 10% are not even sufficient to main-

a All dollar amounts cited in this column are expressed in U.S. currency.

ILLUSTRATION BY ESKIMO SQUARE

tain let alone upgrade laboratories, buildings or libraries, or sufficient for pursuing research. Therefore, the National Knowledge Commission has requested—and the government has approved—a 190% increase in such funding from 2008 to 2009. However, it is not clear whether the Indian government has any additional funds to spend especially because the Indian K–12 education system requires even more urgent attention.

According to the University Grants Commission (UGC) of India (www.ugc.ac.in), the number of engineering colleges increased from 571 in 1998 to 1,645 in 2008, which represents an annual growth rate of 11.2%. Currently, out of these 1,645 engineering colleges, approximately 85% are privately owned, 10% are government colleges, and others are parts of universities that are wholly or partly funded by the central, state, or local governments. Approximately 88% of students graduate from private colleges, whereas others graduate from government-owned or government-aided colleges.

Broadly speaking, private engineering colleges can be classified into two categories. Less than 10% of colleges belong to the first category whose management and advisory boards comprise eminent and experienced individuals with substantial knowledge of technical education systems worldwide. More than 90% of private colleges belong to the second category and were mainly started for making profits. These colleges typically do not provide high-quality education and are therefore often unable to enroll enough students or attract good faculty. Since earning profit is their main motive, many Indian citizens feel such profit-making institutions should not be allowed; others welcome this private initiative because it fulfills an otherwise-unmet need.

Underwriting loans by the Indian government will allow the banks to become more liberal in giving loans and also reduce the carrying costs for students.

Only 12% of students graduate from colleges that are partly or wholly funded by the Indian government; most others resort to taking education-related loans from Indian banks. The amount of such education-related loans has grown from \$110 million in 2001 to \$3,450 million in 2007. Moreover, the educational loan as a fraction of total bank credit has also increased from less than 0.2% in 2001 to 1.1% in 2007, and the number of student borrowers has grown ninefold, from approximately 110,000 in 2001 to more than one million in 2007. Such a rapid increase in the number as well as the amount of loans is causing substantial discomfort within the Indian middle class, especially because the previous generation had to pay almost no tuition fee for college education (because all colleges used to be funded by the Indian government).

Keeping this in mind, the Indian government has recently announced establishing the National Student Loan Guarantee Corporation, which will underwrite most student loans and would also have a provision for waiving off loans if a student works for a government agency or in an underserved region in India for a predefined amount of time after graduating. Underwriting loans by the Indian government will allow the banks to become more liberal in giving loans and also reduce the carrying costs for students. However, this may also imply a bigger fiscal deficit for the Indian government, which is already reeling from a burgeoning deficit.

The All India Council for Technical Education (AICTE; www.aicte.ernet.in) is a part of the Indian government and approves new colleges after these colleges have fulfilled certain criteria. The National Board of Accreditation (NBA) is an autonomous body constituted by AICTE and provides accreditation to individual programs run by engineering colleges. Although NBA accreditation is mandatory for all engineering colleges, because of certain loopholes in the system, more than 80% of Indian engineering colleges currently do not have proper accreditation. From the colleges' perspective, however, many find it extremely cumbersome and frustrating to get all approvals and accreditation from AICTE and NBA. Hence, the National Knowledge Commission has requested the Indian government to establish a separate regulatory authority that is autonomous and devoid of any direct control by the Indian government.

Limited numbers of qualified faculty is currently the biggest challenge for engineering and IT education colleges in India. A recent report, *Faculty Development in Technical Education*, by Rama Rao et al. (http://www.isteonline.in/) estimated that currently there is a shortage of 75,000 faculty members (who have a minimum of master's degree in engineering or IT) and this shortfall is likely to exceed 100,000 by 2011. Moreover, approximately 95% of the 1,645 engineering and IT institutions in India do not have research programs, and in 2005, all the engineering and IT institutions combined awarded only 968 doctoral degrees. Multinational companies (IBM, Microsoft, Google, Yahoo) are exacerbating this shortage by opening research centers in India and paying their employees three to five times more than the corresponding universities and institutes.

Because private institutes are mainly looking to make profits and government-funded colleges and universities can only provide similar salaries as those provided to other employees of the Indian government, very few colleges in India are able to match the salaries being provided by multinational companies. Given these circumstances, even premier institutes like IITs have 10%–15% unfulfilled faculty positions; the situation is worse for second- and third-tier colleges. Moreover, a recent Evalueserve survey regarding the preferences of IIT graduates indicates fewer IIT graduates are likely to pursue academics, which would further decrease the availability of such faculty members. Hence, to mitigate this shortage, many colleges have started requesting retired faculty return and teach; others are hiring recent graduates (with only a bachelor's degree) who are not able to find jobs elsewhere.

Since there is an acute shortage of faculty and because previously retired faculty members or those with substandard skills are being recruited to teach in most engineering and IT colleges, the curriculum in such colleges has remained unchanged during the last two decades. For example, grades are still determined by only a final examination and not by grading intermittent tests, laboratory work, or homework assignments, and the final examinations only test students' memory rather than their creativity or ability to think independently. Therefore, students often forget the basics related to microprocessors, Internet Protocol, or the Java language by the time they graduate.

In addition, while the Indian IT exports industry has a severe need for students to have "soft qualifications" (such as proficiency in spoken and written English and the ability to manage projects efficiently), this education system does not provide any such courses. Hence, it is not surprising that a recent study by the strategy consulting firm McKinsey and Company (www.mckinsey.com/mgi) indicated only 25% of engineering graduates would have the requisite qualifications to be employed in the Indian IT exports industry. Those who do not possess such qualifications either work in the domestic IT industry or change their careers altogether, thereby earning half to two-thirds of what they would have earned otherwise.

Clearly both the public-funded and the private-funded education programs are inherently inadequate, and even combined would not be able to meet the burgeoning demands of the Indian domestic and exports industries. With this in mind, the three initiatives described in the following paragraphs have gained momentum during the last decade.

Realizing the Indian college system (which was primarily government funded) was unable to provide much IT training, in 1981 two entrepreneurs started an IT training college, NIIT, which provided supplemental tutorials and training classes. During 1998–2008, NIIT grew enormously and resulted in the formation of a supplemental IT training industry in India, which provided 3.2 million student-course units and had approximately $610 million in revenue in 2007. These training colleges provide short-term courses such as Java, C++, SQL, SQL database, and .NET to corporate clients who send their employees for supplemental training; a course typically costs between $75 and $300 per student. Long-term courses have durations of three years and cost between $2,500 and $3,500 per student. Most students enroll in their long-term courses while very few working people enroll in evening or weekend classes. Also, most students who enroll in long-term courses are also concurrently pursuing undergraduate or graduate programs. Finally, most of these training institutes provide a job guarantee for students enrolled in long-term courses and also have partnerships with banks to provide five- to seven-year loans.

Many IT export companies in India have created their own training institutes and have begun training new employees.

During 1998–2000, when the Indian IT industry was experiencing unprecedented growth due to the dot-com boom and the Y2K problem, Indian central and state governments established Indian Institutes of Information Technology (IIITs), mainly as public-private partnership institutions.[b] The main goal of these IIITs is to provide education at the master's and Ph.D. levels although some of them are also providing undergraduate education.

Currently, there are eight IIITs in India that graduate approximately 3,000 students every year, and the Indian government has announced the creation of another 20 during the next five years with the goal of producing at least 20,000 master's and Ph.D.s every year. Furthermore, faculty members are likely to be given competitive salaries and each IIIT will have the autonomy to decide its own salary structure so as to compete with private institutes and multinational corporations. For each IIIT, the Indian government would provide 50 to 150 acres of land and approximately $25 million in seed funding and would have only a minority representation on the board. Private companies have been playing a major role in providing ongoing finance, faculty, and most importantly, governance. Furthermore, the collaborating private organizations would be requested to send their experienced employees as visiting faculty members and more faculty-exchange programs with universities abroad are being envisioned.

Many IT export companies in India have created their own training institutes and have begun training new employees. For example, Infosys intends to spend $170 million in training and Wipro has its own training campus where it can train more than 5,000 employees simultaneously. Overall, Evalueserve estimates the Indian IT industry may ultimately spend approximately $1.1 billion—or about 3% of the revenue it earned—during 2008–2009.

Some of the key programs started by Indian IT companies include a seven-month program, Tata Ignite, wherein Tata Consulting Services would train science and mathematics graduates (with three-year degree courses) to become software professionals and then employ them; Campus Connect launched by Infosys in May 2004; Mission10X launched by Wipro Technologies, which is trying to promote changes to current teaching-learning paradigms, will include 3,000 faculty members in 2009 and 6,000 faculty in 2010; Satyam Entry-Level Engineering Development (SEED) program launched by Satyam, which is a four-month training program for entry-level engineers who will be eventually employed by Satyam; and the establishment of approximately 100 Centres of Excellence (CoE) jointly by HCL Infosystems and Microsoft, which will provide training and certification in Microsoft technology to 50,000 students.

As I've described in this column, the process of IT education in India continues through its chaotic and lumbering evolution. Nevertheless, in spite of its fits and starts, this process seems to be progressive, and by 2016, India will have the second-largest IT labor force in the world that is likely to be only 20%–25% smaller than its U.S. counterpart.

b For example, the chairman for IIIT Bangalore is Narayana Murthy, who is the cofounder and former chairman of Infosys.

Alok Aggarwal (alok.aggarwal@evalueserve.com) is the cofounder and chairman of Evaluserve, Inc. in Saratoga, CA.

DOI:10.1145/1409360.1409370

Cameron Wilson and Peter Harsha

IT Policy
Advising Policymakers Is More Than Just Providing Advice

What are the factors that make certain advisory committee reports successful while others are not?

In our September column we discussed the somewhat grim truth that good science and the presence of a rational argument do not guarantee that policymakers will do the "right thing" when crafting policy. The intersection of science and politics is fraught with compromise and trade-offs, and elected representatives will always seek to balance competing interests. While this might appear to diminish the importance of getting sound science and technical advice to policymakers, the reality is that science can have a significant impact on policy—particularly in the context of federal advisory committees. But not all panels and advisory boards are created equal.

In 1999, the President's Information Technology Advisory Council (PITAC) issued a report, *Information Technology Research: Investing in Our Future*, reviewing the U.S. federal government's investment in IT research and calling for substantial new investments in basic IT research. Six years later, a re-formed, newly populated PITAC issued a report on cyber security, calling the research portfolio unbalanced toward short-term investments and recommending modest new resources be dedicated toward long-term cyber security research. The 1999 report helped set in motion one of the largest increases in funding for IT basic research, while the 2005 cyber security report was largely ignored by policy leaders. In both cases

ILLUSTRATION BY BRIAN STAUFFER

these advisory committees had highly qualified members, produced well-grounded reports, and had specific recommendations. So what makes one advisory committee report successful, while another collects dust?

With thousands of advisory committees of all different shapes and sizes operating each year it is difficult to draw specific conclusions. We have had the fortunate experience to work with many members of our community who have been asked to serve on advisory committees, or asked to staff or convene them while working as officials in federal agencies. Drawing from these experiences, we have identified some general characteristics—transparency, access to key staff, understanding the political context and balance—that successful panels appear to share, and that policymakers and advisory committee members should strive to emulate when establishing a panel.

Federal advisory committees lay at the heart of an age-old tension of how to inject scientific knowledge into the policy-making process. Congress or the Executive Branch can create committees to serve a number of different roles, and usually they are comprised of a mix of outside experts and government officials. Official committees are chartered under the Federal Advisory Committee Act (FACA), which provides the framework for their membership, operations, and oversight. With approximately 1,000 advisory committees operating for the U.S. federal government in any given year, they come with differing membership, goals, focus, and scoping.[a]

There are "blue ribbon" commissions, typically created to provide some focused policy direction; standing committees, usually created by law, such as the now-disbanded PITAC, to provide high-level strategic direction; highly specialized committees providing detailed guidance on narrow technical problems, and countless other incarnations mixing these three areas. Of growing importance to the scientific community is a fourth type created under the National Academies. Congress and the administration routinely fund the Academies to bring together top technical and subject matter experts to provide recommendations on some of the most difficult policy issues. The Academies are becoming the scientific advisory arm of the U.S. federal government.

Advisory committees are created for a number of reasons. It is common for members of Congress, mired in difficult policy negotiations, to create an advisory committee to break political deadlock over a key issue. This effectively delays some difficult decision until a later date with an implicit promise the issue will be revisited when a report is issued. While this is a fairly common scenario, there are many others. Agencies can create committees to answer very narrow, highly

It is common for members of Congress, mired in difficult policy negotiations, to create an advisory committee to break political deadlock over a key issue.

technical questions; look to standing committees for high-level strategic advice; or get the perspective of many different groups before regulating an industry.

Given the variation in types of committees combined with the numerous circumstances motivating their creation, the different contexts in which advisory committees operate is infinite. But the ultimate goal for any advisory committee is to provide useful recommendations and have their advice be heard, considered, and acted upon. Answering what will make an advisory committee effective or not depends on many variables (including completely unanticipated external events), but our view is that some general, largely interconnected rules, can apply to most advisory committees.

Transparency. The mission and focus should be clear to members of the committee, to the staff or principles receiving the advice, and to the public. FACA and agency guidance outline the requirements for chartering a committee and the ground rules for public or private meetings, but there is more to transparency than open meetings and what is spelled out in official documents. Committee members should have a good sense of the goals of the programs they are reviewing, the key questions officials would like answered and, perhaps most importantly, whether those officials are willing to listen to the answers. Without such information committees can spend endless cycles looking for direction.

Access. Access encompasses a number of aspects from interacting with senior policymakers to support from lower-level staff. First, committee members should have access to dedicated agency staff to help with in-depth research. Key documents are often buried deep within a bureaucracy and can only be obtained by having a staff advocate on the inside. Second, since most members of advisory committees are volunteers, whether a committee has support from staff to draft reports can mean the difference between having a final product that is a comprehensive report or a basic PowerPoint presentation. Third, the committee should have the ear of policymakers that can make or help drive new policy directions. FACA requires that agencies appoint staff to oversee and attend meetings, and approve a committee's agenda, but this engagement can vary widely from active participation of senior officials to pro forma staff attendance.

One particularly successful model has been the U.S. Department of Defense's Information Science and Technology Board. These advisory committees typically have a narrow technical focus and, in choosing topics, have a fair amount of interaction with senior department managers. This level of engagement can lead to a potential tradeoff between independence and success that is typical of many committees. As ISAT member Peter Lee from the Carnegie Mellon computer science department said, "While the input of DARPA program managers in the selection of study topics can be viewed as 'meddling', it also means that ISAT studies have an unusually high 'hit rate', in the sense of influencing DARPA investment decisions, particularly in new programs."[b]

Understanding Political Context. The type of access a committee has often goes hand-in-hand with the political context in which it operates. These contexts cover a wide spectrum. At one end, the scientific and technical questions being asked are highly specific and the values of those asking for the report

a According to the General Services Administration, there were 963 advisory committees operating under FACA during Fiscal Year 2008, see summary data at http://www.fido.gov/facadatabase/AgenciesList.asp.

b P. Lee. Guestrin, Goldstein, and Haigh present DARPA ISAT study. *CS Diary*, http://www.csdhead.cs.cmu.edu/blog/2007/10/15/guestrin-goldstein-and-haigh-present-darpa-isat-study/ (Oct. 15, 2007).

are broadly shared leading to a narrow context. For example, the blue-ribbon commission investigating the Space Shuttle *Challenger* disaster received widespread support because members of both major political parties shared the same values about finding what led to the disaster and addressing the issues with the manned space flight program. At the other end, committee members themselves or the intended audience of policymakers may have widely divergent values about the key policy issue under consideration by the committee. This can lead to a lack of access to key policymakers, who may seek to marginalize a committee, or a report that is political non-starter and ends up gathering dust.

At times the U.S. Homeland Security's Privacy Advisory Committee, which often deals with scientific and technical questions underpinning privacy issues, has produced reports on controversial subjects where it is clear the committee and the administration have two different sets of values, hampering the impact of the committee and often leading to technical recommendations not being adequately addressed. Last year, the committee issued a report questioning the proposed implementation of a highly controversial federal driver's license identification standard for all 50 states called the "Real ID" Act.[c] Many from the privacy and technical communities shared the committee's concerns and recommendations (including ACM's U.S. Public Policy Committee), but the divergence of values between the committee and the administration ultimately marginalized the committee on the issue and the report seemed to have little impact on the administration's final policy choices.

Navigating the political landscape is arguably one of the most difficult things an advisory committee can do. In fact, we've heard the view that advisory committees simply shouldn't worry about politics and the "science" should win out. But this view doesn't take into account the reality of the policy-making process. As we noted in our September column, although it is a worthwhile goal to present the best science, simply providing the best science will not necessarily lead to a more "scientifically based" policy choice. Policy-making is built upon a political system that, for the most part, seeks to resolve value differences, not scientific differences, between groups. The same dynamic applies to scientific and technical advisory committees, as the findings they make will have to feed back into the political system that called for the advisory committee. Committees should have a good understanding of whether their recommendations are realistic, just as policymakers should truly listen to the committee to see where science and technology can help expand the set of options available and change the political dynamics.

Navigating the political landscape is arguably one of the most difficult things an advisory committee can do.

Balance. Despite requirements for both regional and ideological balance in FACA, the selection of advisory committee members remains highly controversial. Outside groups have claimed the current administration has stacked advisory committees to provide third-party validation of predetermined policy decisions instead of giving truly impartial advice. The issue took on such significance that the National Academies issued a report commenting on advisory committee membership stating, "With regard to appointing scientists and engineers to federal advisory committees, charges have surfaced recently that the process of making these appointments has become politicized and results in a skewing of the impartial perspective critical to independent advice. It is essential that the government's capacity to consider and incorporate science and technology information as part of the basis for public-policy decisions not be compromised by unnecessary obstacles."[d]

In recent years, this issue has dominated the discussion surrounding scientific and technical advisory committees. While we agree ensuring balanced scientific representation is important, we would argue there is another key factor to consider. Having communicators and leaders that can navigate the political waters and push the committee's findings and recommendations is equally important. Simply ensuring balance and appropriate technical focus without looking at the broader issues involved with most advisory committees propagates the myth that providing the "right" scientific information will naturally lead to better policy. In our view, the most effective committees have all had some members who were effective communicators and who were willing to carry the committee's recommendations to other audiences.

The intersection of science and policy is, by necessity, complex and imperfect. While policymakers often strive to understand the science underpinning issues before them, the desire to balance the needs of a seemingly endless collection of stakeholders can often render the evidence of that understanding very difficult to find. Increasingly, however, issues of science and technology are fundamental to almost every major policy issue before Congress and the administration—national security, voting rights, health care, and the economy—to name just a few. It has never been more important that federal policymakers get timely and pertinent science and technology advice, and it has never been more important that members of the science and technology community participate actively in this process. We hope our observations will help make the time and effort of those participants more effective.

c See http://www.dhs.gov/xlibrary/assets/privacy/privacy_advcom_05-2007_realid.pdf.

d *Science and Technology in the National Interest: Ensuring the Best Presidential and Federal Advisory Committee Science and Technology Appointments.* The National Academies 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:10.1145/1409360.1409372 George V. Neville-Neil

Kode Vicious
Permanence and Change

Highlighting the importance of doing one's best in view of code longevity and the impermanence of the changineer.

Dear KV,
When you first write a piece of code how long do you plan to maintain it for? Are you more careful in writing a significant class versus a 100-line, throwaway script? It can't be the case that every piece of code you write has to undergo the same process and the same level of scrutiny as every other, can it? Do you ever just leave a piece of code behind and let it fade into the past?

Fearing and Loathing Past Code

Dear Fearing and Loathing
I believe I am like other programmers in that there is plenty of code I'd like to forget. Unfortunately, although you can personally forget a piece of code, and at my age I have now forgotten more code than I've written—wait, is that possible? At any rate, although yes, you can forget old code, it never forgets you and thanks to the Internet many other people may be able to read what you've forgotten. If you've ever contributed to an open source project your code will probably never be lost, though perhaps it will be happily obscured by more recent checkins.

Having grown up with a mother who would make me redo my homework after saying, "Never time to do it right, always time to do it over." I have more than a little bit of guilt associated with doing things right. Happily, I've since been able to work through some of the guilt but there is still a thin patina of it present in my soul that makes me care for each piece of code as if it is something someone else will someday read, and I don't want them to be disappointed. My main goal is for the code to be good enough that even if, in 10 years, when I see the code again, I won't find it lacking. I have not always achieved this goal.

Computer programmers and engineers are in a unique position when it comes to the work we do. We are both scientists, dealing in the abstractions that help us solve problems as well as craftspersons, turning those abstractions upon our modern lathes that help make the modern world what it is. When we craft our code well it works more effectively and more beautifully, helping to create better systems. When we rush the job, and ignore the nicks and scratches, the parts fail to fit and eventually we're left with software that looks more like that fort you built in the backyard when you were a child. Sure, it was fun to hang out in that fort, but you ran back to the house at the first sign of a real rainstorm.

My favorite word in Japanese is ganbarimasu, 頑張ります, which is a single word that takes the place of the English phrase, "Do your best." If someone at work admonished you to do your best you'd probably think them a bit odd. The Japanese say this to each other relatively frequently, and they take the admonishment seriously, whether they are sweeping a street, cooking food, building a house, or writing code. I admonish you to do the same with all your code.

When you make a practice of doing things right the first time, every time, it becomes easy and natural. So, yes, whether you're writing a guidance system for an aircraft or a clean-up script for your mail folder, you should do your best. Someday someone else may need that code and you don't want them to think your code is lacking, do you?

KV

Dear KV,
I was installing, or actually trying to install, some software the other day and was caught in this infinite loop of swapping software, hardware, and cables just to get one small thing to work. What a waste of time! There must be some way to avoid these types of loops.

Feeling Loopy

Dear Loopy
Clearly since you had time to write to me the loop wasn't infinite, now was it? You know how to detect an infinite loop? Figure out how many times you're willing to do the same thing without getting a different result and then stop after that many iterations! After all, insanity is often defined as doing the same thing over and over

again but expecting a different result. You don't want your coworkers to think you're insane, unless, of course, you like sitting along at lunch. Still, there are better ways to convince people you're out of your mind than repeating the same stupid process and getting nowhere.

What you describe makes you sound more like a changeineer than an engineer. What's a changeineer? Usually this is the person they send out to a remote site, like a data center, when something breaks. They don't really know how the system works or how to fix it, but they do know how to change every single component in a system until the problem magically disappears, at least for a few days. Disk is slow? Change the disk. That didn't work? Upgrade the CPU. Still slow? Get faster RAM. Ad nauseam.

Changineer is a term I learned from some friends who work in a data center. Unsurprisingly I heard the term over drinks, at lunch—yes, it had been that kind of day. They see changineers every day, and learn to spot them and do away with them quickly so they can get problems solved instead of having them obscured by many random new components.

The thing you want to avoid is becoming a changeineer. Changineers use their hands, or worse, a text editor, while engineers use their heads. When you find yourself doing the same thing over and over and you're still frustrated it's time to walk away from the problem and think about it. Paper and pen or pencil are still the best for this, or perhaps a whiteboard. Write down the symptoms. Think about what could be causing the problem. Come up with a hypothesis—that is, a guess—about the real cause of the problem. Figure out how you would test your guess. Then test the guess. The process can also be repeated but unlike the process you described each change is preceded by careful thought. You're a lot less likely to go on spinning around.

KV

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

DOI:10.1145/1409360.1409371
Peter J. Denning, Chris Gunderson, and Rick Hayes-Roth

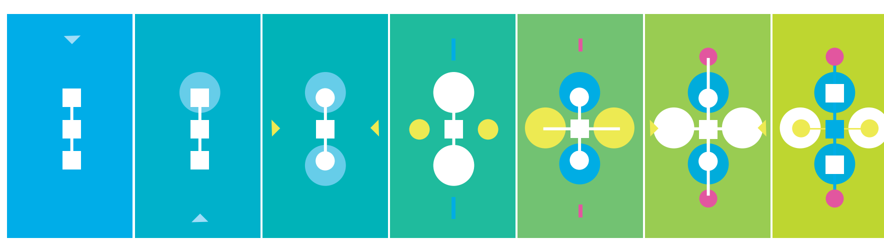
The Profession of IT
Evolutionary System Development

Large systems projects are failing at an alarming rate. It's time to take evolutionary design methods off the shelf.

Many critical large systems are failing. The replacement FAA air traffic control system, the FBI virtual case file, and the Navy Marine Corps Internet (NMCI), are a few of the many billion-dollar systems that could not deliver the functions needed. In stark contrast, the Boeing 777 aircraft, the Global Positioning System (GPS), and the U.S. Census database system have been outstanding successes. Why do some systems fail and others succeed?

Development time is the critical factor. This is the time to deliver a system that meets the requirements set at the beginning of the development process. If development time is shorter than the environment change time, the delivered system is likely to satisfy its customers. If, however, the development time is long compared to the environment change time, the delivered system becomes obsolete, and perhaps unusable, before it is finished. In government and large organizations, the bureaucratic acquisition process for large systems can often take a decade or more, whereas the using environments often change significantly in as little as 18 months (Moore's Law).

The Boeing 777, GPS, and U.S. Census data systems were developed for stable environments—they were completed before any significant changes occurred in their requirements. In contrast, the FAA replacement system, FBI Virtual Case File (see www.spectrum.ieee.org/sep05/1455), NMCI (GAO[4], www.nmcistinks.com) all faced dynamic environments that changed faster than their development processes could. Predecessors of these systems were successful because their environments were stable, but the current generations encountered trouble because their environments had become too dynamic.



The traditional acquisition process tries to avoid risk and control costs by careful preplanning, anticipation, and analysis. For complex systems, this process usually takes a decade or more. Are there any alternatives that would take much less time and still be fit for use?

Yes. Evolutionary system development produces large systems within dynamic social networks. The Internet, World Wide Web, and Linux are prominent examples. These successes had no central, preplanning process, only a general notion of the system's architecture, which provided a framework for cooperative innovation. Individuals in the network banded into small groups to quickly produce or modify modules in the architecture. They tested their modules by asking other users to try them. The systems evolved rapidly in many small increments that aligned with current perceptions of the using environment.

Moreover, the evolutionary process embraces risk, and the patience to see what emerges. It works with nature's principle of fitness in the environment: components that work well survive, and those that do not are abandoned.

The astonishing success of evolutionary development challenges our common sense about developing large systems. We need to learn from these systems, because evolutionary development may be the only way to achieve satisfactory replacements for aging large systems and to create new, unprecedented systems.

Evolutionary development is a mature idea that has languished away from mainstream practice. In this column, we will analyze why evolutionary development does not fit the current common sense and why we need to work to change that.

Our Current Common Sense

From its founding in 1968, the software engineering field set out to address the "software crisis," a persistent inability to deliver dependable and usable software. Fritz Bauer, one of the field's founders, believed a rigorous engineering approach was needed. He famously quipped, "Software engineering is the part of computer science that is too hard for computer scientists." Over the years, software engineers produced many powerful tools: languages, module managers, version trackers, visualizers, and debuggers are some examples. In his famous "No silver bullet" assessment (1986), Fred Brooks concluded that the software crisis had not abated despite huge advancements in tools and methods; the real problem was getting an intellectual grasp of the problem and translating that understanding into an appropriate system architecture.[2] The tools of 1986, while better than those of 1968, relied on concepts that did not scale up to ever-larger systems. The situation today is much the same: tools are more powerful, but we struggle with scalability, usability, and predictability.

Current software engineering is based on four key assumptions:

- Dependable large systems can only be attained through rigorous application of the engineering design process (requirements, specifications, prototypes, testing, acceptance).
- The key design objective is an architecture that meets specifications derived from knowable and collectable requirements.
- Individuals of sufficient talent and experience can achieve an intellectual grasp of the system.
- The implementation can be completed before the environment changes very much.

What if these assumptions no longer hold? The first assumption is challenged by the failures of large systems that used the traditional design process and the successes of other large systems that simply evolved. The remaining assumptions are challenged by the increasingly dynamic environments, often called ecosystems, in which large systems operate. There is no complete statement of requirements because no one person, or even small group, can have complete knowledge of the whole system or can fully anticipate how the community's requirements will evolve.

The astonishing success of evolutionary development challenges our common sense about developing large systems.

System Evolution: A New Common Sense

To avoid obsolescence, therefore, a system should undergo continual adaptation to the environment. There are two main alternatives for creating such adaptations. The first, successive releases of a system, is the familiar process of software product releases. It can work in a dynamic environment only when the release cycle is very short, a difficult objective under a carefully prescribed and tightly managed process. Windows Vista, advertised as an incremental improvement over XP, was delivered years late and with many bugs.

The second approach to adaptation is many systems competing by mimicking natural evolution; the more fit systems live on and the less fit die out. Linux, the Internet, and the World Wide Web illustrate this with a constant churn of experimental modules and subsystems, the best of which are widely adopted.

Evolutionary system design can become a new common sense that could enable us to build large critical systems successfully. Evolutionary approaches deliver value incrementally. They continually refine earlier successes to deliver more value. The chain of increasing value sustains successful systems through multiple short generations.

Designs by Bureaucratic Organizations

Fred Brooks observed that software tends to resemble the organization that built it. Bureaucratic organizations tend toward detailed processes constrained by many rules. The U.S. government's standard acquisition practices, based on careful preplanning and risk avoidance, fit this paradigm. Their elaborate architectures and lengthy implementation cycles cannot keep up with real, dynamic environments.

It may come as a surprise, therefore, that practices for adaptability are allowed under government acquisition rules. In 2004, the Office of Secretary of Defense sponsored the launch of W2COG, the World Wide Consortium for the Grid (w2cog.org) to help advance networking technology for defense using open-development processes such as in the World Wide Web Consortium (w3c.org). The W2COG took advantage of a provision of acquisition regulations that allows Limited Technology Experiments (LTEs). The W2COG recently completed an experiment to develop a secure service-oriented architecture system, comparing an LTE using evolutionary methods against a standard acquisition process. Both received the same government-furnished software for an initial baseline. Eighteen months later, the LTE's process delivered a prototype open architecture that addressed 80% of the government requirements, at a cost of \$100K, with all embedded software current, and a plan to transition to full COTS software within six months.

In contrast, after 18 months, the standard process delivered only a concept document that did not provide a functional architecture, had no working prototype, deployment plan, or timeline, and cost \$1.5M. The agile method produced a "good enough" immediately usable 80% success for 1/15 the cost of the standard method, which seemed embarked on the typically long road to disappointment.

Agile Methods for Large Systems

Agile system development methods have been emerging for a decade.[1,3,6] These methods replace the drawn-out preplanning of detailed specifications with a fast, cyclic process of prototyping and customer interaction. The evolutionary design approach advocated here is a type of agile process.

The U.S. Government Accounting Office (GAO) has scolded the government on several occasions for its uncommitted lip service to agile processes.[4] The GAO believes agile processes could significantly shorten time to delivery, reduce failure rate, and lower costs. Many people resist the GAO advice because they assume careful preplanning minimizes risk and maximizes dependability and usability. However, more leaders are pushing for agile acquisition because the track record of the normal process in dynamic environments is so dismal.

The software engineering community has hotly debated preplanned versus agile processes. After a while they reached a truce where they agreed that preplanning is best for large systems where reliability and risk-avoidance are prime concerns, and agile is best for small to medium systems where adaptability and user friendliness are prime concerns.

We challenge that conclusion. Preplanning is ceasing to be a viable option for large systems. Moreover, many small systems aim to be ultra-reliable.

Evolutionary Ecosystems

Evolutionary development uses "loosely managed" processes. Numerous successful large systems evolved through such a process—CTSS, Unix, Linux, Internet, Google, Amazon, eBay, Apple iPhone Apps, and banking applications are notable examples. All these systems relied on a common platform used by all members of the community, from developers to users. In such an ecosystem, successful prototypes transition easily to working products. It appears that the common ecosystem provides enough constraints that loose management works. The successful ecosystems were guided by a vision and a set of interaction rules that everyone in the community accepted. Building ecosystems for governments is quite challenging because of organizational impediments to information sharing.[5] We advocate much more aggressive use of loosely managed ecosystems. The W2COG was conceived to allow government to join a large ecosystem that could adaptively address its information networking needs.

Loosely managed does not mean unmanaged. Scrum and Extreme Programming (XP) are often cited as successful management approaches for agile processes.[6] Even the respected Capability Management Model (CMM) is amenable to agile development.

Whereas preplanned development seeks to avoid risks, evolutionary development mimics nature and embraces risks. The developers purposely expose emerging systems to risks to see how they fail, and then they build better system variants. It is better to seek risk out and learn how to survive it. In a natural ecosystem, only the most fit organisms survive. Fitness is nature's way of managing risk.

Whereas preplanned development seeks to avoid risks, evolutionary development mimics nature and embraces risks.

All the evidence says that that evolutionary processes works for systems large and small, and that risk seeking is the fastest route to fitness. There is too much at stake to continue to allow us to be locked into a process that does not work.

References

1. Boehm, B. Making a difference in the software century. *IEEE Computer* (Mar. 2008), 32–38.
2. Brooks, F. *The Mythical Man Month.* Anniversary Edition. Addison-Wesley, 1995.
3. Cao, L. and Balascubramaniam, R. Agile software development: Ad hoc practice or sound principles? *IEEE Pro* (Mar.–Apr. 2007), 41–47.
4. GAO. *Defense Acquisitions: Assessments of Selected Weapons Programs.* Report GAO-06-391 (Mar. 2006); http://www.gao.gov/new.items/d06391.pdf, and *Information Technology: DOD Needs to Ensure That Navy Marine Corps Intranet Program Is Meeting Goals and Satisfying Customers.* Report GAO-07-51. (Dec. 2006); http://www.gao.gov/new.items/d0751.pdf.
5. Hayes-Roth, R., Blais, C., Brutzman, D. and Pullen, M. How to implement national information sharing strategy. *AFCEA-GMU C4I Center Symposium: Critical Issues in C4I*, George Mason University, Fairfax, VA, AFCEA (2008); http://c4i.gmu.edu/events/reviews/2008/papers/25_Hayes-Roth.pdf.
6. Schwaber, K. *Agile Project Management with Scrum.* Microsoft Press, 2004.

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

Chris Gunderson (cgunders@w2cog.org), Captain (retired) U.S. Navy, is Principal Investigator of the Naval Postgraduate School W2COG and Netcentric Certification Office initiatives.

Rick Hayes-Roth (hayes-roth@nps.edu) is Professor of Information Systems at the Naval Postgraduate School in Monterey, California, and was CTO for Software at Hewlett-Packard Company.

Calendar of Events

December 17-20
HiPC '08: 15th International Conference on High Performance Computing
Bangalore, India
Sponsored: SIGARCH
Contact: Ajay K. Gupta,
Phone: 269-276-3104
Email: ajay.gupta@wmich.edu

January 7-9
International Conference on Multimedia Modeling
Sophia Antipolis, France
Contact: Benoit Huet,
Phone: +33-0-493008179
Email: benoit.huet@eurecom.fr

January 14-17
International Conference on Bio-Inspired Systems and Signal Processing
Porto, Portugal
Contact: Joaquim B. Filipe,
Phone: 351-91-983-3996
Email: jfilipe@insticc.org

January 19-21
International Conference on Agents and Artificial Intelligence
Porto, Portugal
Contact: Joaquim B. Filipe,
Phone: 351-91-983-3996
Email: jfilipe@insticc.org

January 19-22
Asia and South Pacific Design Automation Conference
Yokohama, Japan
Contact: Yutaka Tamiya,
Phone: +81-44-754-2663
Email: tamiya.yutaka@jp.fujitsu.com

January 20-23
The Eleventh Australasian Computing Education Conference
Wellington, New Zealand
Contact: Margaret Hamilton,
Phone: 613-992-52939
Email: mh@cs.rmit.edu.au

January 23-24
International Conference on Advances in Computing, Communication and Control
Mumbai, India
Contact: Srija Unnikrishnan,
Phone: +919869005457
Email: srija.unni@gmail.com

DOI:10.1145/1409360.1409373

James Hamilton

Interview
Database Dialogue with Pat Selinger

Relational database pioneer Patricia G. Selinger explores the vast realm of database technology and trends in a wide-ranging discussion with Microsoft's James Hamilton.

IBM FELLOW AND retired director of the Database Technology Institute at IBM's Almaden Research Center Patricia G. Selinger has made tremendous contributions to the database industry. After joining IBM Research in 1975, Selinger became a leading member of the team that built System R, the first proof that relational database technology was practical. Her innovative work on cost-based query optimization has been adopted by the majority of relational database vendors and is taught in most university database courses. Selinger managed the Almaden Computer Science department and established the Database Technology Institute, a joint program between IBM Research and the IBM software development team that accelerated advanced technology into data management products such as DB2 Universal Database for z/OS, IMS, DB2 UDB on Linux, Windows, and Unix.

The interview presented here was conducted by James Hamilton, a member of Microsoft's SQL Server Team and the former lead architect on DB2 and leader of the C++ compiler project at IBM.[a]

JAMES HAMILTON Let's start with the role of a query optimizer in a relational database management system and your

invention of cost-based optimizers.

PAT SELINGER As you know, the fundamental tenet of a relational database is that data is stored in rows and columns. It's value-based in that the values themselves stand up for—represent—the data. No information is contained in pointers. All of the information is reflected in a series of tables, and the tables have a certain well-known shape and form: there's an orders table, a customers table, an employees table, and so forth. Each of those tables has a fixed set of columns: the first name, the last name, the address.

Relational systems have a higher-level language called SQL, which is a set-oriented query language. This is a unique concept and really what distinguishes relational database systems from anything that came before or after.

The set-oriented concept of the query language allows asking for all the programmers who work in department 50; or all of the orders over $5,000; or all of the San Jose customers who have orders over $5,000; and so forth. The information in relational tables can be combined in many different ways, based on their values only.

How do you take this very high-level set-oriented question that the user asks and turn it into an exact recipe for navigating the disk and getting the information from each of the different records within each of the different tables? This process is query optimization: the idea of mapping from the higher-level SQL down to the lower-level recipe or series of actions that you go through to access the data.

Query optimizers have evolved as an enabling technology to make this high-level programming language—this data-access language, SQL—work. Without that, you would end up using brute force: let's look at each row and see if it matches the description of what's asked for. Is it department 50? Is it an order that's over $5,000? That would be very inefficient to scan all of the data all of the time.

So we have access techniques that allow you to look at only a subset of the

a This interview took place just prior to Selinger's retirement from IBM in October 2005.

data, and then you have to plan which of those access techniques makes sense for any given kind of query.

The trick to cost-based query optimization is estimating a cost for each of the different ways of accessing the data, each of the different ways of joining information from multiple tables, and estimating the sizes of results and the savings that you would get by having data in the buffer pools, estimating the number of rows you'll actually touch if you use an index to access the data, and so forth.

The more deeply you can model the cost of accessing the data, the better the choice you'll make for an access path. What we did back in the late 1970s was to invent this cost-based query optimization and provide a model that was good enough for searching a very large space of choices within a reasonable amount of time, then coming up with a very good cost estimate and, therefore, a very good access path.

JH It's amazing that this number of years later, this work remains the dominant approach to relational database system query optimization. Cost-based optimizers have been a real success in technology transfer from research to industry. Can you tell us a little bit about why that was so successful?

PS The quality of cost-based query optimization has really made it possible for people to have relatively hands-free application development. That is, the application developer doesn't have to know a huge amount about the layout of the data on disk and the exact places where the records are and the exact access paths to those records. There's a huge upside from the application productivity standpoint to being able to do really good cost-based query optimization. So, there's a compelling marketplace force for having good cost-based query optimization.

I participated in inventing the System R query optimizer, which was taken lock, stock, and barrel and put into IBM's DB2 relational database product where it has been continually enhanced. Many of the simplifying assumptions to make the problem trackable back in the late 1970s have been eliminated, and the model is now deeper and richer and includes more techniques for accessing the data.

"Outstanding cost-based query optimization is critical to speeding application productivity and lowering total cost of ownership."

It's a growing science, and I think that's part of its success. It has been able to grow and adapt as new inventions come along for accessing the data, or for joining the data in different ways. All of the relational database products have gone to a cost-based query optimization approach and have moved away from a rules-based approach, which was really too inflexible to get you good performance all the time.

The technology still has room to grow—for example, when the data itself behaves differently from what the model assumes. Many optimizers do not model highly correlated data really well. For example, 90210 is a ZIP code that's only in California. ZIP codes are not evenly distributed across states, and there isn't a 90210 in every state of the union. For a user request, nailing down the ZIP code to 90210 is sufficient and applying another predicate, such as state equals California, doesn't change the result. It won't reduce the number of rows because the only 90210 is in California.

JH One of the enemies of industrial query optimizers is complexity, and that can sometimes yield lack of query plan robustness. Small changes in the queries or in the data being queried can lead to substantially different plans. Customers often ask me for a good plan that is stable rather than a near-optimal plan that changes frequently in unpredictable ways. What direction should we be looking to make progress on the optimal query plan versus query plan robustness problem?

PS I think we have to approach it in two ways. One is that you have to be able to execute good plans, and during the execution of a plan you want to notice when the actual data is deviating dramatically from what you expected. If you expected five rows and you've got a million, chances are your plan is not going to do well because you chose it based on the assumption of five. Thus, being able to correct mid-course is an area of enhancement for query optimizers that IBM is pursuing.

Second, you have to continue to deepen the model because you've got to come up with reasonable plans before you can fine-tune them dynamically. Understanding the correlation between rows or between columns in different tables—noting the ZIP code example I gave before—is a very important part of continuing to understand the data more deeply and therefore being able to do even better query optimization.

The fact is that as customers use more and more shrink-wrapped packages or have ad hoc users who haven't gone to SQL school for a year, there's a real need to be able to do good query optimization. You can't have database administrators running into the room, saying, "Don't hit Enter yet. I've got to look at your query to see if it's going to be OK." Outstanding cost-based query optimization is critical to speeding application productivity and lowering total cost of ownership.

JH Let's look for a moment at query optimization and where the technology can be taken beyond database management systems. IBM, and the industry as a whole, have been investing in recent years in auto-tuning and in autonomic computing. Do you see a role for cost-based optimization in this application area?

PS Absolutely. Companies have a lot of data that is quite well structured—an order, a customer, an employee record—but that's maybe 15% of all of the data that a company has. The rest of it is in document files, it's in XML, it's in pictures, it's on Web pages—all of this information also needs to be managed. XML provides a mechanism for being able to do that, but the data isn't quite as regularly structured. It's very dynamic. Every record could look different from the next record even in a related collection of things such as orders or documents.

So you have to have a query language such as XQuery that will be able to navigate and to ask set-oriented questions about this new kind of data. That opens up a different repertoire of data access techniques and requires enhancement of the query optimization process. But I think it's absolutely essential to continue on the path of automatic query optimization rather than put programmers back into the game of understanding exact data structures and doing the navigation in the application program manually. That's simply cost-prohibitive.

JH Looking at new optimization techniques, feedback-directed systems, and dynamic execution time decisions—all significant areas of continuing research—what do you see as the most important next steps looking out, say, five years or so?

PS I think the cost of ownership is on every customer's mind, not just because of the economic downturn that some of them are still in or have just experienced, but because the cost of processors, disk space, and memory are all going down—and the cost of labor is going up.

Furthermore, you have to look at the ratio of how many administrators you need to take care of a terabyte worth of data. Unless you can dramatically improve that ratio, as you accumulate more and more terabytes of data, pretty soon you're looking at employing half the planet to administer it. So we are inventing ways to make administrators capable of handling 20, 100, 1,000 times more data than they do today.

At the same time, we're under pressure to incorporate, search, understand, and take advantage of information that's in this more unstructured form—email, for example. Companies want to be able to look at email or customer service files to give their customers better service, and have to manage and understand and analyze more kinds of information. As we look at what it's going to take to do that, we have to change the game in terms of the cost of organizing, administering, and searching this data.

JH I've seen two pictures painted of the future of unstructured data. One of them has file systems augmented with search appliances, and another is based upon an expanding role of structured stores that are much more flexible and much more capable of dealing with dynamic schema and content. Is there a role for file systems and search appliances? Where do you see this playing out?

PS I don't think that any current or future data storage mechanism will replace all the others. For example, there are many cases where file systems are just fine and that's all you need, and people are perfectly happy with that. We have to be able to reach out to those data sources with a meta-engine that knows how to reach and access all those different data repositories and understands all the different formats—.jpg, .mpg, .doc—and knows how to interpret that data.

The notion of an intergalactic-size, centralized repository is neither reasonable nor practical. You can't just say to a customer, "Put all your data in my repository and I'll solve all your problems." The right answer from my perspective is that customers will have their data in a variety of places under a variety of applications in file systems and database engines. They're not going to centralize it in one kind of data store. That's just not practical. It's not economically feasible

So, file systems will still be around. They may get enhanced with special search techniques as we have more capability and processing power in RAID systems and disk servers, file servers,

"I think the cost of ownership is on every customer's mind, not just because of the economic downturn... but because... the cost of labor is going up."

and so forth, and relational systems will get richer in what they can handle, but we're not going to replace all of the technologies with any one single answer.

JH Do you see content management systems of the future mostly layered on relational database systems, or do you see them as independent stores built using some of what we've learned over the past 30 years of working on relational technologies?

PS I like the architecture in the DB2 content manager, where DB2 is the library server—the card catalogue, so to speak—and it uses some extra semantics in a system-level application surrounding DB2 with some new user-defined types and functions, and stored procedures implementing those applications. Then it has separate resource managers, which are capable of handling a certain class of data types and styles with this kind of document, these kinds of images. They could be physically stored in either the DB2 as the library server or some separate place or file system out on a number of different engines.

It gives you a flexible configuration. You can exploit as much as you like of the functions of DB2—XML, for example—or you can choose to use some of these repository managers. They may be less feature-rich but are expert in a particular kind of information and could be stored locally to where you need that data—particularly if it's massive amounts of data, such as mass spectrometry results. Those are huge files and you want them close to where you're doing the analysis.

JH Given that relational stores now support XML and full-text search, what's missing? Why haven't extended relational systems had a bigger impact in the unstructured world?

PS The semantics of content management go beyond just the data storage parts, the data storage engines, the DB2s of the world. There's a significant set of other abstractions and management techniques that either have to go on top or have to come from a content management system that uses and exploits an extended relational engine but doesn't solely rely on it.

For example, content management systems have the ability to allow Pat access to Chapter 1 of a document, and James access to Chapter 2, and Ed access to Chapters 1 and 2, at the sub-sub-document level. This is something that relational systems don't do today. Similarly, foldering, the idea of document collections that really aren't related to similar structure but are tied to some higher-level semantic content, is beyond what relational systems are undertaking at this point.

"I love the idea of open source. My dream is that this allows many more opportunities for using databases in places where people wouldn't ordinarily go out and buy a database engine."

JH Are there other areas where you see research needed for content managers and relational stores to improve and help customers manage a wider variety of data?

PS If I were choosing today to do research or advanced development, there are a number of areas that are very, very interesting to me. There's continued invention needed in the autonomics. What do you have to do to have a truly hands-free data system that could be embedded in anything? What do you have to do to have truly mass-parallelism at the millions-of-systems (e.g., Internet) level? As commodity hardware becomes smaller and smaller, can we link and talk to systems and compute things on a scale of millions, where today we're at a technology level of thousands? How do you deal with data streams where the queries are fixed and the data is rushing by, and it could be unstructured data? How do you accumulate metadata and keep it up to date? How do you manage it, learn from it, derive information from it?

Searching is still in its first generation. There are lots of opportunities to make search better. If it knew you were angry when you typed in your three keywords to a search engine, would that help it understand what you were searching for? If it knew what email you had just seen before you typed those search keywords, would that help it understand what you were looking for? How can a search engine find what you intended as opposed to what you typed?

How reliable is derived information? There are many sources of unreliability. What if I have a source of information that's right only half the time? How do I rate that information compared with another source who's right all of the time? How do I join together that information, and what's the level of confidence I have in the resulting joined information?

All of those things, as we start dealing with unstructured data and incomplete answers and inexact answers and so forth, are great opportunities for research and advanced development.

JH We've started to see open source having an increasingly large role in server-side computing. Specifically in the database world, we've now got a couple of open source competitors. Is open source a good thing for the database world?

PS I love the idea of open source. I was the manager of the IBM Cloudscape team at the time that we contributed it to Apache, where it has become an incubator project under the name Derby. My dream is that this allows many more opportunities for using databases in places where people wouldn't ordinarily go out and buy a database engine.

So open source can bring the benefits of the reliability, the recoverability, the set-oriented query capabilities to another class of applications—small businesses—and the ability to exploit the wonderful characteristics of database systems across a much richer set of applications. I think it's good for the industry. ■

A previous version of this interview appeared in the April 2005 issue of *ACM Queue*.

DOI:10.1145/1409360.1409374

Want to make your Web site fly? Focus on frontend performance.

BY STEVE SOUDERS

High-Performance Web Sites

Google Maps, Yahoo! Mail, Facebook, MySpace, YouTube, and Amazon are examples of Web sites built to scale. They access petabytes of data sending terabits per second to millions of users worldwide. The magnitude is awe-inspiring.

Users view these large-scale Web sites from a narrower perspective. The typical user has megabytes of data that they download at a few hundred kilobits per second. Users are less interested in the massive number of requests per second being served, caring more about their individual requests. As they use these Web applications they inevitably ask the same question: "Why is this site so slow?"

The answer hinges on where development teams focus their performance improvements. Performance for the sake of scalability is rightly focused on the backend. Database tuning, replicating architectures, customized data caching, and so on, allow Web servers to handle a greater number of requests. This gain in efficiency translates into reductions in hardware costs, data center rack space, and power consumption. But how much does the backend affect the user experience in terms of latency?

The Web applications listed here are some of the most highly tuned in the world, and yet they still take longer to load than we'd like. It almost seems as if the high-speed storage and optimized application code on the backend have little impact on the end user's response time. Therefore, to account for these slowly loading pages we must focus on something other than the backend: we must focus on the *frontend*.

The Importance of Frontend Performance

Figure 1 illustrates the HTTP traffic sent when your browser visits iGoogle with an empty cache. Each HTTP request is represented by a horizontal bar whose position and size are based on when the request began and how long it took. The first HTTP request is for the HTML document (http://www.google.com/ig). As noted in Figure 1, the request for the HTML document took only 9% of the overall page load time. This includes the time for the request to be sent from the browser to the server, for the server to gather all the necessary information on the backend and stitch that together as HTML, and for that HTML to be sent back to the browser.

The other 91% percent is spent on the *frontend*, which includes everything that the HTML document commands the browser to do. A large part of this is fetching resources. For this iGoogle page there are 22 additional HTTP requests: two scripts, one stylesheet, one iframe, and 18 images. Gaps in the HTTP profile (places with no network traffic) are where the browser is parsing CSS, and parsing and executing JavaScript.

The primed cache situation for iGoogle is shown in Figure 2. Here there are only two HTTP requests: one for the HTML document and one for a dynamic script. The gap is even larger because it includes the time to read the cached resources from disk. Even in the primed cache situation, the HTML document accounts for only 17% of the

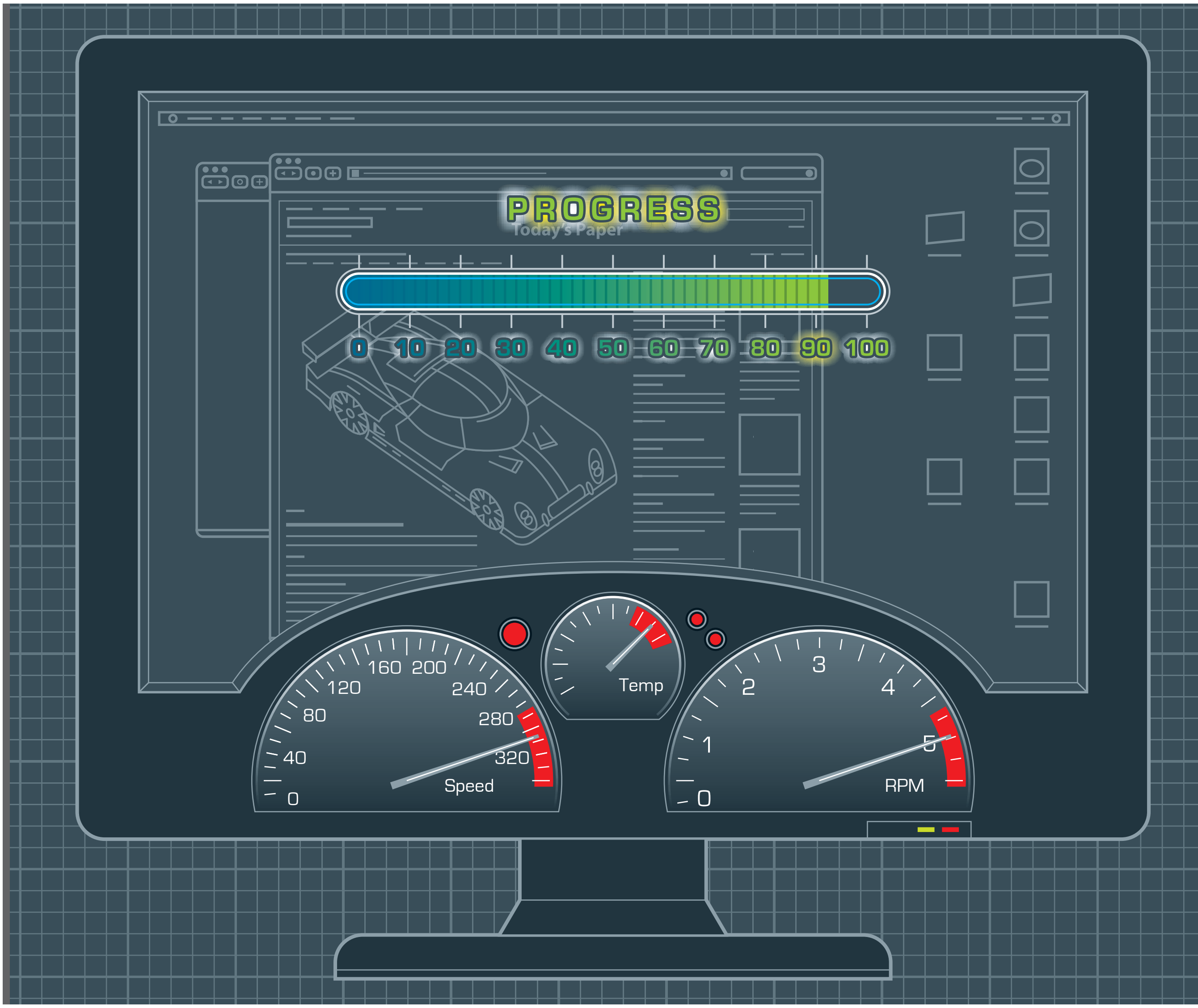


ILLUSTRATION BY NIK SCHULZ

overall page load time.

This situation, in which a large percentage of page load time is spent on the frontend, applies to most Web sites. Table 1 shows that eight of the top 10 Web sites in the U.S. (as listed on Alexa.com) spend less than 20% of the end user's response time fetching the HTML document from the backend. The two exceptions are Google Search and Live Search, which are highly tuned. These two sites download four or fewer resources in the empty cache situation, and only one request with a primed cache.

The time spent generating the HTML document affects overall latency, but for most Web sites this backend time is dwarfed by the amount of time spent on the frontend. If the goal is to make the user experience faster, the place to focus is on the frontend. Given this new focus, the next step is to identify best practices for improving frontend performance.

Frontend Performance Best Practices

Through research and consulting with development teams, I've developed a set of performance improvements that have been proven to speed up Web pages. A big fan of *Harvey Penick's Little Red Book*[1] with advice like "Take Dead Aim," I set out to capture these best practices in a simple list that is easy to remember. The list has evolved to contain the following 14 prioritized rules:

1. Make fewer HTTP requests
2. Use a content delivery network
3. Add an Expires header
4. Gzip components
5. Put stylesheets at the top
6. Put scripts at the bottom
7. Avoid CSS expressions
8. Make JavaScript and CSS external
9. Reduce DNS lookups
10. Minify JavaScript
11. Avoid redirects
12. Remove duplicate scripts
13. Configure ETags
14. Make Ajax cacheable

A detailed explanation of each rule is the basis of my book, *High Performance Web Sites*.[2] What follows is a brief summary of each rule.

Rule 1: Make Fewer HTTP Requests

As the number of resources in the page grows, so does the overall page load time. This is exacerbated by the fact that most browsers only download two resources at a time from a given hostname, as suggested in the HTTP/1.1 specification (http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.1.4).[a] Several techniques exist for reducing the num-

a Newer browsers open more than two connections per hostname including Internet Explorer 8 (six), Firefox 3 (six), Safari 3 (four), and Opera 9 (four).

Figure 1. iGoogle HTTP traffic with an empty cache.

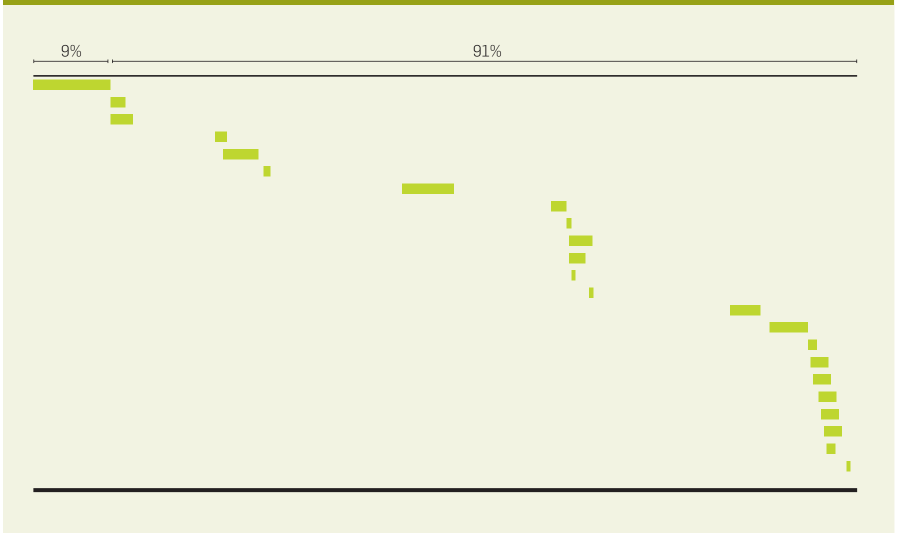


Figure 2. iGoogle HTTP traffic with a primed cache.

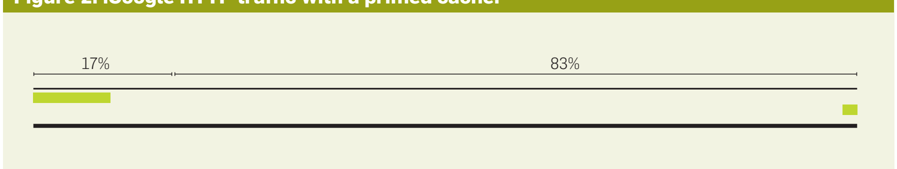


Table 1. Percentage of time spent on the backend.

Web Site	Empty Cache	Primed Cache
http://www.aol.com/	3%	3%
http://www.ebay.com/	5%	19%
http://www.facebook.com/	5%	19%
http://www.google.com/search?q=flowers	53%	100%
http://search.live.com/results.aspx?q=flowers	33%	100%
http://www.msn.com/	2%	6%
http://www.myspace.com/	2%	2%
http://en.wikipedia.org/wiki/Flowers	6%	9%
http://www.yahoo.com/	3%	4%
http://www.youtube.com/	2%	3%

ber of HTTP requests without reducing page content:

- Combine multiple scripts into a single script.
- Combine multiple stylesheets into a single stylesheet.
- Combine multiple CSS background images into a single image called a CSS sprite (see http://alistapart.com/articles/sprites).

Rule 2: Use a Content Delivery Network

A content delivery network (CDN) is a collection of distributed Web servers used to deliver content to users more efficiently. Examples include Akamai Technologies, Limelight Networks, SAVVIS, and Panther Express. The main performance advantage provided by a CDN is delivering static resources from a server that is geographically closer to the end user. Other benefits include backups, caching, and the ability to better absorb traffic spikes.

Rule 3: Add an Expires Header

When a user visits a Web page, the browser downloads and caches the page's resources. The next time the user visits the page, the browser checks to see if any of the resources can be served from its cache, avoiding time-consuming HTTP requests. The browser bases its decision on the resource's expiration date. If there is an expiration date, and that date is in the future, then the resource is read from disk. If there is no expiration date, or that date is in the past, the browser issues a costly HTTP request. Web developers can attain this performance gain by specifying an explicit expiration date in the future. This is done with the Expires HTTP response header, such as the following:

```
Expires: Thu, 1 Jan 2015 20:00:00 GMT
```

Rule 4: Gzip Components

The amount of data transferred over the network affects response times, especially for users with slow network connections. For decades developers have used compression to reduce the size of files. This same technique can be used for reducing the size of data sent over the Internet. Many Web servers and Web hosting services enable compression of HTML documents by default, but compression shouldn't stop there. Developers should also compress other types of text responses, such as scripts, stylesheets, XML, JSON, among others. Gzip is the most popular compression technique. It typically reduces data sizes by 70%.

Rule 5: Put Stylesheets at the Top

Stylesheets inform the browser how to format elements in the page. If stylesheets are included lower in the page, the question arises: What should the browser do with elements that it can render before the stylesheet has been downloaded?

One answer, used by Internet Explorer, is to delay rendering elements in the page until all stylesheets are downloaded. But this causes the page to appear blank for a longer period of time, giving users the impression that the page is slow. Another answer, used by Firefox, is to render page elements and redraw them later if the stylesheet changes the initial formatting. This causes elements in the page to "flash" when they're redrawn, which is disruptive to the user. The best answer is to avoid including stylesheets lower in the page, and instead load them in the HEAD of the document.

Rule 6: Put Scripts at the Bottom

External scripts (typically, ".js" files) have a bigger impact on performance than other resources for two reasons. First, once a browser starts downloading a script it won't start any other parallel downloads. Second, the browser won't render any elements below a script until the script has finished downloading. Both of these impacts are felt when scripts are placed near the top of the page, such as in the HEAD section. Other resources in the page (such as images) are delayed from being downloaded and elements in the page that already exist (such as the HTML text in the document itself) aren't displayed until the earlier scripts are done. Moving scripts lower in the page avoids these problems.

Rule 7: Avoid CSS Expressions

CSS expressions are a way to set CSS properties dynamically in Internet Explorer. They enable setting a style's property based on the result of executing JavaScript code embedded within the style declaration. The issue with CSS expressions is that they are evaluated more frequently than one might expect—potentially thousands of times during a single page load. If the JavaScript code is inefficient it can cause the page to load more slowly.

Rule 8: Make JavaScript and CSS External

JavaScript can be added to a page as an inline script:

```
<script type="text/javascript">
var foo="bar";
</script>
```

or as an external script:

```
<script src="foo.js" type="text/javascript"></script>
```

Similarly, CSS is included as either an inline style block or an external stylesheet. Which is better from a performance perspective?

HTML documents typically are not cached because their content is constantly changing. JavaScript and CSS are less dynamic, often not changing for weeks or months. Inlining JavaScript and CSS results in the same bytes (that haven't changed) being downloaded on every page view. This has a negative impact on response times and increases the bandwidth used from your data center. For most Web sites, it's better to serve JavaScript and CSS via external files, while making them cacheable with a far future Expires header as explained in Rule 3.

Rule 9: Reduce DNS Lookups

The Domain Name System (DNS) is like a phone book: it maps a hostname to an IP address. Hostnames are easier for humans to understand, but the IP address is what browsers need to establish a connection to the Web server. Every hostname that's used in a Web page must be resolved using DNS. These DNS lookups carry a cost; they can take 20–100 milliseconds each. Therefore, it's best to reduce the number of unique hostnames used in a Web page.

Rule 10: Minify JavaScript

As described in Rule 4, compression is the best way to reduce the size of text files transferred over the Internet. The size of JavaScript can be further reduced by minifying the code. Minification is the process of stripping unneeded characters (comments, tabs, new lines, extra white space, and so on) from the code. Minification typically reduces the size of JavaScript by 20%. External scripts should be minified, but inline scripts also benefit from this size reduction.

Rule 11: Avoid Redirects

Redirects are used to map users from one URL to another. They're easy to implement and useful when the true URL is too long or complicated for users to remember, or if a URL has changed. The downside is that redirects insert an extra HTTP roundtrip between the user and her content. In many cases, redirects can be avoided with some additional work. If a redirect is truly necessary, make sure to issue it with a far future Expires header (see Rule 3), so that on future visits the user can avoid this delay.[b]

Rule 12: Remove Duplicate Scripts

If an external script is included multiple times in a page, the browser has to parse and execute the same code multiple times. In some cases the browser will request the file multiple times. This is inefficient and causes the page to load more slowly. This obvious mistake would seem uncommon, but in a review of U.S. Web sites it could be found in two of the top 10 sites. Web sites that have a large number of scripts and a large number of developers are most likely to suffer from this problem.

Rule 13: Configure ETags

Entity tags (ETags) are a mechanism used by Web clients and servers to verify that a cached resource is valid. In other words, does the resource (image, script, stylesheet, among others) in the browser's cache match the one on the server? If so, rather than transmitting the entire file (again), the server simply returns a 304 Not Modified status telling the browser to use its locally cached copy. In HTTP/1.0, validity checks were based on a resource's Last-Modified date: if the date of the cached file matched the file on the server, then the validation succeeded. ETags were introduced in HTTP/1.1 to allow for validation schemes based on other information, such as version number and checksum.

ETags don't come without a cost. They add extra headers to HTTP requests and responses. The default ETag syntax used in Apache and IIS makes it likely that the validation will erroneously fail if the Web site is hosted on multiple servers. These costs impact performance, making pages slower and increasing the load on Web servers. This is an unnecessary loss of performance, because most Web sites don't take advantage of the advanced features of ETags, relying instead on the Last-Modified date as the means of validation. By default, ETags are enabled in popular Web servers (including Apache and IIS). If your Web site doesn't utilize ETags, it's best to turn them off in your Web server. In Apache, this is done by simply adding "FileETag none" to your configuration file.

Rule 14: Make Ajax Cacheable

Many popular Web sites are moving to Web 2.0 and have begun incorporating Ajax. Ajax requests involve fetching data that is often dynamic, personalized, or both. In the Web 1.0 world, this data is served by the user going to a specified URL and getting back an HTML docu-

b Caching redirects is not supported in some browsers.

Table 2. Percentage of unused JavaScript functions.

Web Site	JavaScript Size	Unused Functions
http://www.aol.com/	115K	70%
http://www.ebay.com/	183K	56%
http://www.facebook.com/	1088K	81%
http://www.google.com/search?q=flowers	15K	55%
http://search.live.com/results.aspx?q=flowers	17K	76%
http://www.msn.com/	131K	69%
http://www.myspace.com/	297K	82%
http://en.wikipedia.org/wiki/Flowers	114K	68%
http://www.yahoo.com/	321K	87%
http://www.youtube.com/	240%	82%
average	**252K**	**74%**

ment. Because the HTML document's URL is fixed (bookmarked, linked to, and so on), it's necessary to ensure the response is not cached by the browser.

This is not the case for Ajax responses. The URL of the Ajax request is included inside the HTML document; it's not bookmarked or linked to. Developers have the freedom to change the Ajax request's URL when they generate the page. This allows developers to make Ajax responses cacheable. If an updated version of the Ajax data is available, the cached version is avoided by adding a dynamic variable to the Ajax URL. For example, an Ajax request for the user's address book could include the time it was last edited as a parameter in the URL, "&edit=1218050433." As long as the user hasn't edited their address book, the previously cached Ajax response can continue to be used, making for a faster page.

Performance Analysis with YSlow

Evangelizing these performance best practices is a challenge. I was able to share this information through conferences, training classes, consulting, and documentation. Even with the knowledge in hand, it would still take hours of loading pages in a packet sniffer and reading HTML to identify the appropriate set of performance improvements. A better alternative would be to codify this expertise in a tool that anyone could run, reducing the learning curve and increasing adoption of these performance best practices. This was the inspiration for YSlow.

YSlow (http://developer.yahoo.com/yslow/) is a performance analysis tool that answers the question posed in the introduction: "Why is this site so slow?" I created YSlow so that any Web developer could quickly and easily apply the performance rules to their site, and find out specifically what needed to be improved. It runs inside Firefox as an extension to Firebug (http://getfirebug.com/), the tool of choice for many Web developers.

The screenshot in Figure 3 shows Firefox with iGoogle loaded. Firebug is open in the lower portion of the window, with tabs for Console, HTML, CSS, Script, DOM, and Net. When YSlow is installed, the YSlow tab is added. Clicking YSlow's Performance button initiates an analysis of the page against the set of rules, resulting in a weighted score for the page.

As shown in Figure 3, YSlow explains each rule's results with details about what to fix. Each rule in the YSlow screen is a link to the companion Web site, where additional information about the rule is available.

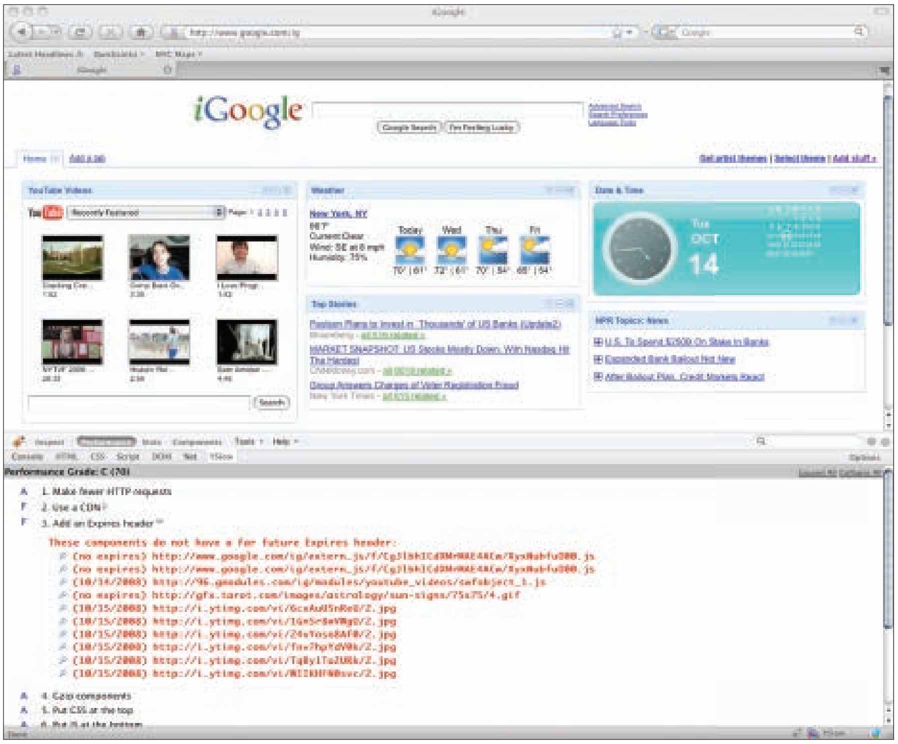


Figure 3: YSlow.

The Next Performance Challenge: JavaScript

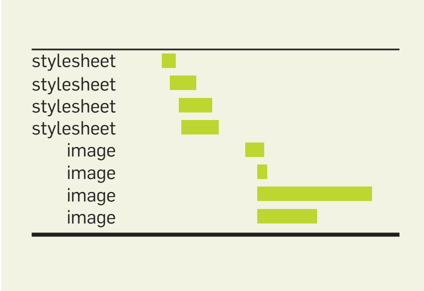
Web 2.0 promises a future where developers can build Web applications that provide an experience similar to desktop apps. Web 2.0 apps are built using JavaScript, which presents significant performance challenges because JavaScript blocks downloads and rendering in the browser. To build faster Web 2.0 apps, developers should address these performance issues using the following guidelines:

- Split the initial payload
- Load scripts without blocking
- Don't scatter scripts

Split the Initial Payload

Web 2.0 apps involve just a single

Figure 4. Stylesheet followed by inline script in www.msn.com/.



page load. Instead of loading more pages for each action or piece of information requested by the user, as was done in Web 1.0, Web 2.0 apps use Ajax to make HTTP requests behind the scenes and update the user interface appropriately. This means that some of the JavaScript that is downloaded is not used immediately, but instead is there to provide functionality that the user might need in the future. The problem is that this subset of JavaScript blocks other content that *is* used immediately, delaying immediate content for the sake of future functionality that may never be used.

Table 2 shows that for the top 10 U.S. Web sites, an average of 74% of the functionality downloaded is not used immediately. To take advantage of this opportunity, Web developers should split their JavaScript payload into two scripts: the code that's used immediately (~26%) and the code for additional functionality (~74%). The first script should be downloaded just as it is today, but given its reduced size the initial page will load more quickly. The second script should be *lazy-loaded*, which means that after the initial page is completely rendered this second script is downloaded dynamically, using one of the techniques listed in the next section.

Load Scripts without Blocking

As described in "Rule 6: Put Scripts at the Bottom," external scripts block the download and rendering of other content in the page. This is true when the script is loaded in the typical way:

```
<script src="foo.js" type="text/javascript"></script>
```

But there are several techniques for downloading scripts that avoid this blocking behavior:

- XHR eval
- XHR injection
- script in iframe
- script DOM element
- script defer
- document.write script tag

You can see these techniques illustrated in Cuzillion (http://stevesouders.com/cuzillion/), but as an example let's look at the *script DOM element approach*:

```
<script type="text/javascript">
var se = document.createElement('script');
se.src = 'http://anydomain.com/foo.js';
document.getElementsByTagName('head')[0].appendChild(se);
</script>
```

A new DOM element is created that is a script. The src attribute is set to the URL of the script. Appending it to the head of the document causes the script to be downloaded, parsed, and executed. When scripts are loaded this way, they don't block the downloading and rendering of other content in the page.

Don't Scatter Inline Scripts

These first two best practices about JavaScript performance have to do with external scripts. Inline scripts also impact performance, occasionally in significant and unexpected ways. The most important guideline with regard to inline scripts is to avoid a stylesheet followed by an inline script.

Figure 4 shows some of the HTTP traffic for http://www.msn.com/. We see that four stylesheet requests are downloaded in parallel, then there is a white gap, after which four images are downloaded, also in parallel with each other. But why aren't all eight downloaded in parallel?[c]

This page contains an inline script after the fourth stylesheet. Moving this inline script to either above the stylesheets or after the images would result in all eight requests taking place in parallel, cutting the overall download time in half. Instead, the images are blocked from downloading until the inline script is executed, and the inline script is blocked from executing until the stylesheets finish downloading. This seems like it would be a rare problem, but it afflicts four of the top ten sites in the U.S: eBay, MSN, MySpace, and Wikipedia.

c Note that these requests are made on different hostnames and thus are not constrained by the two-connections-per-server restriction of some browsers, as described in "Rule 1: Make Fewer HTTP Requests."

Life's Too Short, Write Fast Code

At this point, I hope you're hooked on building high-performance Web sites. I've explained why fast sites are important, where to focus your performance efforts, specific best practices to follow for making your site faster, and a tool you can use to find out what to fix. But what happens tomorrow, when you're back at work facing a long task list and being pushed to add more features instead of improving performance? It's important to take a step back and see how performance fits into the bigger picture.

Speed is a factor that can be used for competitive advantage. Better features and a more appealing user interface are also distinguishing factors. It doesn't have to be one or the other. The point of sharing these performance best practices is so we can all build Web sites to be as fast as they possibly can—whether they're barebones or feature rich.

I tell developers "Life's too short, write fast code!" This can be interpreted two ways. Writing code that executes quickly saves time for our users. For large-scale Web sites, the savings add up to lifetimes of user activity. The other interpretation appeals to the sense of pride we have in our work. Fast code is a badge of honor for developers.

Performance must be a consideration intrinsic to Web development. The performance best practices described here are proven to work. If you want to make your Web site faster, focus on the frontend, run YSlow, and apply these rules. Who knows, fast might become your site's most popular feature.

References

1. Penick, H. *Harvey Penick's Little Red Book: Lessons and Teachings From A Lifetime In Golf.* Simon and Schuster, 1992.
2. Souders, S. *High Performance Web Sites: Essential Knowledge for Front-End Engineers.* O'Reilly, 2006.

Steve Souders (http://stevesouders.com) works at Google on Web performance and open source initiatives. He is the author of *High Performance Web Sites* and the creator of YSlow, Cuzillion, and Hammerhead. He teaches at Stanford and is the co-founder of the Firebug Working Group.

DOI:10.1145/1409360.1409375

When it comes to virtualization platforms, experts say focus first on the services to be delivered.

BY MACHE CREEGER

CTO Virtualization Roundtable, Part II

Last month we published Part I of a CTO Roundtable forum on virtualization. Sponsored by the ACM Professions Board, the roundtable features five experts on virtualization discussing the current state of the technology and how companies can use it most effectively. In this second and final installment, the participants address key issues such as choosing the most appropriate virtual machine platform, using virtualization to streamline desktop delivery, and using virtualization as an effective disaster-recovery mechanism.

Participants:

Mache Creeger (Moderator): Creeger is a longtime technology industry veteran based in Silicon Valley. Along with being an *ACM Queue* columnist, he is the principal of Emergent Technology Associates, marketing and business development consultants to technology companies worldwide.

Tom Bishop is CTO of BMC Software. Prior to BMC, Bishop worked at Tivoli, both before and after its initial public offering and acquisition by IBM, and also at Tandem Computers. Earlier in his career Bishop spent 12 years at Bell Labs' Naperville, IL facility and then worked for UNIX International. He graduated from Cornell University with both bachelor's and master's degrees in computer science.

Simon Crosby is the CTO of the Virtualization Management Division at Citrix. He was one of the founders of XenSource and was on the faculty of Cambridge University, where he earned his Ph.D. in computer science. Crosby grew up in South Africa and has master degrees in applied probability and computer science.

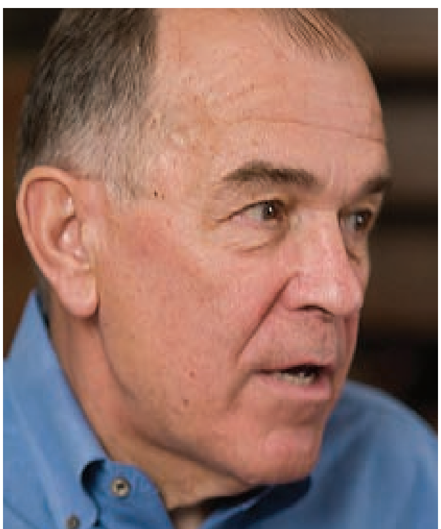
Gustav. This is a pseudonym due to the policies of his employer, a large financial services company where he runs distributed systems. Early in his career, Gustav wrote assembler code for telephone switches and did CAD/

CAM work on the NASA space station *Freedom*. He later moved over to large system design while working on a government contract and subsequently worked for a messaging and security startup company in Silicon Valley, taking it public in the mid-1990s. After starting his own consulting firm, he began working at his first large financial firm. Seven or eight years later, he landed at his current company.

Allen Stewart is a Principle Program Manager Lead in the Window Server Division at Microsoft. He began his career working on Unix and Windows operating systems as a system programmer and then moved on to IBM, where he worked on Windows systems integration on Wall Street. After IBM, Stewart joined Microsoft, where for the first six years he worked as an architect in the newly formed Financial Services Group. He then moved into the Windows Server Division Engineering organization to work on Windows Server releases. His primary focus is virtualization technologies: hardware virtualization, virtualization management, and application virtualization. Stewart is a Microsoft Certified Architect and is on the Board of Directors of the Microsoft Certified Architect Program.

Steve Herrod is the CTO of VMware, where he's worked for seven years. Before VMware, Herrod worked in Texas for companies such as EDS and Bell Northern Research. Earlier in his career Herrod attended school with Mendel Rosenblum, the founder of VMware, and then worked for TransMeta, a computer hardware and software emulation company.

Steve Bourne is chair of the ACM Professions Board. He is a former president of ACM and Editor-in-Chief of the *ACM Queue* editorial advisory board. A fellow alumnus with Simon Crosby, Bourne received his Ph.D. from Trinity College, Cambridge. Bourne held management roles at Cisco, Sun, DEC, and SGI and currently is CTO at El Dorado Ventures, where he advises the firm on their technology investments.



Steve Bourne

Mache Creeger

Allen Stewart

STEVE BOURNE: So I'm in this small- to medium-size business (SMB) shop. You have just told me that I have to balance out the disk and the network with my CPUs. This is all very complicated. What am I going to do next year?

GUSTAV: If you want to implement high up in the service stack today, you should choose VMware. It's the one vendor that sells a fully integrated solution. If you're an SMB with 20 people, want maximum flexibility, and want a single-vendor solution, it's VMware. Because to Simon's earlier point, right now they're trying to sell cars; they are not trying to sell engines.

SIMON CROSBY: No, SMBs should choose Citrix XenServer, HP ProLiant Select Edition, which is an entirely HP-branded product. It is an integrated virtualization solution that is part of ProLiant Server, entirely packaged and managed by HP VMM (Virtual Machine Manager), which manages Microsoft, VMware, and XenServer today. It's got the bundled HP toolset included, our Xen technology built in, and is one of HP's embedded hypervisors. It's the perfect mid-market product.

Virtualization today is not a real "market" and will not be until there are multiple independent, economically successful vendors. There is one very successful vendor today, and my hat's off to VMware. However, its success is equivalent to the TCP/IP stack vendors of the early 1990s, before the stack became a commodity. But things are about to change because until now nobody else has played. The change is that the core value proposition is about to become free.

With Microsoft's Hyper-V hypervisor virtual machine (VM) platform currently at $28 and my company's Xen hypervisor being free, the price of a hypervisor is heading toward free. If you look at HP's embedded hypervisor offering using our product, it is an incredible value proposition. That same product has more functionality than what made VMware their first $500 million of revenue. While VMware has had the benefit of market lead and brand presence, HP has knocked the value proposition out of the park. Is Citrix with XenServer an independent viable competitor against VMware? Yes, but that's a tough slog. Enabling companies to create alterative products like Citrix XenServer, HP ProLiant Select Edition greatly expands customer choice for a wide range of market needs.

STEVE HERROD: If you're an SMB with one thousand employees or less—and 70% of our customers are what we define to be SMBs—you don't care if it's Xen, VMware, or Microsoft. You want simplicity, availability, security, and you want something that can be supported by your staff.

ALLEN STEWART: If you're an SMB and already running Windows Servers 2008, you enable Hyper-V, use the same management tools that you have been using, and depend on the management construct to help you beyond the virtualization platform.

In the SMB market, Microsoft has pushed System Center outside of the enterprise to System Center Essentials. If you have a small amount of servers, buy Essentials, and you can inexpensively manage the platform.

existing player. They challenge the OS guys because virtualization separates them from the hardware; the storage guys because storage management for virtualization is done on the host and that threatens the whole Symantec/VERITAS model; and the management players, too.

VMware confronts a lot of entrenched interests and threatens them. So VMware could end up as a systems management play, a storage management play, or a big brain that manages the future data center—and that would threaten Cisco or their competitors. The interesting thing for VMware is where does it go from here? Every step they take threatens an established ven-

us or other large enterprises today, but may at some future date. These are the real bleeding edge, radically thinking folks. One of the things we saw was that people are putting management in the software deployment layer on top of EC2.[a]

My advice for SMBs able to tolerate offsite data processing is that management options, possibly from third parties, will be available in the not-too-distant future for EC2 and other cloud models and provide management flexibility similar to solutions from VMware, Veridian, and Xen. Even in the cloud, where you literally care about nothing, third-party vendors will come in to provide common abstractions.


Tom Bishop


Simon Crosby


Steve Herrod

TOM BISHOP: I think that's right. I think you start with what you know, stick with the vendors you know and the technology you know, and it's going to allow you to get the biggest bang for the least cost.

MACHE CREEGER: If you run lots of Oracle would you work backward from Oracle and ask what would work best?

GUSTAV: No, I'd work backward from the OS level you manage because that's really what you're managing. But back to your point of "I'm not worried that there are three hypervisor vendors," companies should worry less about that because hypervisors from all the vendors are slowly but surely providing the same functionality.

SIMON CROSBY: But then it's about virtualization management. As the market leader in a new category, everything that VMware does challenges an

dor in an existing market sector.

TOM BISHOP: SMB players that purchase management software will get it from the virtualization vendors and the rest will to do it by hand, which is what they've always done. I am not saying that their management functions aren't important. It's just that the problems to be solved for the SMB market are not big enough, hard enough, and expensive enough for management companies to address.

MACHE CREEGER: Well, what happens to the management business when management companies cede it to virtualization companies on the SMB side and alternatively get squeezed by offerings from the cloud?

GUSTAV: It's a counter-trend. One of the things we do at our CTO event in California is to bring in early-stage companies that have little chance to sell to

As a classic example, one of the things that I'm most interested in virtualizing right now is the desktop. I might actually use Citrix on top of Xen or VMware, or CXD on top of VMware to do that particular function.

The Citrix technology is much better for the presentation layer of virtualization. At the present state of technology, I find VMware's framework for doing physical–to-virtual migrations and similar functions to be better. In addition to that, I may do an application virtualization layer with a Softgrid-like technology.[b]

I might use all three major vendors, depending on their strengths, and

a Amazon's cloud product offering—http://aws.amazon.com/ec2/.

b http://en.wikipedia.org/wiki/Microsoft_Application_Virtualization/.

PHOTOGRAPHS BY JASON GARDNER

match to our needs to create a single integrated desktop—DaaS, or Desktop as a Service: one vendor for the presentation layer, a different one for application virtualization, and a third one for the hypervisor.

The more interesting point is that from an IT management point of view, the hypervisor is getting less and less interesting. Worry less about where you get your hypervisor from and more about where you get your management from. Ask whether the management you need today can be sufficiently provided by your vendor, whether that is the hardware partner, a third-party, or the direct vendor.

Is the Microsoft hypervisor going to perform slightly better with small packet operations than the other competitors? Maybe, but that is just for this release; it's ephemeral. By the time you've installed it, the competitive matrix has changed, so realistically you don't care. It all comes down to the fact that management technologies change slower than hypervisor technologies.

MACHE CREEGER: It's like the TPC[c] wars of the early 1990s. Vendors would jockey back and forth after every release, but at some point customers realized that they could not pick a vendor based on who was ahead at any given moment.

It was mentioned that some companies were putting management tools together over the cloud, and that Microsoft is developing a multi-hypervisor management console.

ALLEN STEWART: Microsoft is an established system-management company and look at managing systems holistically. Initially we focused on the workload and were moving the VM based on the performance of the VM. Now we're looking at the workload that's running in the VM and making decisions based on that.

TOM BISHOP: This is the hard part. You want to make the decision based on application behavior, not on VM behavior.

MACHE CREEGER: You want service-level agreements (SLAs).

TOM BISHOP: The problem is that by and large SLAs are not available today.

ALLEN STEWART: And that is totally our focus in the system-management space.

c http://www.tpc.org/.

“

SIMON CROSBY

Virtualization is a feature set, not an objective. It's a technology that we should look at in the same way as compilers or TCP/IP stacks. It's a passing fad. The real benefits will come out of the overall ability to compose and manage an application throughout its life cycle.

”

TOM BISHOP: All you're going to do is change the problem. Will everybody build all of their applications using Microsoft tools? No. All we've done is change the context in which we address the SLA.

ALLEN STEWART: Actually in the Microsoft Systems Center world, we don't require you to do that any more. We do require that you have some knowledge about the actual application, and ISVs are building in that knowledge. Once you have that knowledge, you can then make decisions based on that knowledge.

MACHE CREEGER: So should we expect that over time vendors will define standards around instrumentation for service-level responsiveness, but that it's going to take a long time to get there?

SIMON CROSBY: I don't think so. Somebody tell me a metric that everybody cares about. Somebody tell me what this means.

TOM BISHOP: It's capacity, throughput, and response level.

ALLEN STEWART: And one of the ways you do that is by standardizing higher up in the stack. When developers are building these applications, this SLA model is composed with the application.

STEVE BOURNE: One of the things that I have heard from the NANOG[d] guys (North American Network Operators Group) is that you are nuts if you're running your desktop in a non-virtual-machine environment and visiting random Web sites. So my question is do you see security on the desktop as a model?

SIMON CROSBY: Yes. There are two layers of virtualization that are useful there. One is the isolation between applications and OSs, where applications are streamed to desktops. The other is having separate VMs for different contexts—a VM for a user's personal context, which can be thrown away and restarted again, and another VM for their corporate work.

People like me want applications to work on an airplane. Another category of user is the task worker. I think there's a ton of different technologies that could provide viable solutions but I think it's too early to comprehensively understand which ones apply to specific user categories.

d http://www.nanog.org/.

GUSTAV: I think you will see the browser itself evolve into a VM architecture. Ultimately the browser will offer the option of either resetting or keeping state.

SIMON CROSBY: That's absolutely wrong. If your browser is attacked and the OS is compromised you're done for.

GUSTAV: What I'm suggesting is that the browser captures the changes made during the session and, post session, gives the user the option of making those changes go away. This amounts to having an embedded hypervisor in the browser and presenting the user with the option of maintaining or erasing state upon exit.

SIMON CROSBY: And you know what? It wrote to the hard disk. No matter what that application does, I will go to the hard disk and find it. This is one of the first security flaws Amazon found with EC2. Reset at the application level is ineffective, because if I can get to the hard disk, I will find stuff anyway. People see that information goes to the hard disk and will look to see what is there.

Amazon thought they solved it in EC2 by writing to a virtual hard disk, but it's actually stored on some spinning plate of aluminum. The next time I go into the EC2 virtual machine, I can go and search through that virtual hard disk and I will find proprietary information. Resetting at the application level is not going to help. You really do need to think about security throughout the entire architectural stack.

Application-layer virtualization does provide some help. We have an isolation layer along with VMware and Microsoft. Because the application is not installed in the OS it is invisible to the registry and the file system. As a result, changes made by the application do not reach the layer below.

GUSTAV: I actually wasn't saying resetting at the application level. I was saying that that a hypervisor will be embedded in the binary for the browser that you run.

SIMON CROSBY: But even that wouldn't satisfy the guys at the NSA who want you to go and write zeroes to every sector on every disk. It won't solve the problem, which is that you actually wrote real blocks of storage to some real disk somewhere.

TOM BISHOP: Probably the most innovative solution I've ever seen is from the Lower Colorado River Authority[e] (LCRA). They are an organization based in Austin, TX that manages dams. The way they solve this problem is when you come into work in the morning they give you a laptop that has all the applications you want in a base disk image. You may do anything you want during working hours, and at the end of the day you give the laptop back. Overnight the disk is wiped and a new disk image is blasted back onto the laptop. The next day, you come in and start over with a new base image.

SIMON CROSBY: At Citrix we have a model within Xen Desktop where all VMs boot off the same OS golden image and all have the same base applications. To deliver a user-specific model, user-specific applications are streamed into the VM based on the user's roaming profile. This approach minimizes the number of OS images and VMs that need to be stored. Anything that's written to disk by an executing VM is cached locally in the VM and never written back to the hard drive, and all changes are discarded on every reboot. For certain classes of users, such as call center operators, this approach works very well.

TOM BISHOP: The only state that persists is well defined through the set of applications.

SIMON CROSBY: That's right.

STEVE BOURNE: Should IT managers care about people who are accessing the Internet through desktops in their shop? Should they be considering VMs to protect the internal networks of their organizations?

MACHE CREEGER: Virtualization introduces too much complexity to effectively encapsulate all the operating restrictions on a general desktop, because at the end of the day, general desktops are still about applications, writing to the disk, and network transmission to other intelligent entities. Virtualization is just another layer of abstraction; it doesn't change the functional levels at which problems occur.

GUSTAV: Several vendors have streaming desktop products that allow a desktop to be streamed from a server to a client machine. The desktop can be cached—on a USB key, for example—or not cached at all. Desktop streaming is useful when I want a client machine to be my desktop for now, but afterward I never want to use it again.

One place you might use this is where you want zero footprint. This would include cases where what you have is known to be good but you want to run it on an environment known to be suspect, such as at an airport kiosk or on people's home machines.

MACHE CREEGER: Looking at the example that Simon suggested earlier, can we define sessions in desktop environments so that at some point you can throw everything away and reauthorize the session with a complete blank slate? Wouldn't that solve a lot of security issues?

TOM BISHOP: Yes, but not independent of the application.

SIMON CROSBY: The key question is whether the virtual hard disk itself is stateful or not. Where does the state that I want to keep live? Is it part of the thing that boots?

GUSTAV: Is it persistent state or is it transitory/disposable state?

SIMON CROSBY: Where does my persistent state live and where does the transient state live?

MACHE CREEGER: You have to define "session" and that's a hard thing to define.

TOM BISHOP: Because it varies from application to application.

SIMON CROSBY: And from user category to user category. In my world, I have VMs on my laptop and each of my VMs is independently snapshotted and stored in S3.[f] However, the VMs are simply runtime entities. My personal and work data are held separately, mapped into the runtime upon boot, and independently backed up, block for block onto S3. If I lose my laptop on any day, the hard disk is locked and the machine is of no use to anyone else. I purchase a new laptop, and within download time everything I have is back.

I also use Citrix WAN optimization technology to ensure that no block of data ever gets sent over the wire twice. A 24MB Powerpoint file with just a few changes takes less than a second to back up because 99% of the blocks are already backed up and only the differences are sent over the wire.

e http://www.lcra.org/.

f Amazon's Simple Storage Service—http://aws.amazon.com/s3/.

GUSTAV: There's actually a really powerful application that comes with this. Along with day-to-day virtualization stuff is the issue of disaster recovery (DR). Most SMBs make zero investment in DR. Virtualization becomes incredibly cost effective when it has the ability to send VMs to the cloud for access only when needed.

SIMON CROSBY: The benefits are huge and the numbers are very compelling.

GUSTAV: Typical disaster-recovery costs are 2N (twice the cost of the infrastructure). To say that I can go to 1.05N is game changing.

SIMON CROSBY: The great thing about this kind of approach is that the cloud vendor can lose a data center and my data is still there. They can lose two simultaneously and my data is still there.

MACHE CREEGER: The virtualization abstraction enables fungible data-center capacity, much like the power industry, where people can trade excess capacity on the open market.

SIMON CROSBY: That's right, and like the power industry you will have purely financial players, people in the business who know nothing about technology, simply trading capacity back and forth. The first arbitrage players on the cloud are already in business.

GUSTAV: I will take it back to the insurance space. I can buy true insurance. I can pay 2% of the value of my assets today and know I can absolutely run my exact stuff.

MACHE CREEGER: So it's a bulletproof insurance premium.

TOM BISHOP: That's right. It's how you compute and manage risk.

GUSTAV: It's "How do I take my 2N problem down to 0.02N?" It's "How do I take 98% of my DR cost to zero?" That is just a different way of saying "How do I take 49% of my total IT cost to zero?"

SIMON CROSBY: At the same time, the high-end fault tolerance (FT) moves down to a commoditized, value-priced capability rather than a high-end, hardware capability.

GUSTAV: To give you an example of the thinking behind DR, take 9/11. 9/11 was a black swan; it never should have happened. Any statistical model that you build fails when the black swan shows up, and DR is only valuable if it actually works when the black swan shows up.

You are actually building a model that goes past the black swan. The thing about 9/11 that made it even more chaotic than the tragedy of the Trade Center towers coming down was that 12 Broad Street (the lower Manhattan telecom switching station) filled with water. This resulted in no teleco for the southern tip of Manhattan, creating a black swan.

Theoretically the thing that could never happen, which is that every divergent teleco path in southern Manhattan becomes blocked, happened. Many of the problems that are solved in the typical case are not sufficient in the DR case because the normal constraints do not apply.

TOM BISHOP: The number-one conclusion at this one event I attended was that during Hurricane Katrina every company's disaster-recovery plan assumed people could get to work. Every disaster-recovery plan in New Orleans failed because people could not get to work.

SIMON CROSBY: 9/11 was about mortality. Nobody reasons about how to recover from mortal events. At the end of the day, the rational guy in the SMB doesn't deal with that level of risk. If an event like that happens, his business is lost.

GUSTAV: There actually are levels of defined risk. You've got systemic risk. If the counterparty doesn't show up, the entire market cannot function. That's one level of badness. But think about the SMB. There's a stat I've seen recently that says that 70% of businesses that are forced to close for more than a month never reopen. Systematic risk, well priced, is more valuable to the SMB than it is to a large enterprise like my employer.

The counterargument made earlier states that if this business fails, it is cheaper to start a new business than to pay 2N for 10 years. The problem is that we have never been able to present a reasonably priced alternative.

Effectively the SMB owner is self-insured and betting on his own ability. I would say that DR for the SMB is actually a richer market than DR for the enterprise. I think part of the problem is defining the minimum requirement. It doesn't need to be up in the next five minutes; he just needs to know that he can get it working in two or three days under any circumstances.

TOM BISHOP: One of the things I learned at Bell Labs was that in terms of fault coverage, you got far better results by recovering from failure than you ever got by avoiding it.

SIMON CROSBY: That is right. A recent Stanford research model tells us to assume that computer systems are inherently fragile, humans build bad software, and applications are going to decay and fail. Therefore we should architect our applications so they inherently contain the concept of failure and restart.

MACHE CREEGER: We are almost out of time here. I'd like you all to summarize what the takeaways are and what kind of advice you're going to give to the poor person who's trying to make sense of the world today and how he can move forward.

STEVE HERROD: At the highest level, I think we should all avoid breathing our own exhaust too much. At the end of the day, virtualization is a tool. The goals are to make life better, and particularly for SMBs, to make computing simpler. To make it easy for SMBs is to enable them to operate highly available and securely, and to solve their business problems with their applications.

It is actually about manageability and how to do more and make things run better with less staff. When you're evaluating your workload and products to address it, you should be looking at the overall story, not just at a snapshot. It's really what you are going to be working with day-to-day. I believe that is what we're all trying to focus on. That is certainly what VMware is trying to focus on.

ALLEN STEWART: Think locally but really have your eye on what you're going to do with virtualization moving forward. Someone in the SMB space is typically looking at virtualization to get flexibility, but think about the actual applications, the use cases, and the user profiles to determine why you want to use virtualization in your environment.

Manageability of the environment is really going to be a critical aspect, not just the fact that you're creating a virtual machine. Integrating the stack into your environment is going to be very important from a small business perspective. You need to determine whether you will need to retrain your staff to integrate virtualization into your environment, and then weigh that against the benefits.

Certainly think about what you're running in your environment. If you're running Windows, think about using Hyper-V and some sort of high-level management construct that doesn't require you to do a large integration effort.

SIMON CROSBY: Virtualization is a feature set, not an objective. It's a technology that we should look at in the same way as compilers or TCP/IP stacks. It's a passing fad. The real benefits will come out of the overall ability to compose and manage an application throughout its lifecycle.

It is the application that IT is charged with delivering and not virtual machines. The sooner we move the debate from virtual machines back to delivering services to end users, the faster people will focus on the tools that will drive them through that application-life cycle process.

TOM BISHOP: I agree with that. IT transformation today is really all about two things: delivering the services that business cares about and doing it as cheaply and efficiently as possible. Virtualization has a role to play in both of those but it's just an enabler; it's not part of the higher level set of objectives. The challenge is how to fold in the capabilities that virtualization provides into a higher level set of mechanisms to enable you to achieve those two objectives. The harder challenge is changing the focus of what IT does and the people who do the work. A large number of IT people still view recovering the database as their job, not delivering business services.

GUSTAV: My definition of good engineering is *ease of removal*, not *ease of implementation*. One of the common characteristics of the available VM platforms is that transitions between them are relatively easy. Physical-to-virtual migrations don't actually depend on *you* being the physical part for them to work. If you were to look today at a physical-to-virtual migration of something that already happens to be in Veridian or VMware or Xen, it's going to work.

Since most of these platforms have quite sophisticated physical-to-virtual movements, worry less about whether you are tying yourself to something that you will be stuck with for many years, and worry more about the types of benefits you will gain from its use.

> **STEVE HERROD**
>
> **At the end of the day, virtualization is a tool. The goals are to make life better, and particularly for SMBs, to make computing simpler. To make it easy for SMBs is to enable them to operate highly available and securely, and to solve their business problems with their applications.**

TOM BISHOP: All of the issues we have been discussing are proxies for the fact that we build applications incorrectly. We build applications without regard to how much they cost to own, how much they cost to manage, and their impacts on their operating environments. As you design your infrastructure architectures, a conversation around application life cycle will be far more productive than a discussion around virtualization.

MACHE CREEGER: So what you're all telling me is something I learned in the AI (Artificial Intelligence) business in the early 1980s. AI was considered to be a market, even though I spent a great deal of time telling folks it was just a technology like compilers and file systems. Virtualization is replaying that old script today with the help of a strong media amplifier. Ultimately, just like AI, virtualization will get subsumed into the toolbox of best IT practices.

Folks need to avoid that hype and have confidence that regardless of vendor choice, all the VM platforms will get you where you need to go. They should focus on the services they need to deliver and work backward to the tools and technologies that best match their needs. They should believe that sensible people in the technical management of all these companies are working toward standards that will allow as much interoperation as is practical and that it will progress over time. As people better understand where virtualization fits as a component in an IT architecture, all the products will evolve towards common functionality. The real analysis should be on what management paradigms you choose and, if you are inclined towards a cloud-based platform, evaluating whether virtualization can be an asset in achieving the benefits of that paradigm.

Mache Creeger (mache@creeger.com) is the principal of Emergent Technology Associates, marketing and business development consultants.

DOI:10.1145/1409360.1409376

Tools for surviving a data deluge to ensure your data will be there when you need it.

BY FRANCINE BERMAN

Got Data? A Guide to Data Preservation in the Information Age

Imagine the modern world without digital data—anything that can be stored in digital form and accessed electronically, including numbers, text, images, video, audio, software, and sensor signals. We listen to digital music on our iPods, watch streaming video on YouTube, record events with digital cameras, and text our colleagues, family, and friends on BlackBerrys and cell phones. Many of our medical records, financial data, and other personal and professional information are in digital form. Moreover, the Internet and its digital products have become our library, shopping mall, classroom, and boardroom. It is difficult to imagine the information age without unlimited access to and availability of the digital data that is its foundation.

Digital data is also fragile. For most of us, an underlying assumption is that our data will be accessible whenever we want it. We also regularly confront the fallacy of this assumption; most of us (or our friends) have had hard drives crash with the loss of valuable information or seen storage media become obsolete, rendering information unavailable (think floppy disks). Loss, damage, and unavailability of important digital business, historical, and official documents regularly make the news, further highlighting our dependence on electronic information.

As a supporting foundation for our efforts in the information age, digital data in the cyberworld is analogous to infrastructure in the physical world, including roads, bridges, water, and electricity. And like physical infrastructure, we want our data infrastructure to be stable, predictable, cost-effective, and sustainable. Creating systems with these and other critical characteristics in the cyberworld of information technology involves tackling a spectrum of technical, policy, economic, research, education, and social issues. The management, organization, access, and preservation of digital data is arguably a "grand challenge" of the information age.

As a society, we have only begun to address this challenge at a scale concomitant with the deluge of data available to us and its importance in the modern world. This article explores the key trends and issues associated with preserving the digital data that is the natural resource of the information age and what's needed to keep it manageable, accessible, available, and secure. (For common terms associated with digital data management and preservation, see the sidebar "Digital Data Terms and Definitions.")

Data Cyberinfrastructure

The supporting framework for digital

PHOTOGRAPH BY THOMAS HERBRICH

data is an example of cyberinfrastructure—the coordinated aggregate of information technologies and systems (including experts and organizations) enabling work, recreation, research, education, and life in the information age. The relationship between cyberinfrastructure in the cyberworld and infrastructure in the physical world was described in the U.S. National Science Foundation's 2003 *Final Report of the Blue Ribbon Advisory Panel on Cyberinfrastructure*, commonly known as the "Atkins Report"[2] after its Chair, Dan Atkins: "The term *infrastructure* has been used since the 1920s to refer collectively to the roads, power grids, telephone systems, bridges, rail lines, and similar public works that are required for an industrial economy to function. Although good infrastructure is often taken for granted and noticed only when it stops functioning, it is among the most complex and expensive things that society creates. The newer term *cyberinfrastructure* refers to infrastructure based upon distributed computer, information, and communication technology. If infrastructure is required for an *industrial* economy, then we could say that cyberinfrastructure is required for a *knowledge* economy."

The implication of the report is that like infrastructure in the physical world, data cyberinfrastructure, or data CI, should exhibit critical characteristics that render it useful, usable, cost-effective, and unremarkable. The innovation, development, prototyping, and deployment of CI with such characteristics constitute a massive endeavor for all sectors, including the academic sector.[1,3]

What are the components of data CI? In the research and education community, users want a coordinated environment that manages digital data from creation to preservation, accommodates data ingested from instruments, sensors, computers, laboratories, people, and other sources, and includes data management tools and resources, data storage, and data use facilities (such as computers for analysis, simulation, modeling, and visualization). Users want to store and use their data for periods spanning the short-term (days) to the long-term (decades and beyond), and they want it to be available to their collaborators and communities through portals and other environments. Figure 1 outlines the portfolio of coordinated components that constitute the data CI environment at the San Diego Supercomputer Center (www.sdsc.edu/). Such environments must be designed to meet the needs of the target user community while being continually maintained and evolved to support digital data over the long term.

Trends

A 2008 International Data Corporation (IDC) white paper sponsored by EMC Corporation[5] described the world we live in as awash in digital data—an estimated 281 exabytes (2.25×10^{21} bits) in 2007. This is equivalent to 281 trillion digitized novels but less than 1% of Avogadro's number, or the number of atoms in 12 grams of carbon (6.022×10^{23}). By IDC estimates, the amount of digital data in our cyberworld will surpass Avogadro's number by 2023.[5] Even if these estimates are off significantly, storing, accessing, managing, preserving, and dealing with digital data is clearly a fundamental need and an immense challenge.

The development of data CI is greatly affected by both current and projected use-case scenarios, and our need to search, analyze, model, mine, and visualize digital data informs how we organize, present, store, and preserve it. More broadly, data CI is influenced by trends in technology, economics, policy, and law. Four significant trends reflect the larger environment in which data CI is evolving:

Trend 1. More digital data is being created than there is storage to host it. Estimates from the IDC white paper

indicate that 2007 marked the "crossover" year in which more digital data was created than there is data storage to host it. At that point, the amount of digital data (information created, captured, or replicated in digital form) exceeded the amount of storage (all empty or usable space on hard drives, tapes, CDs, DVDs, and volatile and nonvolatile memory). At the crossover point, this amount was estimated to be around 264 exabytes (264 × 1018 bytes).[5] This is almost a million times the amount of digital data hosted in 2008 by the U.S. Library of Congress (www.loc.gov/library/libarch-digital.html) and more than 20,000 times the aggregate of permanent electronic records projected to be stored in 2010 by the U.S. National Archives and Records Administration (www.archives.gov/era/). The IDC report further projected that by 2011 the amount of digital information created will be nearly 1.8 zettabytes (1.8×10^{21}), or more than twice the amount of available storage, estimated at 800+ exabytes.

The methodology under which these estimates were derived (what is counted and how it is calculated[6]) is fascinating and has generated considerable community discussion. However, even under alternative variations of the IDC methodology, the trend is unmistakable: We do not produce storage capacity at the same rate we produce digital information. Even if we wanted to, we cannot keep all of our digital data.

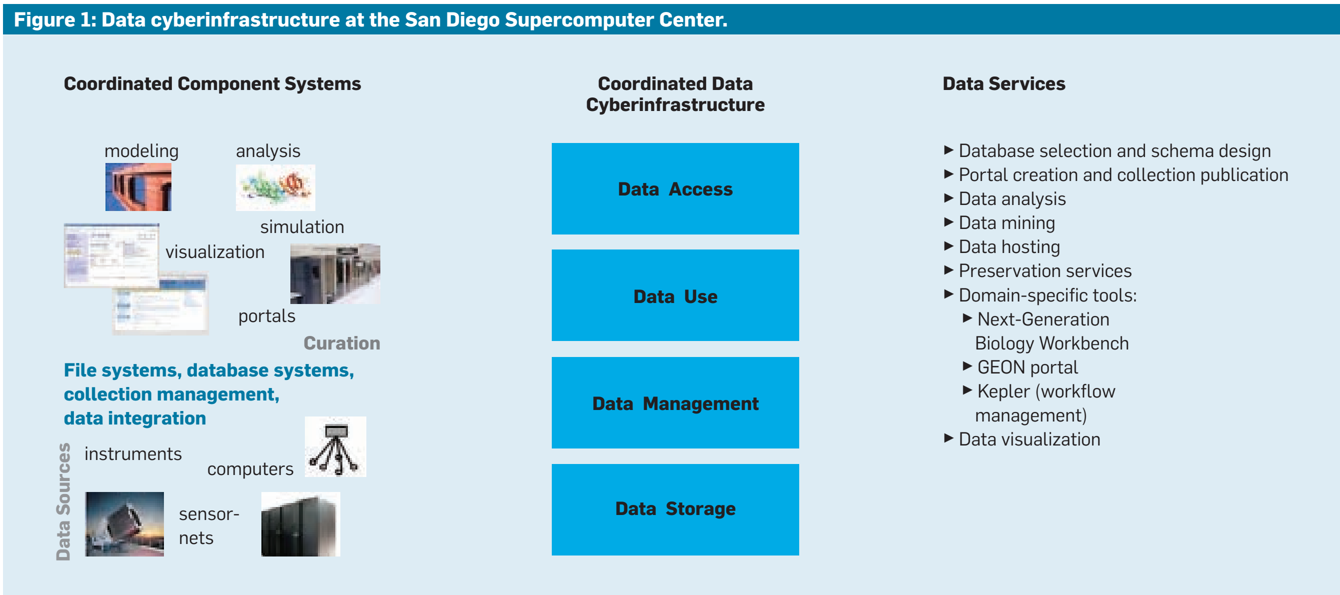
The thoughtful and methodical selection of which data sets to keep (called "appraisal" in the archival world) will be critical to communities used to keeping it all. In the research and education community, methods for community appraisal (coupled with the need for budgets to ensure adequate data stewardship and preservation for selected data sets) will likewise be more important over the next decade.

The need for community appraisal will push academic disciplines beyond individual stewardship, where project leaders decide which data is valuable, which should be preserved, and how long it should be preserved (except where regulation, policy, and/or publication protocols mandate specific stewardship and preservation timeframes). Some communities are beginning to develop explicit appraisal criteria and community stewardship models for valuable reference data collections (such as the Protein Data Bank, www.rcsb.org/pdb/home/home.do, in the life sciences and the Panel Study of Income Dynamics, psidonline.isr.umich.edu/Guide/, in the social sciences). Over the next decade, as more data is generated and the costs of data CI are incorporated into the "IT bill" at our institutions and enterprises, we can expect to devote more time and attention to the criteria and process through which we appraise data for stewardship and preservation.

Trend 2. More and more policies and regulations require the access, stewardship, and/or preservation of digital data. Even before the information age, the Copyright Clause (Article 1, Section 8) of the U.S. Constitution and subsequent regulation set the stage for policy with respect to the rights and dissemination of information in the U.S. Today, many forms of digital rights management and a broad range of public policies govern the access, stewardship, and preservation of digital data around the world. In the U.S., the Sarbanes-Oxley Act of 2002 promotes appropriate responsible management and preservation of digital financial and other records for publicly owned companies, and the Health Insurance Portability and Accountability Act of 1996 ensures the privacy of digital medical records. On the research front, investigators at the U.S. National Institutes of Health are required to submit digital copies of their publications to PubMed Central (publicaccess.nih.gov/), and the U.S. National Science Foundation's data-sharing policy "expects its awardees to share results of NSF-assisted research and education projects with others both within and outside the scientific and engineering research and education community."[10]

Increased emphasis on the access, preservation, and use of digital materials is not limited to the U.S. For example in the U.K., the Joint Information Systems Committee (www.jisc.ac.uk/) and the British Library (www.bl.uk/npo)

Figure 1: Data cyberinfrastructure at the San Diego Supercomputer Center.



have been leaders in data curation, access, and preservation issues. DigitalPreservationEurope (www.digitalpreservationeurope.eu) in the E.U., the National Library of Australia (www.nla.gov.au/policy/digpres.html), Koninklijke Biliotheek (the National Library of The Netherlands, www.kb.nl/index-en.html), and others around the world are contributing to an increasing body of knowledge and infrastructure to support data preservation and access for efforts enabled by technology.

The digital data generated by research, industry, and governments over the next decade will be subject to increased regulation and evolving community formats, standards, and policies. This means the CI developed to host and preserve it will need to incorporate mechanisms to enforce community policies and procedures like auditing, authentication, monitoring, and association of affiliated metadata. (As an unconventional example, think tagging of Facebook and Flickr photos.). Emerging data CI and management environments and systems, including IRODS (www.irods.org/), LOCKSS (www.lockss.org/lockss/Home), the Fedora Commons (www.fedora-commons.org/), and D-Space (www.dspace.org/), are beginning to develop and incorporate mechanisms that implement relevant policies and procedures. Over the next decade, the ability to automatically address the requirements of policy and regulation will be needed to ensure that our data CI empowers rather than limits us.

Trend 3. Storage costs for digital data are decreasing (but that's not the whole story). One of the most important trends affecting digital data is the decrease in price over time for a terabyte (10^{12}B) of data storage. According to IDC[11], a terabyte of "enterprise" storage was priced at roughly $440,000 in 1997. A decade later, the price for a terabyte of enterprise storage averaged around $5,400. In 2008, terabyte drives cost approximately $200 (OEM cost). In addition, holographic memory and other new technologies promise even better performance per price unit.

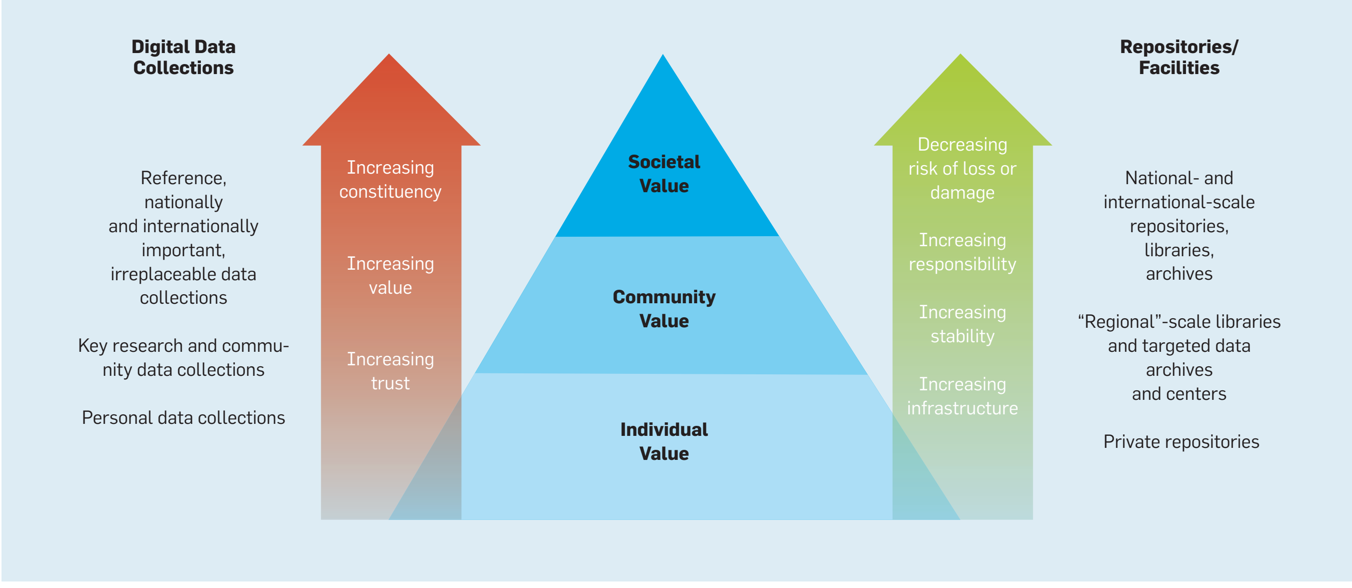
With storage so affordable, one would expect the "data bill" of institutions and enterprises to be equivalently affordable. However, as storage costs decrease, critical components of the data bill (such as power, curation/annotation, and professional expertise) are not decreasing. Today's companies and institutions are investing in enterprise data centers in locations selected to minimize the power bill. Google, Microsoft, and other technology companies spend billions on such data centers—the heart of their businesses—and the cost savings rendered through strategic placement can be immense. Storage costs may be going down, but the number of data centers and the cost of powering them are taking a bigger and bigger bite out of current and projected data budgets. (See Moore et al.[8] for a 2007 assessment of the San Diego Supercomputer Center's total cost of providing storage infrastructure.)

In addition, most data centers employ a knowledgeable, professional work force to ensure appropriate curation and annotation of digital data for the smooth running of the data center infrastructure and to plan ahead for future institutional and enterprise data needs. A capable data work force is important for all sectors and will likely increase as a percentage of the overall IT work force, along with the increasing need for a well-managed, sustainable digital data CI.

Finally, data centers must also factor in the cost of compliance with current and future regulations (possibly requiring additional physical and/or cyberinfrastructure for power backup and monitoring) and the need for replication of valuable data sets. (Data with long-term or substantive value is commonly stored with at least three copies, some hosted off-site.) We should expect the overall costs of data centers to continue to be substantial for the foreseeable future.

Trend 4. Increasing commercialization of digital data storage and services. The 2006 introduction of Amazon Simple Storage Solutions (www.amazon.com/gp/browse.html?node=16427261) was a high-profile example of the trend toward commercialization of data storage and data services. Today, there is considerable activity in the private sector around data storage and services for the consumer; for example, we share

Figure 2: The Data Pyramid.



and store digital photos through Flickr, employ Apple's Time Capsule for regular personal computer backup, and use LexisNexis for online legal services.

The commercialization of data storage and services contributes an important component of the data CI environment needed to harness the potential of our information-rich world. However, private-sector storage and services are not the solution to all digital data needs. For some digital data considered to be "in the public interest" (such as census data, official records, critical scientific data collections, and a variety of irreplaceable data), a greater level of trust, monitoring, replication, and accountability is required to minimize the likelihood of loss or damage and ensure the data will be there for a very long time. For such community data sets, stewardship by a trusted entity (such as libraries, archives, museums, universities, and institutional repositories), whose mission is the public good rather than profit, is generally required.

There is no one-size-fits-all solution for data stewardship and preservation. The "free rider" solution of "Let someone else do it"—whether that someone else is the government, a library, a museum, an archive, Google, Microsoft, the data creator, or the data user—is unrealistic and pushes responsibility to a single company, institution, or sector when what is needed are cross-sector economic partnerships. Sustainable economic models for digital data in the public interest are the focus of an international Blue Ribbon Task Force for Sustainable Digital Preservation and Access (brtf.sdsc.edu), whose goal is to examine digital preservation as an economic activity and explore cost frameworks for various institutional scenarios. The Task Force's final report, due at the end of 2009, will focus on economic models, components, and actionable recommendations for sustainable digital preservation and access, though it is already clear is that a diverse set of economic approaches are necessary.

In aggregate, these four trends point to the need to take a comprehensive and coordinated approach to data CI and treat the problem of sustainability holistically, creating strategies that make sense from a technical, policy, regulatory, economic, security, and community perspective.

We do not produce storage capacity at the same rate we produce digital information. Even if we wanted to, we cannot keep all of our digital data.

Value and Sustainability

In developing effective models for data CI, perhaps the greatest challenge is economic sustainability. A key question is: Who is responsible for supporting the preservation of valued digital data? Critical to answering is the recognition that "value" means different things to different people. There is general agreement that official digital government records (such as presidential email and videos of congressional hearings in the U.S.) are preservation-worthy and of great political and historical value to society, but the video of your niece's voice recital is likely to be of value to a much smaller family group (unless, of course, your niece is, say, Tina Turner).

Sustainability solutions for digital data are inextricably related to who values it and who is willing to support its preservation. Governments worldwide are willing to support the preservation of digital content of national value, a substantial undertaking that involves hosting multiple copies of the same data, migration of the data from one generation of storage media to the next to ensure it lives in perpetuity, and protection of its integrity and authenticity. Your niece's voice recital may live on the hard drives of one or more family members, but there is rarely an explicit plan for how such a treasured family artifact will be preserved for the next decade and beyond.

How might we distinguish among all the data use, stewardship, and preservation scenarios to create and identify the data CI solutions needed to support them? One way is to borrow from the world of computation and adapt the Branscomb Pyramid model to today's data-use and data-stewardship scenarios. In 1993, the NSF asked Lewis Branscomb to lead a distinguished committee to consider the future of high-performance computing in the U.S. The final report included a useful framework, now known as the Branscomb Pyramid, where the base of the Pyramid associated the least-powerful computational platforms with users needing computation for "everyday" applications, the middle associated more powerful computational platforms with users whose applications require greater performance, and the tip associated the most powerful computational platforms with the users requiring the greatest performance for "hero" applications. The same approach can be used to create a Data Pyramid (see Figure 2) to frame today's digital information and stewardship options.

The Data Pyramid outlines the spectrum of data-collection and data-stewardship alternatives. The bottom includes data of individual ("local") value whose stewards focus primarily on individual needs (such as personal tax records and digital family photographs and videos). We back this up on our hard drives, with an additional copy off-site if we are methodical, but little of this data will ever be considered of great societal value.

At the top is data of widespread and/or societal value whose stewards are primarily public-interest institutions (such as government agencies, libraries, museums, archives, and universities). Included are official records, data infeasible or too expensive to replace (such as the Shoah Collection of holocaust survivor testimony, college.usc.edu/vhi/, and digital photographs from the most recent NASA space voyage). Much of it must be preserved over the long term by trusted institutions. It is typically replicated many times, the focus of explicit plans for preservation, and hosted by only the most reliable cyberinfrastructure.

In the middle of the Pyramid is data of value to a specific community whose stewards range from individuals to community groups to companies to public-interest institutions. It includes digital records from your local hospital, scientific research data preserved in community repositories, and digital copies of motion pictures preserved for decades, commercially valuable in the future in the form of, say, the "director's cut." In every sector, groups are beginning to grapple with the responsibility of creating plans for data stewardship that are cost-effective, support reliable digital preservation, and are not subject to the whims of markets and/or community social dynamics.

The Data Pyramid makes it easy to see that multiple solutions for sustainable digital preservation must be

Digital Data Terms and Conditions

The following definitions are derived from a number of sources, including the American Library Association (www.lita.org/ala/), National Information Assurance Glossary (www.cnss.gov/), and Joint Information Systems Committee Digital Information Briefing Paper (www.jisc.ac.uk):

APPRAISAL
Evaluation and selection of digital material for long-term curation and preservation, documented policies, guidance, and legal requirements may require that it be done securely;

AUTHENTICATION
Security measure designed to establish the validity of a transmission, message, or originator or a means of verifying an individual's authority to receive specific categories of information;

CURATION
Digital curation, broadly interpreted, is about maintaining and adding value to a trusted body of digital information for current and future use. It builds on the underlying concepts of digital preservation while emphasizing opportunities for added value and knowledge through annotation and continuing resource management;

DIGITAL RIGHTS MANAGEMENT
The use of technologies to control how digital content is used and reused;

INGEST
Controlled or secure transfer of material to an archive, repository, data center, or other custodial environment in adherence to documented guidance, policies, or legal requirements;

INTEGRITY
The condition when data is unchanged from its source and has not been accidently or maliciously modified, altered, or destroyed;

METADATA
Documentation relating to data content, structure, provenance (history), and context (such as experimental parameters and environmental conditions). Standards for metadata provide a basis for widespread community data sharing; and

PRESERVATION ACTION
Actions undertaken to ensure the long-term viability and availability of the authoritative nature of digital material. Preservation actions should ensure the material remains authentic, reliable, and usable while its integrity is maintained; such actions include validation, assigning preservation metadata, assigning representation information, and ensuring acceptable data structures and file formats.

devised. At the bottom, commercial services fill the need for primary, additional, or backup sites for collections of individual or private value. At the top, stewardship is primarily in the hands of libraries, museums, archives, government funding agencies, and other trusted institutions. In the middle, institutions, communities, enterprises, and others are the primary stewards of data, wrestling with institutional and community solutions for stable and sustainable digital stewardship and preservation. The next decade will likely see more creative partnerships among all the players in the Pyramid, as well as more attention to who actually pays the data bill and how its costs are managed.

Creating an economically viable Data Pyramid must also be complemented with continued research into and development of solutions that address the technical challenges of data management and preservation, resulting in the ability to utilize and create new knowledge from the data being stored. For example, the process of searching and mining data depends on how it is organized, what additional information (metadata) is associated with it, and what information might be included about the relationship (ontological structure) of data items to one another in a collection. All these functions are active and important areas for research. Privacy and policy controls for data collections and security of the supporting infrastructure are also critical research areas. Addressing the technical, economic, and social aspects of digital preservation will be critical to ensuring that the information age has the foundation required to achieve its potential.

Top 10 Guidelines for Data Stewardship

Whether your data portfolio is of personal, community, or societal value (or some combination), its viability and usefulness to you will result from how you plan for stewardship and preservation over its lifetime. The following guidelines help promote effective stewardship and preservation of digital data:

1. *Make a plan.* Create an explicit strategy for stewardship and preservation for your data, from its inception to the end of its lifetime; explicitly consider what that lifetime may be;
2. *Be aware of data costs and include them in your overall IT budget.* Ensure that all costs are factored in, including hardware, software, expert support, and time. Determine whether it is more cost-effective to regenerate some of your information rather than preserve it over a long period;
3. *Associate metadata with your data.* Metadata is needed to be able to find and use your data immediately and for years to come. Identify relevant standards for data/metadata content and format, following them to ensure the data can be used by others;
4. *Make multiple copies of valuable data.* Store some of them off-site and in different systems;
5. *Plan for the transition of digital data to new storage media ahead of time.* Include budgetary planning for new storage and software technologies, file format migrations, and time. Migration must be an ongoing process. Migrate data to new technologies before your storage media goes obsolete;
6. *Plan for transitions in data stewardship.* If the data will eventually be turned over to a formal repository, institution, or other custodial environment, ensure it meets the requirements of the new environment and that the new steward indeed agrees to take it on;
7. *Determine the level of "trust" required when choosing how to archive data.* Are the resources of the U.S. National Archives and Records Administration necessary or will Google do?;
8. *Tailor plans for preservation and access to the expected use.* Gene-sequence data used daily by hundreds of thousands of researchers worldwide may need a different preservation and access infrastructure from, say, digital photos viewed occasionally by family members;
9. *Pay attention to security.* Be aware of what you must do to maintain the integrity of your data; and
10. *Know the regulations.* Know whether copyright, the Health Insurance Portability and Accountability Act of 1996, the Sarbanes-Oxley Act of 2002, the U.S. National Institutes of Health publishing expectations, or other policies and/or regulations are relevant to your data, ensuring your approach to stewardship and publication is compliant.

While adherence is not a magic bullet guaranteeing the long-term safety and accessibility of fragile digital data, these guidelines help focus appropriate attention, effort, and support on the maintenance and preservation of our valued digital information. Such attention is critical to our ability to harness the immense potential of the information age to illuminate and empower us in our changing world.

Acknowledgment

I am grateful to John Gantz, Chris Greer, Nancy McGovern, David Minor, David Reinsel, Brian Schottlaender, Jan Zverina, and the reviewers for their useful comments and generous help with this article.

References

1. Alvarez, R. *Developing and Extending a Cyberinfrastructure Model.* Research Bulletin 5. Educause Center for Applied Research, Boulder, CO, 2008.
2. Atkins, D. *Revolutionizing Science and Engineering Through Cyberinfrastructure: Report of the National Science Foundation Blue Ribbon Advisory Panel on Cyberinfrastructure.* NSF report. NSF, Arlington, VA, 2003; www.nsf.gov/od/oci/reports/toc.jsp.
3. Berman, F. Making cyberinfrastructure real. *Educause Review 43*, 4 (July/Aug. 2008), 18–32.
4. Branscomb, L. et al. *From Desktop to TeraFlop: Exploiting the U.S. Lead in High-Performance Computing. Final Report of the National Science Foundation Blue Ribbon Panel on High-Performance Computing.* National Science Foundation, Arlington, VA, 1993; www.nsf.gov/pubs/stis1993/nsb93205/nsb93205.txt.
5. Gantz, J. *The Diverse and Exploding Digital Universe.* White paper. International Data Corporation, Framingham, MA, Mar. 2008; www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf.
6. Gantz, J. *The Expanding Digital Universe.* White paper. International Data Corporation, Framingham, MA, Mar. 2007; www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf (methodology discussion begins on 17).
7. Higgins, S. Draft DCC curation model. *International Journal of Digital Curation 2*, 2 (2007), 82–87.
8. Moore, R., D'Aoust, J, McDonald, R., and Minor, D. Disk and tape storage cost models. In *Proceedings of the Society for Imaging Science and Technology's Archiving Conference* (Arlington, VA, 2007), 29–32; users.sdsc.edu/~mcdonald/content/papers/dt_cost.pdf.
9. National Science Board, *Long-Lived Digital Data Collections: Enabling Research and Education in the 21st Century*. Arlington, VA, Sept. 2005; www.nsf.gov/pubs/2005/nsb0540/nsb0540.pdf.
10. National Science Foundation. *NSF Data Sharing Policy.* Arlington, VA, 2001; www.nsf.gov/pubs/2001/gc101/gc101rev1.pdf.
11. Reinsel, D. Personal communication. Group Vice President, Storage and Semiconductors, International Data Corporation, July, 2008.

Francine Berman (berman@sdsc.edu) is Director of the San Diego Supercomputer Center, Professor of Computer Science and Engineering, and HPC High Performance Computing Endowed Chair in the Jacobs School of Engineering at the University of California, San Diego, and is also co-chair of the Blue Ribbon Task Force on Sustainable Digital Preservation and Access.

DOI:10.1145/1409360.1409377

How ontologies provide the semantics, as explained here with the help of Harry Potter and his owl Hedwig.

BY IAN HORROCKS

Ontologies and the Semantic Web

While phenomenally successful in terms of amount of accessible content and number of users, today's Web is a relatively simple artifact. Web content consists mainly of distributed hypertext and hypermedia, accessible via keyword-based search and link navigation. Simplicity is one of the Web's great strengths and an important factor in its popularity and growth; even naive users quickly learn to use it and even create their own content.

However, the explosion in both the range and quantity of Web content also highlights serious shortcomings in the hypertext paradigm. The required content becomes increasingly difficult to locate via search and browse; for example, finding information about people with common names (or famous namesakes) can be frustrating. Answering more complex queries, along with more general information retrieval, integration, sharing, and processing, can be difficult or even impossible; for example, retrieving a list of the names of E.U. heads of state is apparently beyond the capabilities of all existing Web query engines, in spite of the fact that the relevant information is readily available on the Web. Such a task typically requires the integration of information from multiple sources; for example, a list of E.U. member states can be found at europa.eu, and a list of heads of state by country can be found at rulers.org.

Specific integration problems are often solved through some kind of software "glue" that combines information and services from multiple sources. For example, in a so-called mashup, location information from one source might be combined with map information from another source to show the location of and provide directions to points of interest (such as hotels and restaurants). Another approach, seen increasingly in so-called Web 2.0 applications, is to harness the power of user communities in order to share and annotate information; examples include image- and video-sharing sites (such as Flickr and YouTube) and auction sites (such as eBay). In them, annotations usually take the form of simple tags (such as "beach," "birthday," "family," and "friends"). However, the meaning of tags is typically not well defined and may be impenetrable even to human users; examples (from Flickr) include "sasquatchmusicfestival," "celebrity-lookalikes," and "twab08."

Despite their usefulness, these approaches do not solve the general problem of how to locate and integrate information without human intervention. This is the aim of the semantic Web[3] according to the World Wide Web Consortium (W3C) Semantic Web FAQ; the goal is to "allow data to be shared effectively by wider communities, and to be processed automatically by tools as well as manually." The prototypical example of a semantic Web application is an automated travel agent that, given constraints and preferences, gives the user suitable travel or vacation suggestions. A key feature of such a "software agent" is that it would not simply exploit a predetermined set of information sources but search the Web for relevant information in much the same way a human user might when planning a vacation.

A major difficulty in realizing this goal is that most Web content is primarily intended for presentation to and consumption by human users; HTML markup is primarily concerned with layout, size, color, and other presentation issues. Moreover, Web pages increasingly use images, often with active links, to present information; even when content is annotated, the annotations typically take the form of natural-language strings and tags. Human users are (usually) able to interpret the significance of such features and thus understand the information being presented, a task that may not be so easy for software agents.

This vision of a semantic Web is extremely ambitious and would require solving many long-standing research problems in knowledge representation and reasoning, databases, computational linguistics, computer vision, and agent systems. One such problem is the trade-off between conflicting requirements for expressive power in the language used for semantic annotations and the scalability of the systems used to process them[7]; another is that integrating different ontologies may prove to be at least as difficult as integrating the resources they describe.[18] Emerging problems include how to create suitable annotations and ontologies and how to deal with the variable quality of Web content.

Notwithstanding such problems, considerable progress is being made in the infrastructure needed to support the semantic Web, particularly in the development of languages and tools for content annotation and the design and deployment of ontologies. My aim here is to show here that even if a full realization of the semantic Web is still a long way off, semantic Web technologies already have an important influence on the development of information technology.

Semantic Annotation

The difficulty of sharing and processing Web content, or resources, derives in part from the fact that much of it (such as text, images, and video) is unstructured; for example, a Web page might include the following unstructured text:

Harry Potter has a pet named Hedwig.

As it stands, it would be difficult or impossible for a software agent (such as a search engine) to recognize the fact that this resource describes a young wizard and his pet owl. We might try to make it easier for agents to process Web content by adding annotation tags (such as `Wizard` and `Snowy Owl`). However, such tags are of only limited value. First, the problem of understanding the terms used in the text is simply transformed into the problem of understanding the terms in the tags; for example, a query for information about raptors may not retrieve the text, even though owls are raptors. Moreover, the relationship between Harry Potter and Hedwig is not captured in these annotations, so a query asking for wizards having pet owls might not retrieve Harry Potter.

We might also want to integrate information from multiple sources; for example, rather than coin our own term for Snowy Owl, we might want to point to the relevant term in a resource providing definitive information about owls. RDF is a language that provides a flexible mechanism for describing Web resources and the relationships among them.[14] A key feature of RDF is its use of internationalized resource identifiers (IRIs)—a generalization of uniform resource locators (URLs)—to refer to resources. Using IRIs facilitates information integration by allowing RDF to directly reference non-local resources. IRIs are typically long strings (such as `hogwarts.net/HarryPotter`), though abbreviation mechanisms are available; here, I usually omit the prefix and just write `HarryPotter`.

ILLUSTRATION BY MIA ANGELICA BALAQUIOT

RDF is a simple language; its underlying data structure is a labeled directed graph, and its only syntactic construct is the triple, which consists of three components, referred to as subject, predicate, and object. A triple represents a single edge (labeled with the predicate) connecting two nodes (labeled with the subject and object); it describes a binary relationship between the subject and object via the predicate. For example, we might describe the relationship between Harry and Hedwig using this triple:

Figure 1: Example RDF graph.

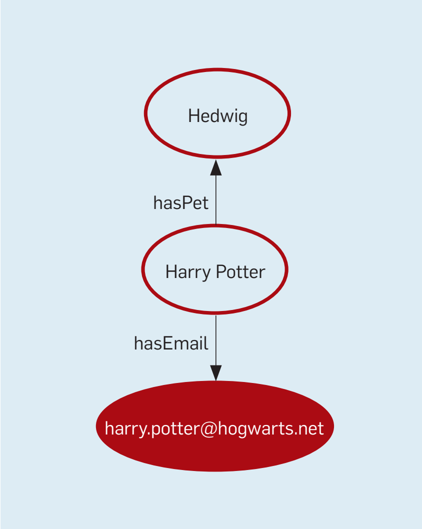
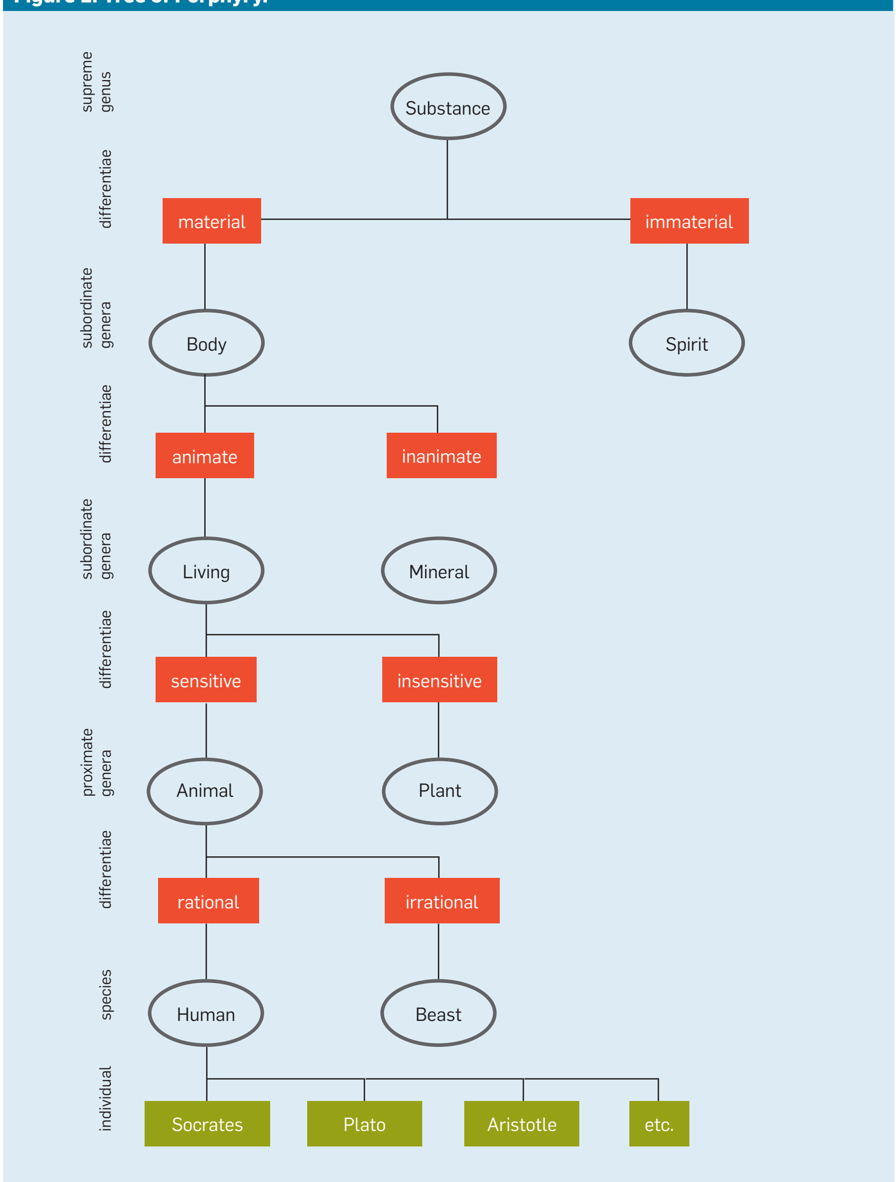


Figure 2: Tree of Porphyry.




```
HarryPotter hasPet Hedwig .
```

where `HarryPotter` is the subject, `hasPet` is the predicate, and `Hedwig` is the object. The subject of a triple is either an IRI or a blank node (an unlabeled node), while the object is an IRI, a blank node, or a literal value (such as a string or integer). For example, we could use the triple:

```
HarryPotter hasemail
"harry.potter@hogwarts.net".
```

to capture information about Harry's email address. The predicate of a triple is always an IRI called a "property." IRIs are treated as names that identify particular resources. Blank nodes also denote resources, but the exact resource being identified is not specified, behaving instead like existentially quantified variables in first-order logic.

A set of triples is called an RDF graph (see Figure 1). In order to facilitate the sharing and exchanging of graphs on the Web, the RDF specification includes an XML serialization. In RDF/XML the triples can be written as

```
  <rdf:Description
rdf:about="#HarryPotter">
  <hasPet
rdf:resource="#Hedwig"/>
  <hasEmail>harry.potter@
hogwarts.net
  </hasEmail>
  </rdf:Description>
```

where `#HarryPotter` and `#Hedwig` are fragment identifiers.

The RDF specification also extends the capabilities of the language by giving additional meaning to certain resources. One of the most important is `rdf:type`, a special property that captures the class-instance relationship; where rdf is an abbreviation (called a "namespace prefix") for the string www.w3.org/1999/02/22-rdf-syntax-ns#. For example, we could use the triple:

```
HarryPotter rdf:type Wizard .
```

to represent the fact that Harry is an instance of Wizard.

RDF provides a flexible mechanism for adding structured annotations but does little to address the problem of understanding the meaning, or semantics, of the terms in annotations. One possible solution would be to fix a set of terms to be used in annotations and agree on their meaning. This works well in constrained settings like annotating documents; the Dublin Core Metadata Initiative (dublincore.org/schemas/) defines just such a set of terms, including, for example, the properties `dc:title`, `dc:creator`, `dc:subject`, and `dc:publisher`. However, this approach is limited with respect to flexibility and extensibility; only a fixed number of terms is defined, and extending the set typically requires a lengthy process in order to agree on which terms to introduce, as well as on their intended semantics. It may also be impractical to impose a single set of terms on all information providers.

An alternative approach is to agree on a language that can be used to define the meaning of new terms (such as by combining and/or restricting existing ones). Such a language should preferably be relatively simple and precisely specified so as to be amenable to processing by software tools. This approach provides greatly increased flexibility, as new terms can be introduced as needed. This is the approach taken in the semantic Web, where ontologies are used to provide extensible vocabularies of terms, each with a well-defined meaning; for example, a suitable ontology might introduce the term SnowyOwl and include the information that a SnowyOwl is a kind of owl and that owl is a kind of raptor. Moreover, if this information is represented in a way that is accessible to our query engine, the engine would be able to recognize that Hedwig should be included in the answer to a query concerning raptors.

Ontology, in its original philosophical sense, is a branch of metaphysics focusing on the study of existence; its objective is to study the structure of the world by determining what entities and types of entities exist. The study of ontology can be traced back to the work of Plato and Aristotle, including their development of hierarchical categorizations of different kinds of entity and the features that distinguish them; for example, the "tree of Porphyry" identifies animals and plants as subcategories of living things distinguished from each other by animals having "sensitive" souls, with powers of sense, memory, and imagination (see Figure 2).

In computer science, an ontology is an engineering artifact, usually a model of (some aspect of) the world; it introduces vocabulary describing various aspects of the domain being modeled and provides an explicit specification of the intended meaning of that vocabulary. However, the specification often includes classification-based information, not unlike Porphyry's tree; for example, Wizard may be described as a subcategory of human, with distinguishing features (such as the ability to perform magic).

The RDF vocabulary description language (RDF schema) extends RDF to include the basic features needed to define ontologies. This extension is achieved by giving additional meaning to more "special" resources, including `rdfs:Class`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain`, and `rdfs: range`, where `rdfs` is an abbreviation for the string www.w3.org/2000/01/rdf-schema#. The `rdfs:Class` resource is the class of all RDF classes; a resource (such as Wizard) that is the object of an `rdf:type` triple is itself an instance of the `rdfs:Class` resource. The `rdfs:subClassOf` and `rdfs:subPropertyOf` properties can be used in an ontology to describe a hierarchy of classes and properties, respectively. For example, the triples:

```
SnowyOwl rdfs:subClassOf Owl .
Owl rdfs:subClassOf Raptor .
```

can be used to represent the fact that a SnowyOwl is a kind of Owl and that an Owl is a kind of Raptor. Similarly, the triple:

```
hasBrother rdfs:subPropertyOf hasSibling .
```

can be used to represent the fact that if x has a brother y, then x also has a sibling y. Additionally, a property's domain and range can be specified using `rdfs:domain` and `rdfs:range`. For example, the triples:

```
hasPet rdfs:domain Human.
hasPet rdfs:range Animal.
```

can be used to represent the fact that only Humans can have pets and that all pets are Animals.

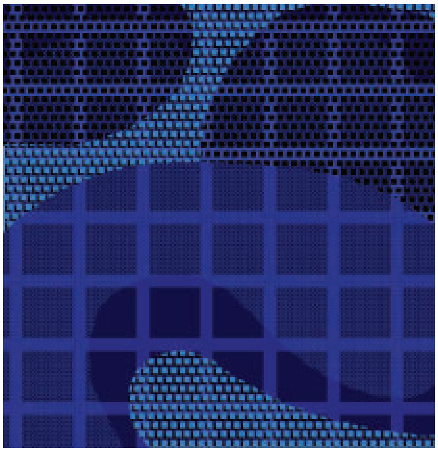
The Web Ontology Language OWL

Though obviously an ontology language, RDF is rather limited; it is not able to, for example, describe cardinality constraints (such as Hogwarts students have at most one pet), a feature in most conceptual modeling languages, or describe even a simple conjunction of classes (such as Student and Wizard). In the late 1990s, the need for a more expressive ontology language was widely recognized within the nascent semantic Web research community and resulted in several proposals for new Web ontology languages, including Simple HTML Ontological Extensions (SHOE), the Ontology Inference Layer (OIL), and DAML+OIL.

In 2001, recognizing that an ontology-language standard is a prerequisite for the development of the semantic Web, the W3C set up a standardization working group to develop a standard for a Web ontology language. The result, in 2004, was the OWL ontology language standard (www.w3.org/2004/OWL/), exploiting the earlier work on OIL and DAML+OIL while tightening the integration of these languages with RDF. Integrating OWL with RDF provided OWL with an RDF-based syntax, with the advantage of making OWL ontologies directly accessible to Web-based applications, though the syntax is rather verbose and difficult to read; for example, in RDF/XML, the description of the class of Student Wizards would be written as:

```
<owl:Class>
<owl:intersectionOf
rdf:parseType="Collection">
 <owl:Class
rdf:about="#Student"/>
 <owl:Class
rdf:about="#Wizard"/>
</owl:intersectionOf>
</owl:Class>
```

For this reason, here I use an informal "human-readable" syntax based on the one used in the Protégé 4 ontology development tool (protege.stanford.edu/) in which the description is written as:



```
Student and Wizard
```

A key feature of OWL is its basis in Description Logics (DLs), a family of logic-based knowledge-representation formalisms descended from Semantic Networks and KL-ONE but that have a formal semantics based on first-order logic.[1] These formalisms all adopt an object-oriented model like the one used by Plato and Aristotle in which the domain is described in terms of individuals, concepts (called "classes" in RDF), and roles (called "properties" in RDF). Individuals (such as Hedwig) are the basic elements of the domain; concepts (such as Owl) describe sets of individuals with similar characteristics; and roles (such as hasPet) describe relationships between pairs of individuals (such as "HarryPotter hasPet Hedwig"). To avoid confusion here I keep to the RDF terminology, referring to these basic language components as individuals, classes, and properties.

Along with atomic-class names like Wizard and Owl, DLs also allow for class descriptions to be composed from atomic classes and properties. A given DL is characterized by the set of constructors provided for building class descriptions. OWL is based on a very expressive DL called *SHOIN*(**D**), a sort of acronym derived from the features of the language.[11] The class constructors available in OWL include the Booleans and, or, and not, which in OWL are called, respectively, intersectionOf, unionOf, and complementOf, as well as restricted forms of existential (∃) and universal (∀) quantification, which in OWL are called, respectively, someValuesFrom and allValuesFrom restrictions. OWL also allows for properties to be declared transitive; if `hasAncestor` is a transitive property, then `Enoch hasAncestor Cain` and `Cain hasAncestor Eve` implies that `Enoch hasAncestor Eve`. The *S* in *SHOIN*(**D**) stands for this basic set of features.

In OWL, someValuesFrom restrictions are used to describe classes, the instances of which are related via a given property to instances of some other class. For example,

```
Wizard and hasPet some Owl
```

describes Wizards having pet Owls. Note that such a description is itself a class, the instances of which are exactly those individuals that satisfy the description; in this case, they are instances of Wizard and are related via the hasPet property to an individual that is an instance of Owl. If an individual is asserted (stated) to be a member of this class, we know it must have a pet Owl, though we may be unable to identify the Owl in question; that is, someValuesFrom restrictions specify the existence of a relationship. In contrast, allValuesFrom restrictions constrain the possible objects of a given property and are typically used as a kind of localized range restriction. For example, we might want to state that Hogwarts students are allowed to have only owls, cats, or toads as pets without placing a global range restriction on the hasPet property (because other kinds of pet may be possible). We can do this in OWL like this:

```
Class: HogwartsStudent
  SubClassOf: hasPet only
(Owl or Cat or Toad)
```

A key feature of OWL is its basis in Description Logics, a family of logic-based knowledge-representation formalisms that are descendants of Semantic Networks and KL-ONE but that have a formal semantics based on first-order logic.

OWL also allows for property hierarchies (the *H* in *SHOIN*(**D**)), extensionally defined classes using the *oneOf* constructor (*O*), inverse properties using the *inverseOf* property constructor (*I*), cardinality restrictions using the *minCardinality*, *maxCardinality*, and *cardinality* constructors (*N*) and XML Schema datatypes and values (**D**) (www.w3.org/TR/xmlschema-2/). For example, we might also state that the instances of HogwartsHouse are Gryffindor, Slytherin, Ravenclaw, and Hufflepuff, that Hogwarts students have an email address (a string), and at most one pet, that `isPetOf` is the inverse of `hasPet`, and that a Phoenix can be the pet only of a Wizard:

```
Class: HogwartsHouse
  EquivalentTo: {Gryffindor, Slytherin, Ravenclaw,
Hufflepuff}
Class: HogwartsStudent
  SubClassOf: hasEmail some
string
  SubClassOf: hasPet max 1
ObjectProperty: hasPet
  Inverses: isPetOf
Class: Phoenix
  SubClassOf: isPetOf only
Wizard
```

An OWL ontology consists of a set of axioms. As in RDF, the axioms *subClassOf* and *subPropertyOf* can be used to define a hierarchy of classes and properties. In OWL, an *equivalentClass* axiom can also be used as an abbreviation for a symmetrical pair of subClassOf axioms. An *equivalentClass* axiom can be thought of as an "if and only if" condition; given the axiom C equivalentClass D, an individual is an instance of C if and only if it is an instance of D. Combining the axioms *subClassOf* and *equivalentClass* with class descriptions allows for easy extension of the vocabulary by introducing new names as abbreviations for descriptions. For example, the axiom

```
Class: HogwartsStudent
  EquivalentTo: Student and
attendsSchool
    value Hogwarts
```

introduces the class name `HogwartsStudent`, asserting that its instances are exactly those Students who attend Hogwarts. Axioms can also be used to state that a set of classes is disjoint and describe additional characteristics of properties. Besides being *Transitive*, a property can be *Symmetric*, *Functional*, or *InverseFunctional*; for example, the axioms

```
DisjointClasses: Owl Cat Toad
Property: isPetOf
  Characteristics: Functional
```

state that Owl, Cat, and Toad are disjoint (that is, they have no instances in common) and that `isPetOf` is functional (that is, pets can have only one owner).

These axioms describe constraints on the structure of the domain and play a role similar to the conceptual schema in a database setting; in DLs, such a set of axioms is called a terminology box (TBox). OWL also allows for axioms that assert facts about concrete situations, like data in a database setting; in DLs, such a set of axioms is called an assertion box (ABox). These axioms might, for example, include the facts

```
Individual: HarryPotter
  Types: HogwartsStudent
Individual: Fawkes
  Types: Phoenix
  Facts: isPetOf Dumbledore
```

Basic facts, or those using only atomic classes, correspond directly to RDF triples; for example, the facts just discussed correspond to the following triples:

```
HarryPotter rdf:type, HogwartsStudent .
Fawkes rdf:type Phoenix .
Fawkes isPetOf Dumbledore .
```

The term "ontology" is often used to refer to a conceptual schema or TBox, but in OWL an ontology can consist of a mixture of both TBox and ABox axioms; in DLs, this combination is called a knowledge base.

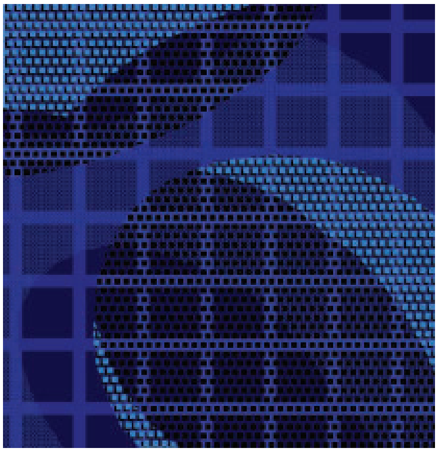
DLs are fully fledged logics and so have a formal semantics. They can, in fact, be understood as decidable subsets of first-order logic, with individuals being equivalent to constants, concepts to unary predicates, and roles to binary predicates. Besides giving a precise and unambiguous meaning to descriptions of the domain, the formal semantics also allows for the development of reasoning algorithms that can be used to correctly answer arbitrarily complex queries about the domain. An important aspect of DL research is the design of such algorithms and their implementation in (highly optimized) reasoning systems that can be used by applications to help them "understand" the knowledge captured in a DL-based ontology.

Ontology Reasoning

Though there are clear analogies between databases and OWL ontologies, there are also important differences. Unlike databases, OWL has a so-called open-world semantics in which missing information is treated as unknown rather than as false and OWL axioms behave like inference rules rather than as database constraints. For example, we have asserted that Fawkes is a Phoenix and a pet of Dumbledore and that only a Wizard can have a pet Phoenix. In OWL, this leads to the implication that Dumbledore is a Wizard; if we were to query the ontology for instances of Wizard, then Dumbledore would be part of the answer. In a database setting the schema could include a similar statement about the Phoenix class, but it would (in this case) be interpreted as a constraint on the data. Adding the fact that Fawkes isPetOf Dumbledore without Dumbledore being known to be a Wizard would lead to an invalid database state; such an update would be rejected by a database management system as a constraint violation.

Unlike databases, OWL makes no unique name assumption; for example, given that isPetOf is a functional property, then additionally asserting that Fawkes isPetOf AlbusDumbledore would imply that Dumbledore and AlbusDumbledore are two names for the same Wizard. In a database setting this would again be treated as a constraint violation. Note that in OWL it is possible to assert (or infer) that two different names do not refer to the same individual; if such an assertion were made about Dumbledore and AlbusDumbledore, then asserting that Fawkes isPetOf AlbusDumbledore would make the ontology inconsistent. Unlike database management systems, ontology tools typically don't reject updates that result in the ontology becoming wholly or partly inconsistent; they simply provide a suitable warning.

The treatment of schema and constraints in a database setting means they can be ignored when answering queries; in a valid database instance, all schema constraints must already be satisfied. This treatment makes query answering highly efficient; for example, in order to determine if Dumbledore is in the answer to a query for Wizards, it is sufficient to check if this fact is explicitly present in the database. In OWL, the schema plays a much more important role and is actively considered at query time. Considering both the schema and the data can be very powerful, making it possible to answer conceptual, as well as extensional, queries; for example, we can ask not only if Dumbledore is a Wizard but if anybody having a Phoenix for a pet is necessarily a Wizard. This power does, however, make query-answering much more difficult (at least in the worst case); for example, in order to determine if Dumbledore is in the answer to a query



for Wizards, it is necessary to check if Dumbledore would be an instance of Wizard in every possible state of the world that is consistent with the axioms in the ontology. Query answering in OWL is thus analogous to theorem proving, and a query answer is often referred to as an "entailment." OWL is therefore most suited to applications where the schema plays an important role, where it is not reasonable to assume that complete information about the domain is available, and where information has high value.

Ontologies may be very large and complex; for example, the Systematized Nomenclature of Medicine–Clinical Terms (SNOMED CT) ontology includes more than 400,000 class names. Building and maintaining such an ontology is costly and time-consuming, so providing tools and services to support the ontology-engineering process is critical to both the cost and the quality of the resulting ontology. Ontology reasoning therefore plays a central role in both the development of high-quality ontologies and the deployment of ontologies in applications.

In spite of the complexity of reasoning with OWL ontologies, highly optimized DL reasoning systems (such as FaCT++, owl.man.ac.uk/factplusplus/, Racer, www.racer-systems.com/, and Pellet, pellet.owldl.com/) have proved effective in practice; the availability of such systems was one of the key motivations for the W3C to base OWL on a DL. State-of-the-art ontology-development tools (such as SWOOP, code.google.com/p/swoop/, Protégé 4, and TopBraid Composer, www.topbraidcomposer.com) use DL reasoners to give feedback to developers about the logical implications of their designs. This feedback typically includes warnings about inconsistencies and synonyms.

An inconsistent (sometimes called "unsatisfiable") class is one for which its description is "overconstrained," with the result that it can never have instances. This inconsistency is typically an unintended consequence of the design (why introduce a name for a class that can never have instances?) and may be due to subtle interactions among axioms. It is therefore useful to be able to detect such classes and bring them to the attention of the ontology engineer. For example, during the recent development of an OWL ontology at NASA's Jet Propulsion Laboratory, the class "OceanCrustLayer" was found to be inconsistent. Engineers discovered (with the help of debugging tools) that this was the result of its being defined as both a region and a layer, one (a layer) a 2D object and the other (a region) a 3D object. The inconsistency thus highlighted a fundamental error in the ontology's design.

It is also possible that the descriptions in an ontology mean that two classes necessarily have exactly the same set of instances; that is, they are alternative names for the same class. Having multiple names for the same class may be desirable in some situations (such as to capture the fact that "myocardial infarction" and "heart attack" are the same thing). However, multiple names could also be the inadvertent result of interactions among descriptions or of basic errors by the ontology designer; it is therefore useful to be able to alert developers to the presence of such synonyms.

Reliability and correctness are particularly important when ontology-based systems are used in safety-critical applications; in those involving medicine, for example, incorrect reasoning could adversely affect patient care.

In addition to checking for inconsistencies and synonyms, ontology-development tools usually check for implicit subsumption relationships, updating the class hierarchy accordingly. This automated updating is also a useful design aid, allowing ontology developers to focus on class descriptions, leaving the computation of the class hierarchy to the reasoner; it can also be used by developers to check if the hierarchy induced by the class descriptions is consistent with their expert intuition. The two may not be consistent when, for example, errors in the ontology result in unexpected subsumption inferences or "underconstrained" class descriptions result in expected inferences not being found. Not finding expected inferences is common, as it is easy to inadvertently omit axioms that express "obvious" information. For example, an ontology engineer may expect the class of patients with a fracture of both the tibia and the fibula to be a subClassOf "patient with multiple fractures"; however, this relationship may not hold if the ontology doesn't include (explicitly or implicitly) the information that the tibia and fibula are different bones. Failure to find this subsumption relationship should prompt the engineer to add the missing DisjointClasses axiom.

Reasoning is also important when ontologies are deployed in applications, when it is needed to answer standard data-retrieval queries, and to answer conceptual queries about the structure of the domain. For example, biologists use ontologies (such as the Gene Ontology, or GO, and the Biological Pathways Exchange ontology, or BioPAX) to annotate (Web-accessible) data from gene-sequencing experiments, making it possible to answer complex queries (such as "What DNA-binding products interact with insulin receptors?"). Answering requires a reasoner to not only identify individuals that are (perhaps only implicitly) instances of DNA-binding products and of insulin receptors but to identify which pairs of individuals are related (perhaps only implicitly) via the interactsWith property.

Finally, in order to maximize the benefit of reasoning services, tools should be able to explain inferences; without explanations, developers may find it difficult to repair errors in an ontology and may even start to doubt the correctness of inferences. Such an explanation typically involves computing a (hopefully small) subset of the ontology that still entails the inference in question and, if necessary, presenting the user with a chain of reasoning steps.[12] The explanation in Figure 3 (produced by the Protégé 4 ontology-development tool) describes the steps that lead to the inference mentioned earlier with respect to the inconsistency of OceanCrustLayer.

Ontology Applications

The availability of tools and reasoning systems has contributed to the increasingly widespread use of OWL, which has become the de facto standard for ontology development in fields as diverse as biology,[19] medicine,[18] geography,[8] geology (the Semantic Web for Earth and Environmental Terminology project, sweet.jpl.nasa.gov/), agriculture,[20] and defense.[15] Applications of OWL are particularly prevalent in the life sciences where OWL is used by developers of several large biomedical ontologies, including SNOMED, GO, and BioPAX, mentioned earlier, as well as the Foundational Model of Anatomy (sig.biostr.washington.edu/projects/fm/) and the U.S. National Cancer Institute thesaurus (www.cancer.gov/cancertopics/terminologyresourceshow).

The ontologies used in these applications might have been developed specifically for the purpose or without any particular application in mind. Many ontologies are the result of collaborative efforts within a given community aimed at facilitating (Web-based) information sharing and exchange; some commercially developed ontologies are also subject to a license fee. Many OWL ontologies are available on the Web, identified by a URI and should, in principle, be available at that location. There are also several well-known ontology libraries and even ontology search engines (such as SWOOGLE, swoogle.umbc.edu/) that are useful for locating ontologies. In practice, however, applications are invariably built around a predetermined ontology or set of ontologies that are well understood and known to provide suitable coverage of the relevant domains.

The importance of reasoning support in ontology applications was highlighted in a paper describing a project in which the Medical Entities Dictionary (MED), a large ontology (100,210 classes and 261 properties) used at the Columbia Presbyterian Medical Center in New York, was converted to OWL and checked using an OWL reasoner.[13] As reported in the paper, this check revealed "systematic modeling errors" and a significant number of missed subClassOf relationships that, if not corrected, "could have cost the hospital many missing results in various decision-support and infection-control systems that routinely use MED to screen patients."

In another application, an extended version of the SNOMED ontology was checked using an OWL reasoner that found a number of missing subClassOf relationships. This ontology is being used by the U.K. National Health Service (NHS) to provide "a single and comprehensive system of terms, centrally maintained and updated for use in all NHS organizations and in research" and as a key component of its $6.2 billion "Connecting for Health" IT program (www.connectingforhealth.nhs.ukhow). An important feature of the system is that it can be extended to provide more detailed coverage if needed by specialized applications; for example, a specialist allergy clinic may need to distinguish allergies caused by different kinds of nut so may need to add new terms to the ontology (such as AlmondAllergy):

```
Class: AlmondAllergy
  equivalentTo: Allergy and
  causedBy some Almond
```

Using a reasoner to insert this new term into the ontology ensures it is recognized as a subClassOf NutAllergy, something that is clearly of crucial importance for ensuring that patients with an AlmondAllergy are correctly identified in the national records system as patients with a NutAllergy.

Ontologies are also widely used to facilitate the sharing and integration of information. The Neurocommons project (sciencecommons.org/projects/data/) aims to provide a platform for, for example, sharing and integrating knowledge in the neuroscience domain; a key component is an ontology of annotations to be used to integrate available knowledge on the Web, including major neuroscience databases. Similarly, the Open Biomedical Ontologies Foundry (www.obofoundry.org) is a library of ontologies designed to facilitate international information sharing and integration in the biomedical domain. In information-integration ap-

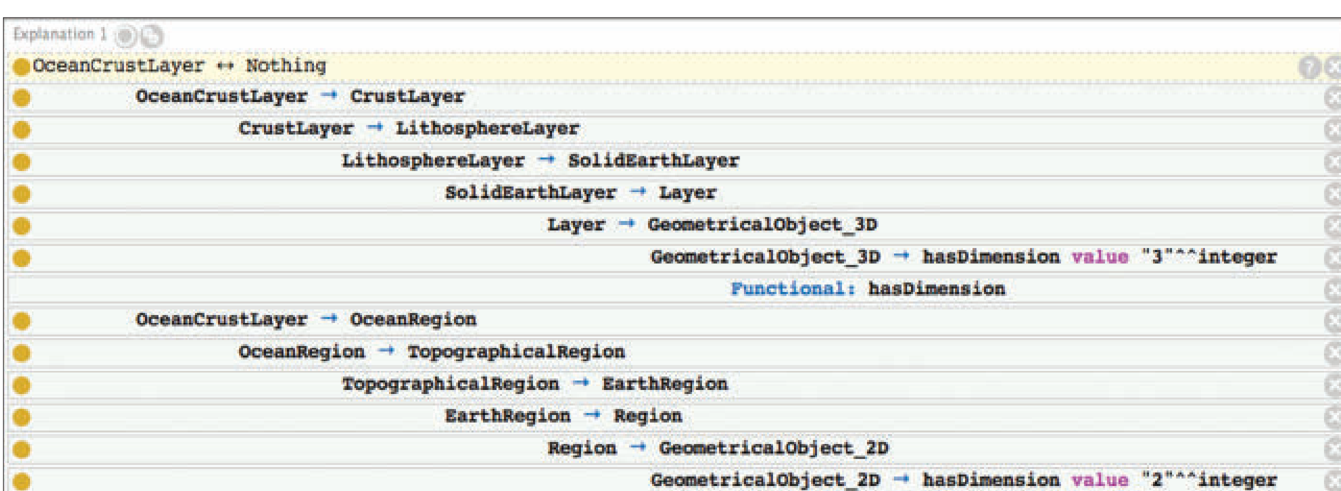
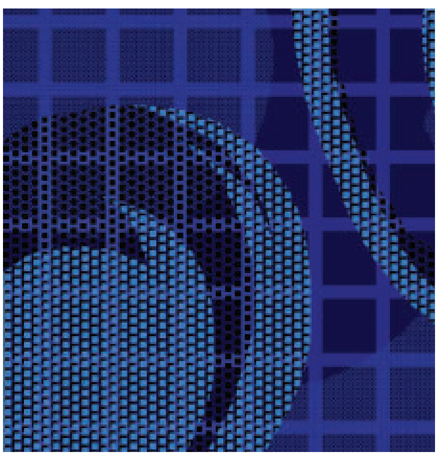


Figure 3: An explanation from Protégé 4.

plications the ontology could play several roles: provide a formally defined and extensible vocabulary for semantic annotations; describe the structure of existing sources and the information they store; and provide a detailed model of the domain against which queries are formulated. Such queries can be answered by using semantic annotations and structural knowledge to retrieve and combine information from multiple sources.[22] It should be noted that the use of ontologies in information integration is far from new and the subject of extensive research within the database community.[2]

With large ontologies, answering conceptual and data-retrieval queries may be a very complex task, and DL



reasoners allow OWL ontology applications to answer complex queries and provide guarantees about the correctness of the result. Reliability and correctness are particularly important when ontology-based systems are used in safety-critical applications; in those involving medicine, for example, incorrect reasoning could adversely affect patient care.

However, RDF and OWL are also used in a range of applications where reasoning plays only a relatively minor role in, for example, the Friend of a Friend, or FOAF, project (www.foaf-project.org) and the Dublin Core Metadata Initiative, (dublincore.org) and when carrying annotations in Adobe's Extensible Metadata Platform (www.adobe.com/products/xmp/). In them, RDF is typically used to provide a flexible and extensible data structure for annotations, with the added advantage that IRIs can be used to refer directly to Web resources.

In FOAF, for example, a simple RDF/OWL ontology provides a vocabulary of terms for describing and linking people and their interests and activities; terms include the `foaf:Person` class and properties, including `foaf:name`, `foaf:homepage`, and `foaf:knows`. OWL is used to declare that some properties (such as `foaf:homepage`) are InverseFunctional; that is, they can be used as a key to identify the subject of the property, often a person. However, the semantics of the vocabulary is mainly captured informally in textual descriptions of each term and procedurally interpreted by applications. This informality reduces the need for reasoning systems but limits the ability of applications to share and understand vocabulary extensions.

Future Directions

The success of OWL also involves many challenges for the future development of both the OWL language and OWL tool support. Central to them is the familiar tension between requirements for advanced features, particularly increased expressive power, and raw performance, particularly the ability to deal with large ontologies and data sets.

Researchers have addressed them by investigating more expressive DLs, developing new and more highly optimized DL reasoning systems and identifying smaller logics that combine still-useful expressive power with better worst-case complexity or other desirable computational properties. Results from these efforts are being exploited by the W3C in order to refine and extend OWL, forming in October 2007 a new W3C Working Group for this purpose (www.w3.org/2007/OWL/). The resulting language is called OWL 2 (initially called OWL 1.1) based on a more expressive DL called SROIQ.[10] OWL 2 extends OWL with the ability to "qualify" cardinality restrictions to, say, describe the hand as having four parts that are fingers and one part that is a thumb; assert that properties are reflexive, irreflexive, asymmetric, and disjoint (such as to describe hasParent as an irreflexive property); and compose properties into property chains (such as to capture the fact that a disease affecting a part of an organ affects the organ as a whole). OWL 2 also provides extended support for datatypes and for annotations.

Besides increasing the expressive power of the complete language, OWL 2 also defines three so-called profiles, in effect language fragments with desirable computational properties (www.w3.org/TR/owl2-profileswww.w3.org/TR/opw12-profiles/). One is based on DL Lite, a logic for which standard reasoning problems can be reduced to standard query language (SQL) query answering; another is based on EL++, a logic for which standard reasoning problems can be performed in polynomial time; and the third is based on DLP, a logic for which query answering can be implemented using rule-based techniques that have been shown to scale well in practice.

In some cases, even the increased expressive power of OWL 2 may not meet application requirements. One way to further increase the expressive power of the language would be to extend it with Horn-like rules; that is, implications like parent $(x, y) \wedge$ bother $(y, z) \Rightarrow$ uncle (x, z) stating that if y is a parent of x and z is a brother of y (the antecedent), then z is an uncle of x (the consequent). A notable proposal along these lines is the Semantic Web Rules Language (www.w3.org/Submission/SWRL/). If the semantics of Horn-like rules is restricted so it applies only to named individuals, then its addition does not disturb the decidability of the underlying DL; this restricted form of rules is known as "DL-safe" rules.[17] A W3C working group was established in 2005 to produce a W3C language standard that will "allow rules to be translated between rule languages and thus transferred between rule systems" (www.w3.org/2005/rules/).

As I discussed earlier, reasoning-enabled tools provide vital support for ontology engineering. Recent work has shown how this support can be extended to modular design and module extraction, important techniques for working with large ontologies. When a team of ontology engineers is developing a large ontology, they should divide it into modules in order to make it easier to understand and facilitate parallel work. Similarly, it may be desirable to extract from a large ontology a module containing all the information relevant to some subset of the domain; the resulting small(er) ontology is easier for humans to understand and applications to use. New reasoning services can be used to alert developers to unanticipated and/or undesirable interactions when modules are integrated and to identify a subset of the original ontology that is indistinguishable from it when used to reason about the relevant subset of the domain.[4]

The availability of an SQL has been an important factor in the success of relational databases, and there have been several proposals for a semantic Web query language. As in the case of RDF and OWL, the W3C in 2004 set up a standardization working group that in January 2008 completed its work on the SPARQL query language standard (www.w3.org/TR/rdf-sparql-query). Strictly speaking, this language is only for RDF, but it is easy to see how it could be extended for use with OWL ontologies, something already happening in practice.

As I mentioned earlier, major research efforts have been directed toward tackling some of the barriers to realizing the semantic Web; considerable progress has been made in such areas as ontology alignment (reconciling ontologies that describe overlapping domains),[18] ontology extraction (extracting ontologies from text),[16] and the automated annotation of both text[6] and images.[5] Of particular interest is the growth of Web 2.0 applications, showing how it might be possible for user communities to collaboratively annotate Web content, as well as create simple forms of ontology via the development of hierarchically organized sets of tags, or folksonomies.[21] Progress has also been made in developing the infrastructure needed to add structured annotations to existing Web resources. For example, in October 2008 the W3C produced a Recommendation for RDFa, a mechanism for embedding RDF in existing XHTML documents (www.w3.org/TR/rdfa-syntax/).

Conclusion

Semantic Web research aims to help Web-accessible information and services be more effectively exploited, particularly by software agents and applications. As a first step, the W3C developed new languages, including RDF and OWL, that allow for the description of Web resources and the representation of knowledge to enable applications to use resources more intelligently.

Although a wide range of semantic Web applications is available today, fully realizing the semantic Web still seems a long way off and would first require the solution of many challenging research problems, including those in knowledge representation and reasoning, databases, computational linguistics, computer vision, and agent systems. Moreover, most of the Web is yet to be semantically annotated, and relatively few ontologies are available (even fewer high-quality ones).

However, semantic Web research already has a major influence on the development and deployment of ontology languages and tools (often called semantic Web technologies). They have become a de facto standard for ontology development and are seeing increased use in research labs, as well as in large-scale IT projects, particularly those where the schema plays an important role, where information has high value, and where information may be incomplete. This emerging role is reflected in extended support for semantic Web technologies, including commercial tools, implementations, and applications, from commercial vendors, including Hewlett-Packard, IBM, Oracle, and Siemens.

Related challenges involve both expressive power and scalability. However, the success of the technologies also motivates research and development efforts in academic institutions and industry to address these challenges; it seems certain these efforts will have a major influence on the future development of information technology.

Acknowledgment

I want to thank Uli Sattler of the University of Manchester and Franz Baader of Dresden Technical University for letting me borrow the idea of using Harry Potter in the ontology examples. C

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P.F., Eds. *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press, Cambridge, U.K., 2003.
2. Batini, C., Lenzerini, M., and Navathe, S.B. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys 18*, 4 (Dec. 1986), 323–364.
3. Berners-Lee, T., Hendler, J., and Lassila, O. The semantic Web. *Scientific American 284*, 5 (May 2001), 34–43.
4. Cuenca Grau, B., Horrocks, I., Kazakov, Y., and Sattler, U. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research 31* (Apr. 2008), 273–318.
5. Datta, R., Joshi, D., Li, J., and Wang, J.Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys 40*, 2 (Apr. 2008).
6. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., and Zien, J.Y. Semtag and seeker: Bootstrapping the semantic Web via automated semantic annotation. In *Proceedings of the 12th International World Wide Web Conference* (Budapest, Hungary, May). ACM Press, New York, 2003, 178–186.
7. Doyle, J. and Patil, R.S. Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence 48*, 3 (Apr. 1991), 261–297.
8. Golbreich, C., Zhang, S., and Bodenreider, O. The foundational model of anatomy in OWL: Experience and perspectives. *Journal of Web Semantics 4*, 3 (Sept. 2006), 181–195.
9. Goodwin, J. Experiences of using OWL at the ordnance survey. In *Proceedings of the First OWL Experiences and Directions Workshop* (Galway, Ireland, Nov.). CEUR-WS, 2005; CEUR-WS.org/Vol-188/.
10. Horrocks, I., Kutz, O., and Sattler, U. The even more irresistible SROIQ. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning* (Lake District, U.K., June). AAAI Press, Menlo Park, CA, 2006, 57–67.
11. Horrocks, I. and Sattler, U. A tableau decision procedure for SHOIQ. *Journal of Automated Reasoning 39*, 3 (Oct. 2007), 249–276.
12. Kalyanpur, A., Parsia, B., Sirin, E., and Hendler, J. Debugging unsatisfiable classes in OWL ontologies. *Journal of Web Semantics 3*, 4 (Dec. 2005), 243–366.
13. Kershenbaum, A., Fokoue, A., Patel, C., Welty, C., Schonberg, E., Cimino, J., Ma, L., Srinivas, K., Schloss, R., and Murdock, J.W. A view of OWL from the field: Use cases and experiences. In *Proceedings of the Second OWL Experiences and Directions Workshop* (Athens, GA, Nov.). CEUR-WS, 2006; CEUR-WS.org/Vol-216/.
14. Klyne, G. and Carroll, J.J. *Resource Description Framework (RDF): Concepts and Abstract syntax.* W3C Recommendation, Feb. 10, 2004; www.w3.org/TR/rdf-concepts/.
15. Lacy, L., Aviles, G., Fraser, K., Gerber, W., Mulvehill, A., and Gaskill, R. Experiences using OWL in military applications. In *Proceedings of the First OWL Experiences and Directions Workshop* (Galway, Ireland, Nov. 2005); CEUR-WS.org/Vol-188/.
16. Maedche, A. and Staab, S. Ontology learning for the semantic Web. *IEEE Intelligent Systems 16*, 2 (Mar./Apr. 2001), 72–79.
17. Motik, B., Sattler, U., and Studer, R. Query answering for OWL-DL with rules. *Journal of Web Semantics 3*, 1 (July 2005), 41–60.
18. Shvaiko, P. and Euzenat, J. A survey of schema-based matching approaches. *Journal on Data Semantics IV, Lecture Notes in Computer Science 3730* (Nov. 2005), 146–171.
19. Sidhu, A., Dillon, T., Chang, E., and Sidhu, B.S. Protein ontology development using OWL. In *Proceedings of the First OWL Experiences and Directions Workshop* (Galway, Ireland, Nov. 2005); CEUR-WS.org/Vol-188/.
20. Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., and Katz, S. Reengineering thesauri for new applications: The AGROVOC example. *Journal of Digital Information 4*, 4 (2004).
21. Spyns, P., de Moor, A., Vandenbussche, J., and Meersman, R. From folksologies to ontologies: How the twain meet. In *Proceedings of On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, Lecture Notes in Computer Science 4275* (Montpellier, France, Oct. 29–Nov. 3). Springer, 2006, 738–755.
22. Stevens, R., Baker, P., Bechhofer, S., Ng, G., Jacoby, A., Paton, N.W., Goble, C.A., and Brass, A. Tambis: Transparent access to multiple bioinformatics information sources. *Bioinformatics 16*, 2 (Feb. 2000), 184–186.

Ian Horrocks (Ian.Horrocks@comlab.ox.ac.uk) is a professor of computer science in the Oxford University Computing Laboratory and a fellow of Oriel College, Oxford, U.K.

DOI:10.1145/1409360.1409378

Targeted IE methods are transforming into open-ended techniques.

BY OREN ETZIONI, MICHELE BANKO, STEPHEN SODERLAND, AND DANIEL S. WELD

Open Information Extraction from the Web

Say you want to select a quiet, centrally located Manhattan hotel. Google returns an overwhelming seven million results in response to the query "new york city hotels." Or, say you are trying to assemble a program committee for an annual conference composed of researchers who have published at the conference in previous years, and to balance it geographically. While today's Web search engines identify potentially relevant documents, you are forced to sift through a long list of URLs, scan each document to identify any pertinent bits of information, and assemble the extracted findings before you can solve your problem.

Over the coming decade, Web searching will increasingly transcend keyword queries in favor of systems that automate the tedious and error-prone task of sifting through documents. Moreover, we will build systems that fuse relevant pieces of information into a coherent overview, thus reducing from hours to minutes the time required to perform complex tasks.

Information extraction (IE)—a venerable technology that maps natural-language text into structured relational data—offers a promising avenue toward this goal. Although extracting data from text is inherently challenging, given the ambiguous and idiosyncratic nature of natural language, substantial progress has been made over the last few decades.

This article surveys a range of IE methods, but we highlight *Open* Information Extraction,[3, 4] wherein the identities of the relations to be extracted are unknown and the billions of documents found on the Web necessitate highly scalable processing.

Information Extraction

At the core of an IE system is an extractor, which processes text; it overlooks irrelevant words and phrases and attempts to home in on entities and the relationships between them. For example, an extractor might map the sentence "Paris is the stylish capital of France" to the relational tuple (Paris, CapitalOf, France), which might be represented in RDF or another formal language.

Considerable knowledge is necessary to accurately extract these tuples from a broad range of text. Existing techniques obtain it in ways ranging from direct knowledge-based encoding (a human enters regular expressions or rules) to supervised learning (a human provides labeled training examples) to self-supervised learning (the system automatically finds and labels its own examples). Here, we briefly survey these methods.

Knowledge-Based Methods. The first IE systems were domain-specific. A series of DARPA Message Understanding Conferences (MUCs) challenged the NLP community to build systems that handled robust extraction from naturally occurring text. The domain

ILLUSTRATION BY GLUEKIT

Figure 1: Training examples and learned patterns for the management-succession domain.

```
<P1> Donald E. Martella
</P1>,formerly vice
president, was named
<POS> president </POS>
of <ORG> Topologix,
Inc </ORG>. <P1> He </
P1> succeeds <P2> Jack
Harper </P2>, a company
founder who was named
chairman.

<P1> Mr. Smith </P1>
succeeds <P2> Jack
Harper </P2>.
```

Supervised Machine Learning →

```
"<P1> was named
<POS> of <ORG>"

"<P1> succeeds <P2>"
```

Figure 2: Sample Wikipedia infobox and the attribute/value data used to generate it.


```
{{Infobox Settlement
|official_name = Běijīng
|other_name = 北京
|native_name =

|settlement_type = [[Municipality of
        China|Municipality
|image_skyline = SA Temple of Heaven.jpg
|image_caption = The [[Temple of
        Heaven]], a symbol of Beijing
|citylogo_size =
|image_map = China-Beijing.png
|mapsize = 275px
|map_caption = Location within China
|subdivision_type = Country
|subdivision_name = [[People's Republic of China]]
|subdivision_type1 = [[Political divisions of
        China#County level|Countylevel&
        nbsp;divisions]]
|subdivision_name1 = 18
|subdivision_type2 = [[Political divisions of
        China#Township
        level|Township divisions]]
|subdivision_name2 = 273
|leader_title =[[Communist Party of
        China|CPC]] Beijing
|leader_name =[[Liu Qi (Communist)|Liu Qi]]
        Committee Secretary
|leader_title1 = [[Mayor]]
|leader_name1 =[[Wang Qishan]]
|established_title = Settled
|established_date = ca. 473 BC
...
}}
```

of MUC-3 and MUC-4 was Latin-American Terrorism;[2] and the task was to fill templates with information about specific terrorist actions, with fields for the type of event, date, location, perpetrators, weapons, victims, and physical targets. Subsequent MUC conferences focused on domains such as joint ventures, microelectronics, or management succession.

The first IE systems relied on some form of pattern-matching rules that were manually crafted for each domain. Rules that assigned the semantic class *PhysicalTarget* space to the term bank in the terrorism domain, for example, needed to be altered to identify instances of the class *Corporation* in the joint-ventures domain. These systems were clearly not scalable or portable across domains.

Supervised Methods. Modern IE, beginning with the works of Soderland,[21, 22] Riloff,[17] and Kim and Moldovan,[11] automatically learns an extractor from a training set in which domain-specific examples have been tagged. With this machine-learning approach, an IE system uses a domain-independent architecture and sentence analyzer. When the examples are fed to machine-learning methods, domain-specific extraction patterns can be automatically learned and used to extract facts from text. Figure 1 shows an example of such extraction rules, learned to recognize persons moving into and out of top corporate-management positions.

The development of suitable training data for IE requires substantial effort and expertise. DIPRE,[5] Snowball,[1] and Meta-Bootstrapping[18] sought to address this problem by reducing the amount of manual labor necessary to perform relation-specific extraction. Rather than demand hand-tagged corpora, these systems required a user to specify relation-specific knowledge through either of the following: a small set of seed instances known to satisfy the relation of interest; or a set of hand-constructed extraction patterns to begin the training process. For instance, by specifying the set *Bolivia*, *city*, *Colombia*, *district*, *Nicaragua* over a corpus in the terrorism domain, these IE systems learned patterns (for example, *headquartered in <x>, to occupy <x>, and shot in <x>*) that identified additional names of locations. Recent advances include automatic induction of features when learning conditional random fields[13] and high-level specification of extraction frameworks using Markov logic networks.[14] Nevertheless, the amount of manual effort still scales linearly with the number of relations of interest, and these target relations must be specified in advance.

Self-Supervised Methods. The KnowItAll Web IE system[9] took the next step in automating IE by learning to label its own training examples using only a small set of domain-independent extraction patterns. KnowItAll was the first published system to carry out extraction from Web pages that was unsupervised, domain-independent, and large-scale.

For a given relation, the set of generic patterns was used to automatically instantiate relation-specific extraction rules, which were then used to learn domain-specific extraction rules. The rules were applied to Web pages identified via search-engine queries, and the resulting extractions were assigned a probability using information-theoretic measures derived from search engine hit counts. For example, KnowItAll utilized generic extraction patterns like "<*X*> is a <*Y*>" to find a list of candidate members *X* of the class *Y*. When this pattern was used, say, for the class *Country*, it would match a sentence that included the phrase "*X* is a country."

Next, KnowItAll used frequency statistics computed by querying search engines to identify which instantiations were most likely to be bona fide members of the class. For example, in order to estimate the likelihood that "China" was the name of a country, KnowItAll used automatically generated phrases associated with the class *Country* to see if there was a high correlation between the numbers of documents containing the word "China" and those containing the phrase "countries such as." Thus KnowItAll was able to confidently label China, France, and India as members of the class *Country* while correctly knowing that the existence of the sentence, "Garth Brooks is a country singer" did not provide sufficient evidence that "Garth Brooks" is the name of a country.[7] Moreover, KnowItAll learned a set of relation-specific extraction patterns (for example, "capital of <country>") that led it to extract additional countries, and so on.

KnowItAll is self-supervised; instead of utilizing hand-tagged training data, the system selects and labels its own training examples and iteratively bootstraps its learning process. But while self-supervised systems are a species of unsupervised systems, unlike classic unsupervised systems they do utilize labeled examples and do form classifiers whose accuracy can be measured using standard metrics. Instead of relying on hand-tagged data, self-supervised systems autonomously "roll their own" labeled examples. (See Feldman[10] for discussion of an additional self-supervised IE system inspired by KnowItAll.)

While self-supervised, KnowItAll is relation-specific. It requires a laborious bootstrapping process for each relation of interest, and the set of relations has to be named by the human user in advance. This is a significant obstacle to open-ended extraction because unanticipated concepts and relations are often encountered while processing text.

The Intelligence in Wikipedia (IWP) project[23] uses a different form of self-supervised learning to train its extractors. IWP bootstraps from the Wikipedia corpus, exploiting the fact that each article corresponds to a primary object and that many articles contain infoboxes—tabular summaries of the most important attributes (and their values) of these objects. For example, Figure 2 shows the "Beijing" infobox for the class *Settlement* that was dynamically generated from the accompanying attribute/value data.

IWP is able to use Wikipedia pages with infoboxes as training data in order to learn classifiers for page type. By using the values of infobox attributes to match sentences in the article, IWP can train extractors for the various attributes. Further, IWP can autonomously learn a taxonomy over infobox classes, construct schema mappings between the attributes of parent/child classes, and thus use shrinkage to improve both recall and precision. Once extractors have been successfully learned, IWP can extract values from general Web pages in order to complement Wikipedia with additional content.

Open Information Extraction

While most IE work has focused on a small number of relations in specific preselected domains, certain corpora—encyclopedias, news stories, email, and the Web itself—are unlikely to be amenable to these methods. Under such circumstances, the relations of interest are both numerous and serendipitous—they are not known in advance. In addition, the Web corpus contains billions of documents, necessitating highly scalable extraction techniques.

The challenge of Web extraction led us to focus on Open Information Extraction (Open IE), a novel extraction paradigm that tackles an unbounded number of relations, eschews domain-specific training data, and scales linearly (with low constant factor) to handle Web-scale corpora.

For example, an Open IE system might operate in two phases. First, it would learn a general model of how relations are expressed in a particular language. Second, it could utilize this model as the basis of a relation-independent extractor whose sole input is a corpus and whose output is a set of extracted tuples that are instances of a potentially unbounded set of relations. Such an Open IE system would learn a general model of *how* relations are expressed (in a particular language), based on unlexicalized features such as part-of-speech tags (for example, the identification of a verb in the surrounding context) and domain-independent regular expressions (for example, the presence of capitalization and punctuation).

Is there a general model of relationships in English, though? To address this question we examined a sample of 500 sentences selected at random from the IE training corpus developed by Bunescu and Mooney.[6] We found that most relationships expressed in this sample could in fact be characterized by a compact set of relation-independent patterns. See Table 1 for these patterns and an estimate of their frequency.[a] In contrast, traditional IE methods learn lexical models of individual relations from hand-labeled examples of sentences that express these relations. Such an IE system might learn that the presence of the phrase "headquarters located in" indicates an instance of the headquarters relation. But lexical features are relation-specific. When using the Web as a corpus, the relations of interest are not known prior to extraction, and their number is immense. Thus an Open IE system cannot rely on hand-labeled examples of each relation. Table 2 summarizes the differences between traditional and Open IE.

Systems such as KnowItAll and IWP may be seen as steps in the direction of Open IE, but the former didn't scale as well as desired and the latter seems incapable of extracting more than 40,000 relations. Knext[19] appears to fit the Open IE paradigm,

a For simplicity, we restricted our study to binary relationships.

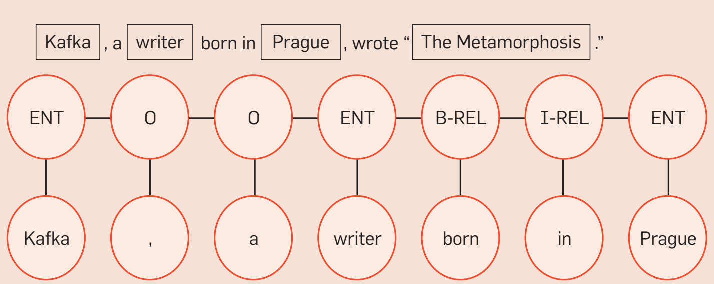
Table 1: Taxonomy of binary relationships. Nearly 95% of 500 randomly selected sentences belong to one of the eight categories noted here.

Relative Frequency	Category	Simplified Lexico-Syntactic Pattern
37.8	Verb	E_1 Verb E_2 *X established Y*
22.8	Noun + Prep	E_1 NP Prep E_2 *X settlement with Y*
16.0	Verb + Prep	E_1 Verb Prep E_2 *X moved to Y*
9.4	Infinitive	E_1 to Verb E_2 *X plans to acquire Y*
5.2	Modifier	E_1 Verb E_2 Noun *X is Y winner*
1.8	Coordinate$_n$	E_1 (and|,|-|:) E_2 NP *X-Y deal*
1.0	Coordinate$_v$	E_1 (and|,) E_2 Verb *X , Y merge*
0.8	Appositive	E_1 NP (:|,)? E_2 *X hometown : Y*

Table 2: The contrast between traditional and open IE.

	Traditional IE	Open IE
Input	Corpus + Labeled Data	Corpus + Domain-Independent Methods
Relations	Specified In Advance	Discovered Automatically
Complexity	$O(D * R)$ *D* documents, *R* relations	$O(D)$ *D* documents

Figure 3: Information extraction as sequence labeling. A CRF is used to identify the relationship, *born in*, between *Kafka and Prague*. Entities are labeled as ENT. The B-REL label indicates the start of a relation, with I-REL indicating the continuation of the sequence.



but its precision, recall, and scalability have yet to be demonstrated.

The TextRunner System

TextRunner[3, 4] is a fully implemented Open IE system that utilizes the two-phase architecture described here.[b]

b See www.cs.washington.edu/research/textrunner

TextRunner extracts high-quality information from sentences in a scalable and general manner. Instead of requiring relations to be specified in its input, TextRunner learns the relations, classes, and entities from its corpus using its relation-independent extraction model.

TextRunner's first phase uses a general model of language. Specifically, it trains a graphical model called a conditional random field (CRF)[12] to maximize the conditional probability of a finite set of labels, given a set of input observations. By making a first-order Markov assumption about the dependencies among the output variables, and thus arranging variables sequentially in a linear chain, extraction can be treated as a sequence-labeling problem. Using a CRF, the extractor learns to assign labels to each of the words in a sentence denoting the beginning and end both of entity names and the relationship string.[c] See Figure 3 for an illustration.

In the second phase, TextRunner's extractor scans sentences linearly and rapidly extracts one or more textual triples that aim to capture (some of) the relationships in each sentence. For example, given the sentence "Kafka, a writer born in Prague, wrote *The Metamorphosis*," the extractor forms the triple (*Kafka*, *born in*, *Prague*). The triple consists of three strings, in which the first and third are meant to denote entities and the second to denote the relationship between them.

Of course, there are many subtleties to successful extraction from a corpus as large and heterogeneous as the Web. First, the same entities may be referred to by a variety of names (for example, *Edison*, *Thomas Edison*, *Thomas Alva Edison*, and so on). Second, the same string (say, *John Smith*) may refer to different entities. Third, vagaries of natural language (such as pronoun resolution, metaphor, anaphora, complex or ungrammatical sentences) have to be unraveled to correctly extract information. Fourth, the Web is rife with incorrect information (for example, *Elvis killed JFK*). In fact, there are many more challenges that we do not have room to discuss here, though we have addressed some of them in our research. For instance, the Resolver system[24] computes the probability that two strings are synonymous based on a highly scalable and unsupervised analysis of TextRunner tuples. Numerous other issues remain

c Although TextRunner has initially focused on extracting binary relationships, its model structure can be extended to identify relationships with greater arity.

as open problems for future work.

Post-extraction, TextRunner's collection of triples is made efficiently searchable by using Lucene, a high-performance indexing and search engine.[d] Thus TextRunner can be queried for tuples containing particular entities (for example, *Edison*), relationships (*invented*), or relationships between two entities (such as *Microsoft* and *IBM*). The different triples returned in response to a query are ranked by a fairly complex formula, but a key parameter that boosts ranking is the number of times a tuple has been extracted from the Web. Because the Web corpus is highly redundant, we have found that repeated extractions are strongly correlated with increased likelihood that an extraction is correct.

We have run TextRunner on a collection of over 120 million Web pages and extracted over 500 million tuples. By analyzing random samples of the output, we have determined that the precision of the extraction process exceeds 75% on average.[4] In collaboration with Google, we have also run a version of TextRunner on over one billion pages of public Web pages and have found that the use of an order-of-magnitude larger corpus boosts both precision and recall. Other researchers have investigated techniques closely related to Open IE, but at a substantially smaller scale.[20, 23]

Applications of Open IE

IE has numerous applications, but some tasks require the full power of Open IE because of the scope and diversity of information to be extracted. This diversity is often referred to as the "long tail" to reflect the distribution of information requests—some are very common but most are issued infrequently.

We consider three such tasks here. First and foremost is "question answering," the task of succinctly providing an answer to a user's factual question. In Figure 4, for example, the question is "What kills bacteria?" It turns out that the most comprehensive answer to that question is produced by collecting information across thousands of Web sites that address this topic. Using Open IE, the range of questions TextRunner can address mirrors the unbounded scope and diversity of its Web corpus.

The two additional tasks are:

- "Opinion mining," in which Open IE can extract opinion information about particular objects (including products, political candidates, and more) that are contained in blog posts, reviews, and other texts.
- "Fact checking," in which Open IE can identify assertions that directly or indirectly conflict with the body of knowledge extracted from the Web and various other knowledge bases.

Opinion Mining is the process of taking a corpus of text expressing multiple opinions about a particular set of entities and creating a coherent overview of those opinions. Through this process, opinions are labeled as positive or negative, salient attributes of the entities are identified, and specific sentiments about each attribute are extracted and compared.

In the special case of mining product reviews, opinion mining can be decomposed into the following main subtasks, originally described in Popescu:[15]

1. *Identify product features.* In a given review, features can be explicit (for example, "the size is too big") or implicit ("the scanner is slow).

2. *Identify opinions regarding product features.* For example, "the size is too big" contains the opinion phrase "too big," which corresponds to the "size" feature.

3. *Determine the polarity of opinions.* Opinions can be positive (for example, "this scanner is so great") or negative ("this scanner is a complete disappointment").

4. *Rank opinions based on their strength.* "Horrible," say, is a stronger adjective than "bad."

Opine[16] is an unsupervised information-extraction system that embodies solutions to all of the mentioned subtasks. It relies on Open IE techniques to address the broad and diverse range of products without requiring hand-tagged examples of each type of product. Opine was the first to report its precision and recall on the tasks of opinion-phrase extraction and opinion-polarity determination in the context of known product features and sentences. When tested on hotels and consumer electronics, Opine was found to extract opinions with a precision of 79% and a recall of 76%. The polarity of opinions could be identified by Opine with a precision of 86% and a recall of 89%.

Fact Checking. Spell checkers and grammar checkers are word-processing utilities that we have come to take

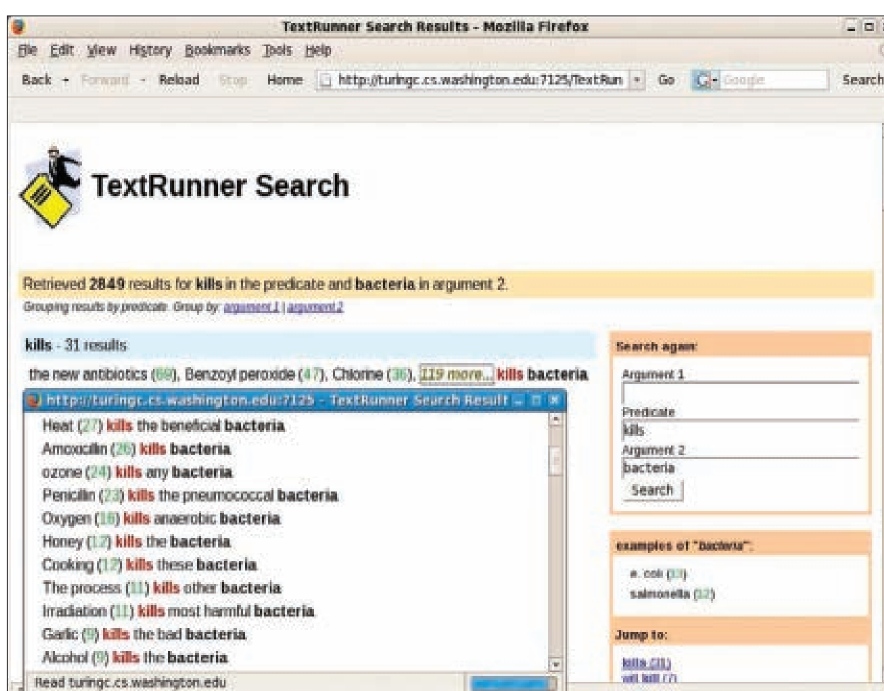


Figure 4: TextRunner aggregates answers to the query "What kills bacteria?"

d http://lucene.apache.org/

for granted. A fact checker based on Open IE seems like a natural next step.[e]

Consider a schoolchild incorrectly identifying the capital of North Dakota, or the date of India's independence, in her homework. The fact checker could automatically detect the error and underline the erroneous sentence in blue.[f] Right-clicking on the underlined sentence would bring up the conflicting facts that led the checker to its conclusion.

Where would the fact checker's knowledge base originate? While resources such as WordNet and the CIA World Fact book are of high quality, they are inherently limited in scope because of the labor-intensive process by which they are compiled. Even Wikipedia, which is put together by a large number of volunteers, only had about two million articles at last count—and they were not guaranteed to contain accurate information. To provide a checker with broad scope, it is natural to use all of the above but also include information extracted from the Web via Open IE.

Of course, the use of information extracted from the Web increases the chance that a correct fact will be flagged as erroneous. Again, this is similar to utilities such as the spell checker and grammar checker, which also periodically misidentify words or sentences as incorrect. Our goal, of course, is to build fact checkers with high precision and recall. In addition, when a fact is flagged as potentially incorrect, the checker provides an easy means of accessing the source of the information that led it to this determination.

Conclusion and Directions for Future Work

This article sketched the transformation of information extraction (IE) from a targeted method, appropriate for finding instances of a particular relationship in text, to an open-ended method (which we call "Open IE") that scales to the entire Web and can support a broad range of unanticipated questions over arbitrary relations. Open IE also supports aggregating, or "fusing," information across a large number of Web pages in order to provide comprehensive answers to questions such as "What do people think about the Thinkpad laptops?" in the Opine system[15] or "What kills bacteria?" in Figure 4.

We expect future work to improve both the precision and recall of Open IE (for example, see Downey[8] and Yates[24]). We have begun to integrate Open IE with inference, which would enable an Open IE system to reason based on the facts and generalizations it extracts from text. The challenge, of course, is to make this reasoning process tractable in the face of billions of facts and rules. We foresee opportunities to unify Open IE with information provided by ontologies such as WordNet and Cyc, as well as with human-contributed knowledge in OpenMind and FreeBase, in order to improve the quality of extracted information and facilitate reasoning. Finally, we foresee the application of Open IE to other languages besides English.

Acknowledgments

This research was supported in part by NSF grants IIS-0535284 and IIS-0312988, ONR grant N00014-08-1-0431, SRI CALO grant 03-000225, and the WRF/TJ Cable Professorship, as well as by gifts from Google. It was carried out at the University of Washington's Turing Center. We wish to thank the members of AI and KnowItAll groups for many fruitful discussions. C

e This idea comes from Krzysztof Gajos.

f Blue is used to distinguish its findings from the red underline for misspellings and the green underline for grammatical errors.

References

1. Agichtein, E. and Gravano, L. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries* (2000).
2. ARPA. *Proceedings of the 3rd Message Understanding Conference* (1991).
3. Banko, M., Cafarella, M., Soderland, S., Broadhead, M. and Etzioni, O. Open information extraction from the Web. In *Proceedings of the International Joint Conference on Artificial Intelligence* (2007).
4. Banko, M. and Etzioni, O. The tradeoffs between traditional and open relation extraction. In *Proceedings of the Association of Computational Linguistics* (2008).
5. Brin, S. Extracting patterns and relations from the World Wide Web. In *Proceedings of the Workshop at the 6th International Conference on Extending Database Technology*, (Valencia, Spain, 1998), 172–183.
6. Bunescu, R. and Mooney, R. Learning to extract relations from the Web using minimal supervision. In *Proceedings of the Association of Computational Linguistics* (2007).
7. Downey, D., Etzioni, O. and Soderland, S. A probabilistic model of redundancy in information extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence* (2005).
8. Downey, D., Schoenmackers, S. and Etzioni, O. Sparse information extraction: Unsupervised language models to the rescue. In *Proceedings of the Association of Computational Linguistics* (2007).
9. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D. and Yates, A. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence 165*, 1 (2005), 91–134.
10. Feldman, R., Rosenfeld, B., Soderland, S. and Etzioni, O. Self-supervised relation extraction from the Web. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems* (2006), 755–764.
11. Kim, J. and Moldovan, D. Acquisition of semantic patterns for information extraction from corpora. In *Proceedings of the 9th IEEE Conference on Artificial Intelligence for Applications* (1993), 171–176.
12. Lafferty, J., McCallum, A. and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 2001 International Conference on Machine Learning.*
13. McCallum, A. Efficiently inducing features of conditional random fields. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence* (Acapulco, 2003), 403–410.
14. Poon, H. and Domingos, P. Joint inference in information extraction. In *Proceedings of the 22nd National Conference on Artificial Intelligence* (2007), 913–918.
15. Popescu, A. and Etzioni, O. Extracting product features and opinions from reviews. In *Proceedings of the Empirical Methods on Natural Language Processing Conference* (2005).
16. Popescu. A-M. Information extraction from unstructured Web text. Ph.D. thesis, University of Washington (2007).
17. Riloff, E. Automatically constructing extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence* (1996), 1044–1049.
18. Riloff, E. and Jones, R. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the AAAI-99 Conference* (1999), 1044–1049.
19. Schubert, L. Can we derive general world knowledge from texts? In *Proceedings of the Human Language Technology Conference* (2002).
20. Shinyama, Y. and Sekine, S. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology/NAACL Conference* (2006).
21. Soderland, S. Learning information extraction rules for semi-structured and free text. *Machine Learning 34*, 1–3 (1999), 233–272.
22. Soderland, S., Fisher, D., Aseltine, J. and Lehnert, W. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (1995), 1314–1321.
23. Weld, D., Wu, F., Adar, E., Amershi, S., Fogarty, J., Hoffmann, R., Patel, K. and Skinner, M. Intelligence in Wikipedia. In *Proceedings of the 23rd Conference on Artificial Intelligence* (2008).
24. Yates, A. and Etzioni, O. Unsupervised resolution of objects and relations on the Web. In *Proceedings of the Human-Language Technology Conference* (2007).

Oren Etzioni (etzioni@cs.washington.edu) is a professor of computer science and the founder and director of the Turing Center at the University of Washington, Seattle.

Michele Banko (banko@cs.washington.edu) is a Ph.D. candidate at the University of Washington, Seattle.

Stephen Soderland (soderland@cs.washington.edu) is a research scientist in the Department of Computer Science and Engineering at the University of Washington, Seattle.

Daniel S. Weld (weld@cs.washington.edu) is the Thomas J. Cable/WRF Professor of Computer Science and Engineering at the University of Washington, Seattle.

research highlights

DOI:10.1145/1409360.1409379

Technical Perspective
One Size Fits All: An Idea Whose Time has Come and Gone

By Michael Stonebraker

BEGINNING IN THE early to mid-1980s the relational model of data has dominated the DBMS landscape. Moreover, descendents of the early relational prototypes (System R and Ingres) have become the primary commercial relational DBMSs. As such, the basic architecture sold by the commercial vendors is more than two decades old. In the meantime the computers have advanced dramatically on which DBMSs are deployed. Grids (blades) have replaced shared memory multiprocessors, CPU speeds have greatly increased, main memory has gotten much bigger and faster, and disks have gotten a lot bigger (but have lagged CPUs in bandwidth increase).

During the same period, several new major applications of DBMS technology have emerged to complement the business data processing market for which RDBMSs were originally designed. These include data warehouses, semi-structured data, and scientific data.

It now seems apparent that the traditional architecture of RDBMSs can be beaten significantly (a factor of 25–50) by a specialized implementation in every major DBMS market. In the data warehouse area, this implementation appears to be a coded column store. A column store represents data column-by-column rather than the traditional row-by-row. In a column architecture the execution engine must read only those data elements relevant to the query at hand, rather than all data elements. Also, data compression is much more effective in a column store because one is compressing only one type of data on a storage block rather than several. As a result, less data is brought from disk to main memory. Moreover, if the execution engine operates on compressed data, then there is less copying and better L2 cache utilization. Hence, CPU execution time is dramatically reduced. These savings have been realized in the original column stores from the 1990s (MonetDB and SybaseIQ) as well as by more recent commercial products from Vertica, Infobright, and Paraccel.

The research presented here by Boncz, Manegold, and Kersten documents these advantages, and is definitely worth reading. It focuses on column execution and compression in main memory and complements other analyses of data warehouse disk behavior. As such, it is exemplary of a collection of recent papers on column store implementation techniques (in VLDB and SIGMOD) to which the interested reader can turn for other analyses.

In other database markets, including business data processing, specialized architectures offer similar advantages. Papers analyzing early prototypes in these areas are beginning to appear. In my opinion, we are seeing "the beginning of the end" of the "one-size-fits-all" systems sold by the major DBMS vendors. I expect specialized architectures to become dominant in several DBMS application areas over the next couple of decades for performance-conscious users. On the other hand, at the low-end open source systems such as MySQL, Postgres, and Ingres are gaining traction.

Expect to see a flurry of additional papers exploring facets of specialized architectures from the DBMS research community. Furthermore, there have been a collection of recent DBMS startups with specialized implementations, and I expect there will be more to come.

It should be clear that the DBMS community is in transition from "the old" to "the new." The next decade should be a period of vibrant activity in our field.

Michael Stonebraker (stonebraker@csail.mit.edu) is an adjunct professor in the Electrical Engineering and Computer Science Department at MIT, Cambridge, MA, and the chief technology officer of Vertica Systems, Inc., and Byledge Corp.

DOI:10.1145/1409360.1409380

Breaking the Memory Wall in MonetDB

By Peter A. Boncz, Martin L. Kersten, and Stefan Manegold

Abstract

In the past decades, advances in speed of commodity CPUs have far outpaced advances in RAM latency. Main-memory access has therefore become a performance bottleneck for many computer applications; a phenomenon that is widely known as the "memory wall." In this paper, we report how research around the MonetDB database system has led to a redesign of database architecture in order to take advantage of modern hardware, and in particular to avoid hitting the memory wall. This encompasses (i) a redesign of the query execution model to better exploit pipelined CPU architectures and CPU instruction caches; (ii) the use of columnar rather than row-wise data storage to better exploit CPU data caches; (iii) the design of new *cache-conscious* query processing algorithms; and (iv) the design and automatic calibration of memory cost models to choose and tune these cache-conscious algorithms in the query optimizer.

1. INTRODUCTION

Database systems have become pervasive components in the information technology landscape, and this importance continues to drive an active database research community, both academic and industrial. Our focus here is on so-called *architecture-conscious* database research that studies the data management challenges and opportunities offered by advances in computer architecture. This area of research started receiving impetus 10 years ago[1,2] when it became clear that database technology was strongly affected by the emergence of the "memory wall"—the growing imbalance between CPU clock-speed and RAM latency.

Database technology, as still employed in the majority of today's commercial systems, was designed for hardware of the 1970–1980s and application characteristics that existed at the time. This translates into the assumption of disk I/O being the dominating performance factor, and an architecture tuned to supporting so-called online transaction processing (OLTP) workloads. That is, sustaining simple lookup/update queries at high throughput. In contrast, modern hardware has since become orders of magnitude faster but also orders of magnitude more complex, and critical database applications now include—besides OLTP—the online analysis of huge data volumes stored in data warehouses, driven by tools that explore hidden trends in the data, such as online analytical processing (OLAP) tools that visualize databases as multidimensional cubes, and data mining tools that automatically construct knowledge models over these huge data-sets. This changed situation has recently made the research community realize that database architecture as it used to be is up for a full rewrite,[21] and to make future systems *self-tuning* to data distributions and workloads as they appear.[4]

In this paper, we summarize the work in the MonetDB[a] project that has focused on redefining database architecture by optimizing its major components (data storage, query processing algebra, query execution model, query processing algorithms, and query optimizer) toward better use of modern hardware in database applications that analyze large data volumes. One of the primary goals in this work has been breaking the memory wall.

Our focus here is the following innovations:

Vertical storage: Whereas traditionally, relational database systems store data in a row-wise fashion (which favors single record lookups), MonetDB uses columnar storage which favors analysis queries by better using CPU cache lines.

Bulk query algebra: Much like the CISC versus RISC idea applied to CPU design, the MonetDB algebra is deliberately simplified with respect to the traditional relational set algebra to allow for much faster implementation on modern hardware.

Cache-conscious algorithms: The crucial aspect in overcoming the memory wall is good use of CPU caches, for which careful tuning of memory access patterns is needed. This called for a new breed of query processing algorithms, of which we illustrate radix-partitioned hash-join in some detail.

Memory access cost modeling: For query optimization to work in a cache-conscious environment, we developed a methodology for creating cost models that takes the cost of memory access into account. In order to work on diverse computer architectures, these models are parameterized at runtime using automatic *calibration* techniques.

2. PRELIMINARIES

Computer architecture evolution in the past decades has had many facets. A general trend is that "latency lags bandwidth,"[16] which holds for both magnetic disk and RAM. This

[a] MonetDB is distributed using a nonrestrictive open-source license, see http://monetdb.cwi.nl

A previous version of this paper entitled "Database Architecture Optimized for the New Bottleneck: Memory Access" appeared in the *Proceedings of the International Conference on Very Large Data Bases* (September 1999), pp. 54–65.

has profoundly influenced the database area and indeed our work on MonetDB.

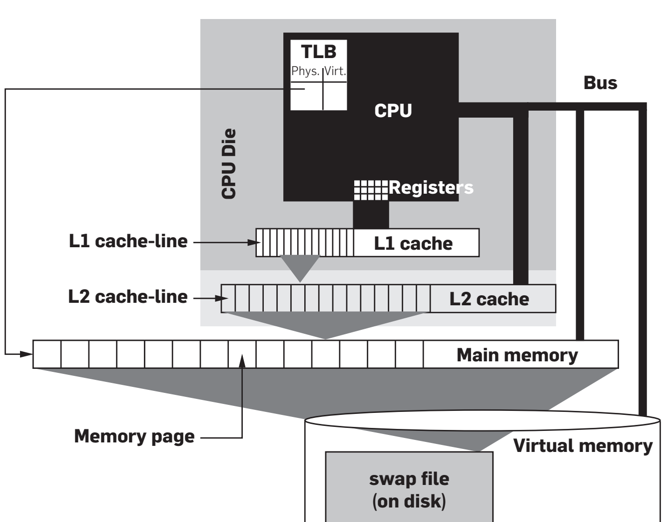
Another facet is that predictable array-wise processing models have been strongly favored in a string of recent CPU architectural innovations. While the rule "make the common case fast" was exploited time and time again to design and construct ever more complex CPUs, the difference in performance efficiency achieved by optimized code and intended use (e.g., "multimedia applications") versus nonoptimized code and nonintended use (e.g., "legacy database applications") has become very significant. A concrete example is the evolution of CPUs from executing a single instruction per clock cycle, to multi-issue CPUs that use deeply pipelined execution; sometimes splitting instructions in more than 30 dependent stages. Program code that has a high degree of independence and predictability (multimedia or matrix calculations) fills the pipelines of modern CPUs perfectly, while code with many dependencies (e.g., traversing a hash-table or B-tree) with unpredictable if-then-else checks, leaves many holes in the CPU pipelines, achieving much lower throughput.

2.1. The memory hierarchy

The main memory of computers consists of *dynamic random access memory* (DRAM) chips. While CPU clock-speeds have been increasing rapidly, DRAM access latency has hardly improved in the past 20 years. Reading DRAM memory took 1–2 cycles in the early 1980s, currently it can take more than 300 cycles. Since typically one in three program instructions is a memory load/store, this "memory wall" can in the worst case reduce efficiency of modern CPUs by two orders of magnitude. Typical system monitoring tools (top, or Windows Task manager) do not provide insight in this performance aspect, a 100% busy CPU could be 95% memory stalled.

To hide the high DRAM latency, the memory hierarchy has been extended with *cache memories* (cf., Figure 1), typically located on the CPU chip itself. The fundamental principle of all cache architectures is *reference locality*, i.e., the assumption that at any time the CPU repeatedly accesses only a limited amount of data that fits in the cache. Only the first access is "slow," as the data has to be loaded from main memory, i.e., a *compulsory cache miss*. Subsequent accesses (to the same data or memory addresses) are then "fast" as the data is then available in the cache. This is called a *cache hit*. The fraction of memory accesses that can be fulfilled from the cache is called *cache hit rate*.

Figure 1: Hierarchical memory architecture.



Cache memories are organized in *multiple cascading levels* between the main memory and the CPU. They become faster, but smaller, the closer they are to the CPU. In the remainder we assume a typical system with two cache levels (L1 and L2). However, the discussion can easily be generalized to an arbitrary number of cascading cache levels in a straightforward way.

In practice, cache memories keep not only the most recently accessed data, but also the instructions that are currently being executed. Therefore, almost all systems nowadays implement two separate L1 caches, a read-only one for instructions and a read-write one for data. The L2 cache, however, is usually a single "unified" read-write cache used for both instructions and data.

A number of fundamental characteristics and parameters of cache memories are relevant for the sequel:

Capacity (*C*). A cache's capacity defines its total size in bytes. Typical cache sizes range from 32KB to 4MB.

Line size (*Z*). Caches are organized in *cache lines*, which represent the smallest unit of transfer between adjacent cache levels. Whenever a cache miss occurs, a complete cache line (i.e., multiple consecutive words) is loaded from the next cache level or from main memory, transferring all bits in the cache line in parallel over a wide bus. This exploits spatial locality, increasing the chances of cache hits for future references to data that is "close to" the reference that caused a cache miss. The typical cache-line size is 64 bytes.

Associativity (*A*). An *A-way set associative* cache allows loading a line into one of *A* different positions. If $A > 1$, some *cache replacement* policy chooses one from the *A* candidates. *Least recently used* (*LRU*) is the most common replacement algorithm. In case $A = 1$, the cache is called *directly mapped*. This organization causes the least (virtually no) overhead in determining the cache-line candidate. However, it also offers the least flexibility and may cause a lot of so-called *conflict misses*. The other extreme case is *fully associative* caches. Here, each memory address can be loaded to any line in the cache ($A = \#$). This avoids conflict misses, and only so-called *capacity misses* occur as the cache capacity gets exceeded. However, determining the cache-line candidate in this strategy causes a relatively high overhead that increases with the cache size. Hence, it is feasible only for smaller caches. Current PCs and workstations typically implement two- to eight-way set associative caches.

Latency (λ) is the time span from issuing a data access until the result is available in the CPU. Accessing data that is already available in the L1 cache causes *L1 access latency* (λ_{L1}), which is typically rather small (1 or 2 CPU cycles). In case the requested data is not found in L1, an *L1 miss* occurs, additionally delaying the data access by *L2 access latency* (λ_{L2})

for accessing the L2 cache. Analogously, if the data is not yet available in L2, an *L2 miss* occurs, further delaying the access by *memory access latency* (λ_{Mem}) to finally load the data from main memory. Hence, the total latency to access data that is in neither cache is $\lambda_{Mem} + \lambda_{L2} + \lambda_{L1}$. As mentioned above, all current hardware actually transfers multiple consecutive words, i.e., a complete cache line, during this time.

Bandwidth (β) is a metric for the data volume (in megabytes) that can be transferred to the CPU per second. The different bandwidths are referred to as *L2 access bandwidth* (β_{L2}) and *memory access bandwidth* (β_{Mem}), respectively. Memory bandwidth used to be simply the cache-line size divided by the memory latency. Modern multiprocessor systems provide excess bandwidth capacity $\beta' \geq \beta$. To exploit this, caches need to be *nonblocking*, i.e., they need to allow more than one outstanding memory load at a time. CPUs that support out-of-order instruction execution can generate multiple concurrent loads, as the CPU does not block on a cache miss, but continues executing (independent) instructions. The number of outstanding memory requests is typically limited inside the CPU. The highest bandwidth in modern hardware is achieved if the access pattern is sequential; in which case the automatic memory prefetcher built into modern CPUs is activated. The difference between *sequential access bandwidth* ($\beta^s = \beta'$) and the respective (nonprefetched) *random access bandwidth* ($\beta^r = Z/\lambda^r$) can be a factor 10, which means that DRAM has truly become a block device, very similar to magnetic disk.

Transition lookaside buffer (**TLB**). A special kind of cache, the TLB is part of the virtual memory support built into modern CPUs: it remembers the latest translations of logical into physical page addresses (e.g., 64). Each memory load/store needs address translation; if the page address is in the TLB (a *TLB hit*), there is no additional cost. If not, a more complex lookup in a mapping table is needed; thus a *TLB miss* implies a penalty. Moreover, the lookup in the (memory-resident) TLB mapping table might generate additional CPU cache misses. Note that with a typical page size of 4KB and 64 entries in the TLB, on many systems TLB delay already comes into play for random access to data structures (e.g., hash-tables) larger than 256KB.

Unified hardware model. Summarizing the above discussion, we describe a computer's memory hardware as a cascading hierarchy of N levels of caches (including TLBs). An index $i \in \{1, \ldots, N\}$ identifies the respective value of a specific level. Exploiting the dualism that an access to level $i + 1$ is caused by a miss on level i allows some simplification of the notation. Introducing the *miss latency* $l_i = \lambda_{i+1}$ and the respective *miss bandwidth* $b_i = \beta_{i+1}$ yields $l_i = Z_i/b_i$. Each cache level is characterized by the parameters given in Table 1.[b] We point out, that these parameters also cover the cost-relevant characteristics of disk accesses. Hence, viewing main memory (e.g., a database system's buffer pool) as cache (level $N + 1$) for I/O operations, it is straightforward to include disk access in this hardware model.

[b] Costs for L1 accesses are assumed to be included in the CPU costs, i.e., λ_1 and β_1 are not used and hence undefined.

Table 1: Cache parameters per level ($i \in \{1, \ldots, N\}$)

Description	Unit	Symbol
Cache name (level)	–	L_i
Cache capacity	Bytes	C_i
Cache-line size	Bytes	Z_i
Number of cache lines	–	$\#_i = C_i/Z_i$
Cache associativity	–	A_i
Sequential ($x = s$) and random ($x = r$) access		
Miss latency	ns	l_i^x
Access latency	ns	$\lambda_{i+1}^x = l_i^x$
Miss bandwidth	Bytes/ns	$b_i^x = Z_i / l_i^x$
Access bandwidth	Bytes/ns	$\beta_{i+1}^x = b_i^x$

We developed a system-independent C program called *Calibrator*[c] to automatically measure these parameters on any computer hardware. The Calibrator uses carefully designed memory access patterns to generate cache misses in a controlled way. Comparing execution times of runs with and without cache misses, it can derive the cache parameters and latencies listed in Table 1. A detailed description of the Calibrator is given in Manegold.[11,12] Sample results for a PC with a 2.4 GHz Intel Core 2 Quad Q6600 CPU look as follows:

```
CPU loop + L1 access:  1.25 ns =  3 cy
Caches:
Level Size Linesize   Asso. Seq-miss-latency  rand-miss-latency
  1  32 KB   64 byte  4-way   0.91ns =  2 cy    4.74ns =  11 cy
  2   4 MB 128 byte  4-way  31.07ns = 75 cy  76.74ns = 184 cy
TLBs:
Level   #entries    pagesize     miss-latency
  1       256         4KB        9.00ns = 22 cy
```

3. MONETDB ARCHITECTURE

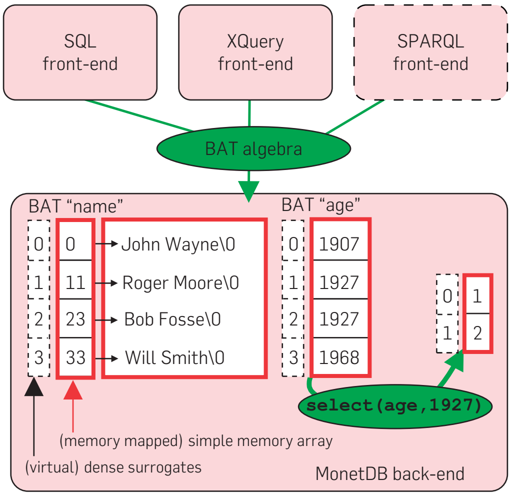
The storage model deployed in MonetDB is a significant deviation of traditional database systems. It uses the decomposed storage model (DSM),[8] which represents relational tables using vertical fragmentation, by storing each column in a separate `<surrogate,value>` table, called binary association table (BAT). The left column, often the surrogate or object-identifier (oid), is called the *head*, and the right column *tail*. MonetDB executes a low-level relational algebra called the *BAT algebra*. Data in execution is always stored in (intermediate) BATs, and even the result of a query is a collection of BATs.

Figure 2 shows the design of MonetDB as a back-end that acts as a BAT algebra virtual machine, with on top a variety of front-end modules that support popular data models and query languages (SQL for relational data, XQuery for XML).

BAT storage takes the form of two simple memory arrays, one for the head and one for the tail column (variable-width types are split into two arrays, one with offsets, and the other with all concatenated data). Internally, MonetDB stores

[c] http://www.cwi.nl/~manegold/Calibrator/calibrator.shtml

Figure 2: MonetDB: a BAT algebra machine.



columns using memory-mapped files. It is optimized for the typical situation that the surrogate column is a densely ascending numerical identifier (0, 1, 2,…); in which case the head array is omitted, and surrogate lookup becomes a fast array index read in the tail. In effect, this use of arrays in virtual memory exploits the fast in-hardware address to disk-block mapping implemented by the memory management unit (MMU) in a CPU to provide an $O(1)$ positional database lookup mechanism. From a CPU overhead point of view this compares favorably to B-tree lookup into slotted pages—the approach traditionally used in database systems for "fast" record lookup.

The Join and Select operators of the relational algebra take an arbitrary Boolean expression to determine the tuples to be joined and selected. The fact that this Boolean expression is specified at query time only, means that the RDBMS must include some *expression interpreter* in the critical runtime code-path of these operators. Traditional database systems implement each relational algebra operator as an iterator class with a *next*() method that returns the next tuple; database queries are translated into a pipeline of such iterators that call each other. As a recursive series of method calls is performed to produce *a single* tuple, computational interpretation overhead is significant. Moreover, the fact that the *next*() method of all iterators in the query plan is executed for each tuple, causes a large *instruction cache footprint*, which can lead to strong performance degradation due to instruction cache misses.[1]

In contrast, each BAT algebra operator has *zero degrees of freedom*: it does not take complex expressions as parameter. Rather, complex expressions are broken into a sequence of BAT algebra operators that perform one simple operation on an entire column of values ("bulk processing"). This allows the implementation of the BAT algebra to forsake an expression interpreting engine; rather all BAT algebra operations in the implementation map onto simple array operations. For instance, the BAT algebra expression

```
R:bat[:oid, :oid]:=select(B:bat[:oid,:int], V:int)
```

can be implemented at the C code level like:

```
for (i = j = 0; i <n; i++)
    if (B.tail[i] == V) R.tail[j++] = i;
```

The BAT algebra operators have the advantage that tight for-loops create high instruction locality which eliminates the instruction cache miss problem. Such simple loops are amenable to compiler optimization (loop pipelining, blocking, strength reduction), and CPU out-of-order speculation.

A potential danger of bulk processing is that it *materializes* intermediate results which in some cases may lead to excessive RAM consumption. Although RAM sizes increase quickly as well, there remain cases that we hit their limit as well. In the MonetDB/X100 project[3] it was shown how partial column-wise execution can be integrated into (nonmaterializing) pipelined query processing.

We can conclude that the MonetDB architecture for realizing database system functionality is radically different from many contemporary product designs, and the reasons for its design are motivated by opportunities for better exploiting modern hardware features.

4. CACHE-CONSCIOUS JOINS

Among the relational algebra operators, the *Join* operator, which finds all matching pairs between all tuples from two relations according to some Boolean predicate, is the most expensive operator—its complexity in the general case is quadratic in input size. However, for equality join predicates, fast (often linear) algorithms are available, such as *Hash-Join*, where the outer relation is scanned sequentially and a hash-table is used to probe the inner relation.

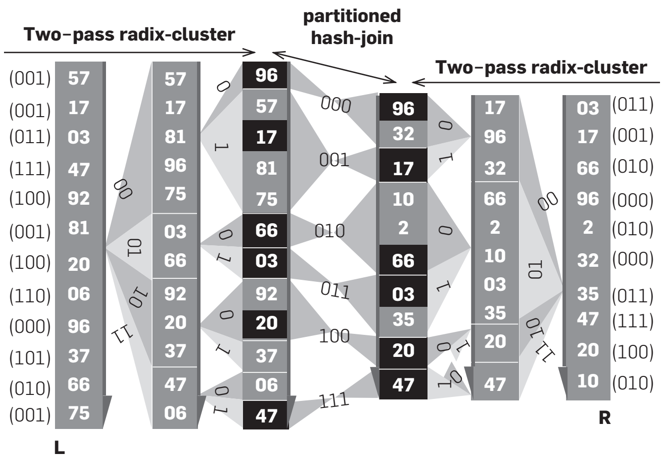
4.1. Partitioned hash-join

The very nature of the hashing algorithm implies that the access pattern to the inner relation (plus hash-table) is random. In case the randomly accessed data is too large for the CPU caches, each tuple access will cause cache misses and performance degrades.

Shatdal et al.[19] showed that a main-memory variant of Grace Hash-Join, in which both relations are first partitioned on hash-number into H separate *clusters*, that each fit into the L2 memory cache, performs better than normal bucket-chained hash-join. However, the clustering operation itself can become a cache problem: their straightforward clustering algorithm that simply scans the relation to be clustered once and inserts each tuple in one of the clusters, creates a random access pattern that writes into H separate locations. If H is too large, there are two factors that degrade performance. First, if H exceeds the number of TLB entries[d] each memory reference will become a *TLB miss*. Second, if H exceeds the number of available cache lines (L1

[d] If the relation is very small and fits the total number of TLB entries times the page size, multiple clusters will fit into the same page and this effect will not occur.

Figure 3: Partitioned hash-join ($H = 8 \Leftrightarrow B = 3$).



Black tuples hit (lowest 3-bits of values in parenthesis)

or L2), *cache thrashing* occurs, causing the number of cache misses to explode.

4.2. Radix-cluster

Our *Radix-Cluster* algorithm[2] divides a relation U into H clusters using multiple passes (see Figure 3). Radix-clustering on the lower B bits of the integer hash-value of a column is achieved in P sequential passes, in which each pass clusters tuples on B_p bits, starting with the leftmost bits ($\sum_1^P B_p = B$). The number of clusters created by the Radix-Cluster is $H = \Pi_1^P H_p$, where each pass subdivides each cluster into $H_p = 2^{B_p}$ new ones. When the algorithm starts, the entire relation is considered one single cluster, and is subdivided into $H_1 = 2^{B_1}$ clusters. The next pass takes these clusters and subdivides each into $H_2 = 2^{B_2}$ new ones, yielding $H_1 * H_2$ clusters in total, etc. With $P = 1$, Radix-Cluster behaves like the straightforward algorithm.

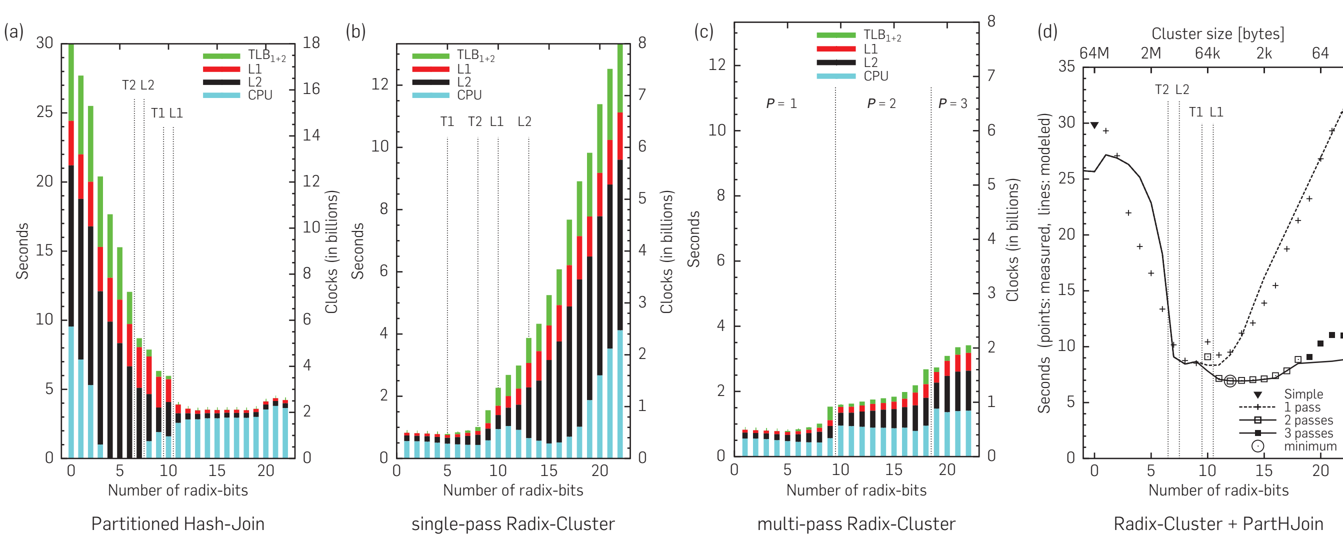
The crucial property of the Radix-Cluster is that the number of randomly accessed regions H_x can be kept low; while still a high overall number of H clusters can be achieved using multiple passes. More specifically, if we keep $H_x = 2^{B_x}$ smaller than the number of cache lines and the number of TLB entries, we completely avoid both TLB and cache thrashing. After Radix-Clustering a column on B bits, all tuples that have the same B lowest bits in its column hash-value, appear consecutively in the relation, typically forming clusters of $|U|/2^B$ tuples (with $|U|$ denoting the cardinality of the entire relation).

Figure 3 sketches a Partitioned Hash-Join of two integer-based relations L and R that uses two-pass Radix-Cluster to create eight clusters—the corresponding clusters are subsequently joined with Hash-Join. The first pass uses the two leftmost of the lower three bits to create four partitions. In the second pass, each of these partitions is subdivided into two partitions using the remaining bit.

For ease of presentation, we did not apply a hash-function in Figure 3. In practice, though, a hash-function should even be used on integer values to ensure that all bits of the join attribute play a role in the lower B bits used for clustering. Note that our surrogate numbers (oids) that stem from a dense integer domain starting at 0 have the property that the lowermost bits are the only relevant bits. Therefore, hashing is not required for such columns, and additionally, a Radix-Cluster on all log(N) relevant bits (where N is the maximum oid from the used domain) equals the well-known *radix-sort* algorithm.

Experiments. Figure 4 show experimental results for a Radix-Cluster powered Partitioned Hash-Join between two memory resident tables of 8 million tuples on an Athlon PC (see Manegold[13]). We used CPU counters to get a breakdown of cost between pure CPU work, TLB, L1, and L2 misses. The vertical axis shows time, while the horizontal axis varies the number of radix-bits B used for clustering (thus it is logarithmic scale with respect to the number of clusters H). Figure 4(a) shows that if a normal Hash-Join is used ($B = 0$), running time is more

Figure 4: Execution time breakdown of individual join phases and overall join performance.



than 30 s due to excessive L1, L2, and TLB misses, but if we join $2^{11} = 2048$ clusters of around 4000 tuples each (i.e., each cluster fits into the Athlon's L1 cache), performance improves around 10-fold. The lines T2, L2, T1, and L1 indicate the clustering degree after which the inner relation (plus hash-table) fits, respectively, the level 2 TLB, L2 data cache, level 1 TLB, and L1 data caches on this Athlon processor. However, Figure 4(b) shows that the straightforward clustering algorithm degrades significantly due to L1 and TLB misses after $B = 8$, as it is filling 256 clusters with only 256 L1 cache lines (on this Athlon), while for similar reasons L2 cache misses become a serious problem after 12 bits. To keep clustering efficient, we should therefore use multipass Radix-Cluster, as shown in Figure 4(c). Since using more clusters improves Partitioned Hash-Join yet degrades Radix-Cluster, the overall results in Figure 4(d) shows a sweet spot at $B = 12$ (two passes).

When a user submits a query to a running database server, its query optimizer determines a physical plan, choosing the right order of the operators as well as choosing the physical algorithm to use. For instance, it may compare SortMerge- with Hash-Join. Additionally, in case of Hash-Join, the optimizer must now also determine how many partitions H, thus, radix-bits B, to use. On the one hand, it needs crucial parameters of the unified hardware model (i.e., the cache configurations) as derived by Calibrator (see Section 2.1); e.g., at DBMS startup. On the other hand, it should model the memory access cost of query processing operators given a value distribution estimate and tuning parameters (such as B). The lines in Figure 4(d) represent the cost prediction of our model for Partitioned Hash-Join, indicating that the techniques described in Section 5 can be quite accurate.

5. MODELING MEMORY ACCESS COSTS

Cache-conscious database algorithms, such as the radix-partitioned hash-join, achieve their optimal performance only if they are carefully tuned to the hardware specifics. Predictive and accurate cost models provide the cornerstones to automate this tuning task. We model the data access behavior in terms of a combination of basic access patterns using the unified hardware model from Section 2.1.

5.1. Memory access cost

Memory access cost can be modeled by estimating the number of cache misses M and scoring them with their respective miss latency l.[13] Akin to detailed I/O cost models we distinguish between random and sequential access. However, we now have multiple cache levels with varying characteristics. Hence, the challenge is to predict the number and kind of cache misses *for all cache levels*. Our approach is to treat all cache levels individually, though equally, and calculate the total cost as the sum of the cost for all levels:

$$T_{\text{Mem}} = \sum_{i=1}^{N} (\mathrm{M}_i^{\mathrm{s}} \cdot l_i^{\mathrm{s}} + \mathrm{M}_i^{\mathrm{r}} \cdot l_i^{\mathrm{r}}).$$

This leaves the challenge to properly estimate the number and kind of cache misses per cache level for various database algorithms. The task is similar to estimating the number and kind of I/O operations in traditional cost models. However, our goal is to provide a generic technique for predicting cache miss rates, sacrificing as little accuracy as possible.

The idea is to abstract data structures as *data regions* and model the complex data access patterns of database algorithms in terms of simple compounds of a few *basic data access patterns*. For these basic patterns, we then provide cost functions to estimate their cache misses. Finally, we present rules to combine basic cost functions and to derive the cost functions of arbitrarily complex patterns.

5.1.1. Basic Access Patterns

Data structures are modeled using a set of *data regions* $\mathbb{D}$. A data region $R \in \mathbb{D}$ consists of $|R|$ *data items* of size $\underline{R}$ (in bytes). We call $|R|$ the *length* of region R, $\underline{R}$ its *width*, and $||R|| = |R| \cdot \underline{R}$ its *size*.

A database table is hence represented by a region R with $|R|$ being the table's cardinality and $\underline{R}$ being the tuple size (width). Similarly, more complex structures like trees are modeled by regions with $|R|$ representing the number of nodes and $\underline{R}$ representing the size (width) of a node.

The following basic access patterns are eminent in the majority of relational algebra implementations.

A **single sequential traversal** `s_trav`(R) sweeps over R, accessing each data item in R exactly once (cf., Figure 5).

A **single random traversal** `r_trav`(R) visits each data item in R exactly once. However, the data items are not accessed in storage order, but chosen randomly (cf., Figure 6).

A **repetitive sequential traversal** `rs_trav`(r, d, R) performs r sequential traversals over R. d = uni (*unidirectional*) indicates that all traversals sweep over R in the same direction. d = bi (*bidirectional*) indicates that subsequent traversals go in alternating directions.

A **repetitive random traversal** `rr_trav`(r, R) performs r random traversals over R. Assuming that the permutation orders of two subsequent traversals are independent, there is no point in discriminating uni- and bidirectional accesses.

Random access `r_acc`(r, R) hits r randomly chosen data items in R after another. The choices are independent of each other. Each data item may be hit more than once. Even

Figure 5: Single sequential traversal: `s_trav(R)`.

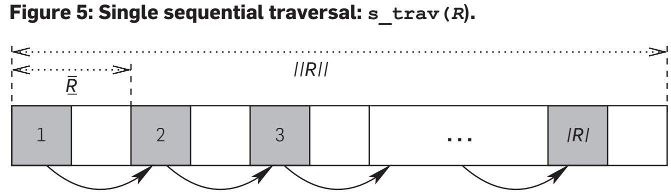
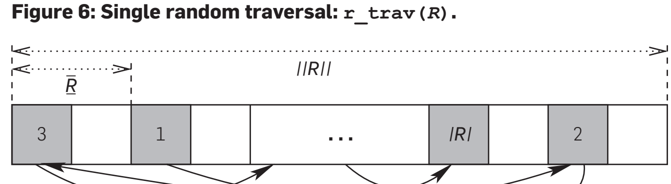


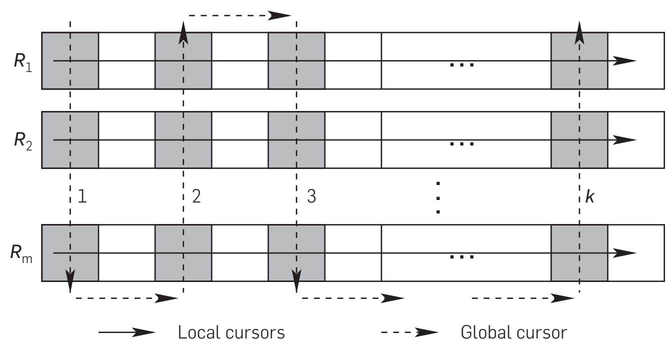
Figure 6: Single random traversal: `r_trav(R)`.



with $r \geq |R|$ we do not require that each data item is hit at least once.

An **interleaved access** nest$(R, m, \mathcal{P}, O[, D])$ models a nested multicursor access pattern where R is divided into m (equal-sized) subregions. Each subregion has its own local cursor. All local cursors perform the same basic pattern $\mathcal{P}$. O specifies, whether the global cursor picks the local cursors randomly (O = ran) or sequentially (O = seq). In the latter case, D specifies, whether all traversals of the global cursor across the local cursors use the same direction (D = uni), or whether subsequent traversals use alternating directions (D = bi). Figure 7 shows an example.

Figure 7: Interleaved multicursor access: nest(R, m, `s_trav(R)`, `seq`, `bi`).



5.1.2. Compound Access Patterns

Database operations access more than one data region, e.g., their input(s) and their output, which leads to *compound* data access patterns. We use $\mathbb{P}_b$, $\mathbb{P}_c$, and $\mathbb{P} = \mathbb{P}_b \cup \mathbb{P}_c$ ($\mathbb{P}_b \cap \mathbb{P}_c = \emptyset$) to denote the set of basic access patterns, compound access patterns, and all access patterns, respectively.

Be $\mathcal{P}_1, \ldots, \mathcal{P}_p \in \mathbb{P}$ $(p > 1)$ data access patterns. There are only two principle ways to combine patterns. They are executed either *sequentially* $(\oplus : \mathbb{P} \times \mathbb{P} \to \mathbb{P})$ or *concurrently* $(\odot : \mathbb{P} \times \mathbb{P} \to \mathbb{P})$. We can apply $\oplus$ and $\odot$ repeatedly to describe more complex patterns.

Table 2 illustrates compound access patterns of some typical database algorithms. For convenience, reoccurring compound access patterns are assigned a new name.

5.2. Cost functions

For each basic pattern, we estimate both sequential and random cache misses for each cache level $i \in \{1, \ldots, N\}$. Given an access pattern $\mathcal{P} \in \mathbb{P}$, we describe the number of misses per cache level as a pair

$$\vec{M}_i(\mathcal{P}) = \left\langle M^s_i(\mathcal{P}), M^r_i(\mathcal{P}) \right\rangle \in \mathbb{N} \times \mathbb{N}$$

containing the number of sequential and random cache misses. The detailed cost functions for all basic patterns introduced above can be found in Manegold.[11,12]

The major challenge with compound patterns is to model cache interference and dependencies among basic patterns.

5.2.1. Sequential Execution

When executed sequentially, patterns do not interfere. Consequently, the resulting total number of cache misses is at most the sum of the cache misses of all patterns. However, if two subsequent patterns operate on the same data region, the second might benefit from the data that the first one leaves in the cache. It depends on the cache size, the data sizes, and the characteristics of the patterns, how many cache misses may be saved this way. To model this effect, we consider the contents or *state* of the caches, described by a set $\mathbf{S}$ of pairs $\langle R, \rho \rangle \in \mathbb{D} \times [0, 1]$, stating for each data region R the fraction ρ that is available in the cache.

In Manegold[11,12] we discuss how to calculate (i) the cache misses of a basic pattern $\mathcal{P}_q \in \mathbb{P}_b$ given a cache state $\mathbf{S}^{q-1}$ as

$$\vec{M}_i\,(\mathrm{S}_i^{q-1}, \mathcal{P}_q) = \mathcal{F}'\,(\mathrm{S}_i^{q-1}, \vec{M}_i(\mathcal{P}_q)),$$

and (ii) the resulting cache state after executing $\mathcal{P}_q$ as

$$\mathrm{S}_i^q\,(\mathrm{S}_i^{q-1}, \mathcal{P}_q) = \mathcal{F}''(\mathrm{S}_i^{q-1}, \mathcal{P}_q).$$

Table 2: Sample data access patterns ($U, V, V', W \in \mathbb{D}$)

Algorithm	Pattern Description	Name
$W \leftarrow select(U)$	$\texttt{s\_trav}(U) \odot \texttt{s\_trav}(W)$	
$W \leftarrow nested\_loop\_join(\mathrm{U}, V)$	$\texttt{s\_trav}(U) \odot \texttt{rs\_trav}(\lvert U\rvert, \texttt{uni}, V) \odot \texttt{s\_trav}(W)$	$=: \texttt{nl\_join}(U, V, W)$
$W \leftarrow zig\_zag\_join(U, V)$	$\texttt{s\_trav}(U) \odot \texttt{rs\_trav}(\lvert U\rvert, \texttt{bi}, V) \odot \texttt{s\_trav}(W)$	
$V' \leftarrow hash\_build(V)$	$\texttt{s\_trav}(V) \odot \texttt{r\_trav}(V')$	$=: \texttt{build\_hash}(V, V')$
$W \leftarrow hash\_probe(U, V')$	$\texttt{s\_trav}(U) \odot \texttt{r\_acc}(\lvert U\rvert, V') \odot \texttt{s\_trav}(W)$	$=: \texttt{probe\_hash}(U, V', W)$
$W \leftarrow hash\_join(U, V)$	$\texttt{build\_hash}(V, V') \oplus \texttt{probe\_hash}(U, V', W)$	$=: \texttt{h\_joins}(U, V, W)$
$\{U_j\}\vert_{j=1}^{m} \leftarrow cluster(U, \mathrm{m})$	$\texttt{s\_trav}(U) \odot \texttt{nest}(\{U_j\}\vert_{j=1}^{m}, m, \texttt{s\_trav}(U_j), \texttt{ran})$	$=: \texttt{part}(U, m, \{U_j\}\vert_{j=1}^{m})$
$W \leftarrow part\_nl\_join(U, V, \mathrm{m})$	$\texttt{part}(U, m, \{U_j\}\vert_{j=1}^{m}) \oplus \texttt{part}(V, m, \{V_j\}\vert_{j=1}^{m}) \oplus \texttt{nl\_join}(U_1, V_1, W_1) \oplus \ldots \oplus \texttt{nl\_join}(U_m, V_m, W_m)$	
$W \leftarrow part\_h\_join(U, V, \mathrm{m})$	$\texttt{part}(U, m, \{U_j\}\vert_{j=1}^{m}) \oplus \texttt{part}(V, m, \{V_j\}\vert_{j=1}^{m}) \oplus \texttt{nl\_join}(U_1, V_1, W_1) \oplus \ldots \oplus \texttt{h\_join}(U_m, V_m, W_m)$	

With these, we can calculate the number of cache misses that occur when executing patterns $\mathcal{P}_1, \ldots, \mathcal{P}_p \in \mathbb{P}$, $p > 1$ sequentially, given an initial cache state $\mathbf{S}^0$, as

$$\vec{M}_i\,(S_i^0, \oplus(\mathcal{P}_1, \ldots, \mathcal{P}_p)) = \sum_{q=1}^{p} \vec{M}_i\,(S_i^{q-1}, \mathcal{P}_q).$$

5.2.2. Concurrent Execution

When executing patterns concurrently, we actually have to consider the fact that they are competing for the same cache. We model the impact of the cache interference between concurrent patterns by dividing the cache among all patterns. Each pattern $\mathcal{P}$ gets a fraction $0 < \nu < 1$ of the cache according to its *footprint size* $\mathbf{F}$, i.e., the number of cache lines that it potentially revisits. The detailed formulas for $\mathbf{F}_i(\mathcal{P})$ with $\mathcal{P} \in \mathbb{P}$ are given in Manegold.[11,12]

We use $\vec{\mathbf{M}}_{i/\nu}$ to denote the number of misses with only a fraction $0 < \nu < 1$ of the total cache size available.

With these tools at hand, we calculate the cache misses for concurrent execution of patterns $\mathcal{P}_1, \ldots, \mathcal{P}_p \in \mathbb{P}$ $(p > 1)$ given an initial cache state $\mathbf{S}^0$ as

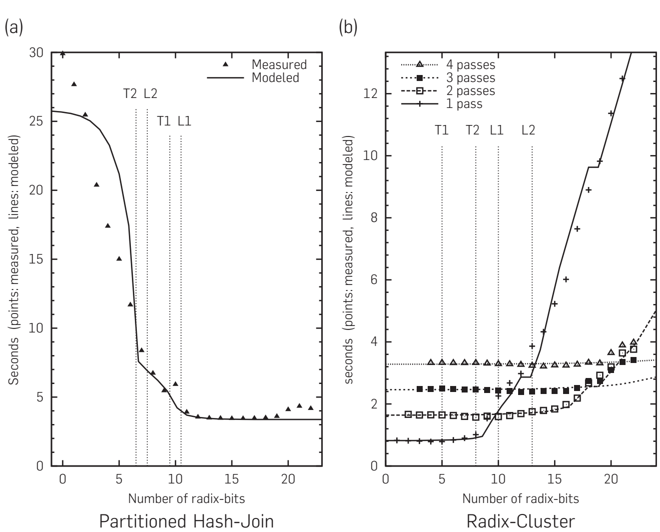
$$\vec{M}_i\,(S_i^0, \odot(\mathcal{P}_1, \ldots, \mathcal{P}_p)) = \sum_{q=1}^{p} \vec{M}_{i/\nu_q}\,(S_i^0, \mathcal{P}_q).$$

For our radix-partitioned hash-join algorithm, Figures 4d and 8 compare the cost predicted by our cost model to the measured execution times on an Athlon PC. An exhaustive experimental validation of our models is presented in Manegold.[11,12]

5.2.3. Query Execution Plans

With the techniques discussed, we have the basic tools at hand to estimate the number and kind of cache misses of complete query plans, and hence can predict their memory access costs. The various operators in a query plan are combined in the same way the basic patterns are combined to form compound patterns. Basically, the query plan describes, which operators are executed one after the other and which are executed concurrently. We view pipelining as concurrent execution of data-dependent operators. Hence, we can derive the complex memory access pattern of a query plan by combining the compound patterns of the operators as discussed above. Considering the caches' states as introduced before takes care of handling data dependencies.

Figure 8: Sample cost model validation.



6. RELATED WORK

The growing mismatch between the way database systems are engineered versus hardware evolution was first brought to light in a number of workload studies. An early study[15] already showed database workloads, compared with scientific computation, to exhibit significantly more instruction cache misses (due to a large code footprint) and more (L2) data cache misses.

Instruction cache misses are specifically prevalent in transaction processing workloads. The STEPS[10] approach therefore organizes multiple concurrent queries into execution teams, and evaluates each query processing operator for all members of the team one after another, while its code is hot. Another proposal in this direction, aimed at analysis queries, proposed to split query plans into sections whose code fits the instruction cache, putting a so-called "Buffer" operator on the section boundary.[23] The Buffer operator repeatedly invoke the query subsection below it, buffering the resulting tuples without passing them on yet, such that the operators in the subsection are executed multiple times when hot, amortizing instruction cache misses. The high locality of the BAT algebra operators in MonetDB and materialization of results can be seen as an extreme form of this latter strategy.

In the area of index structures, Cache-sensitive B+ Trees (CSB+-Trees)[17] ensure that internal nodes match the cache-line size, optimizing the number of cache-line references, and introduce highly optimized in-node search routines for faster lookup.

The MonetDB work[2,12,13] showed vertical data fragmentation (DSM[8]) to benefit analysis queries, due to reduced memory traffic and an increased spatial locality. Column-stores have since received much attention for use in data warehousing environments (e.g., C-Store,[20] and the CWI follow-up system MonetDB/X100[3]), introducing column-store specific compression and query evaluation techniques.

Considering hash-join, cache-sized partitioning was first proposed in Shatdal[19] and subsequently improved in Boncz,[2] as summarized in Section 4. The Radix-Cluster algorithm was later supplemented with an inverse Radix-Decluster algorithm,[14] that allows to perform arbitrary data permutations in a cache-efficient manner (this can be used for sorting, as well as for postponing the propagation of join columns to after the join phase).

An alternative hash-join approach uses software prefetching, exploiting the explicit memory-to-cache prefetching instructions offered by modern CPUs. Group prefetching was shown in Chen[6] to perform better than cache-partitioning and was also shown to be more resistant to interference by other programs. Prefetching was also successfully applied in B-tree access[7] to increase the width of the nodes without paying the latency cost of fetching the additional cache lines. Memory prefetching has also been applied to optimize various data accesses in the Inspector Join algorithm.[5] A general disadvantage of hardware prefetching is that it is notoriously platform-dependent and difficult to tune, therefore hindering its application in generic software packages. A precondition for such tuning is the availability of a unified hardware model that provides parameters, and memory cost formulas, as introduced in Manegold[11,12] and summarized in Section 5.

Architecture-conscious results continue to appear regularly in major database research publications, and also have a specialized workshop (DaMoN) colocated with SIGMOD. Other topics that have been addressed include minimizing branch mispredictions in selection operations,[18] using SIMD instructions for database tasks,[22] and query processing with GPU hardware,[9] which led in 2006 to a NVIDIA graphics card to become the PennySort sorting benchmark champion. Recently there is interest in the use of Flash memory for database storage as well as query parallelization for multicore CPUs.

7. CONCLUSION

When MonetDB debuted more than a decade ago, the idea of using vertical storage was radical, however in the past few years the database community has confirmed its benefits over horizontal-only storage,[20] and the principle is currently being adopted broadly in commercial systems.

Less repeated, as of yet, have been the MonetDB results that focus on highly CPU-efficient execution. The reshaping of relational algebra to map it into tightly looped array processing, leads to as yet unmatched raw computational efficiency, benefiting from trends in CPU design and compiler optimizer support.

In the broad sense, the research around MonetDB aims at redefining database architecture in the face of an ever-changing computer architecture and database application landscape. This research still continues, for instance by making database systems *self-tuning* using automatic on-the-fly indexing strategies that piggyback on query execution ("database cracking"), and by improving query optimizer efficiency and robustness using a modular *runtime* framework that transforms query optimization from a static procedure that precedes query execution, into a dynamic mechanism where query optimization and execution continuously interleave.

References

1. Ailamaki, A.G., DeWitt, D.J., Hill, M.D., and Wood, D.A. DBMSs on a modern processor: Where does time go? In *International Conference on Very Large Data Bases (VLDB)*, Sept. 1999, 266–277.
2. Boncz, P.A., Manegold, S., and Kersten, M.L. Database architecture optimized for the new bottleneck: Memory access. In *International Conference on Very Large Data Bases (VLDB)*, Sept. 1999, 54–65.
3. Boncz, P.A., Zukowski, M., and Nes, N. MonetDB/X100: Hyper-pipelining query execution. In *International Conference on Innovative Data Systems Research (CIDR)*, Jan. 2005, 225–237.
4. Chaudhuri, S. and Weikum, G. Rethinking database system architecture: Towards a self-tuning RISC-style database system. In *International Conference on Very Large Data Bases (VLDB)*, Sept. 2000, 1–10.
5. Chen, S., Ailamaki, A., Gibbons, P.B., and Mowry, T.C. Inspector joins. In *International Conference on Very Large Data Bases (VLDB)*, Aug. 2005.
6. Chen, S., Ailamaki, A., Gibbons, P.B., and Mowry, T.C. Improving hash join performance through prefetching. *ACM Trans. Database Syst.*, *32*, 3 (2007).
7. Chen, S., Gibbons, P.B., and Mowry, T.C. Improving index performance through prefetching. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, June 2001.
8. Copeland, G.P. and Khoshafian, S. A decomposition storage model. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, May 1985, 268–279.
9. Govindaraju, N., Gray, J., Kumar, R., and Manocha, D. GPUTeraSort: High performance graphics co-processor sorting for large database management. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, June 2006.
10. Harizopoulos, S. and Ailamaki, A. STEPS towards cache-resident transaction processing. In *International Conference on Very Large Data Bases (VLDB)*, Aug. 2004.
11. Manegold, S. Understanding, modeling, and improving main-memory database performance. PhD thesis, Universiteit van Amsterdam, Amsterdam, the Netherlands, Dec. 2002.
12. Manegold, S., Boncz, P.A., and Kersten, M.L. Generic database cost models for hierarchical memory systems. In *International Conference on Very Large Data Bases (VLDB)*, Aug. 2002, 191–202.
13. Manegold, S., Boncz, P.A., and Kersten, M.L. Optimizing main-memory join on modern hardware. *IEEE Trans. Knowl. Data Eng.*, *14*, 4 (July 2002), 709–730.
14. Manegold, S., Boncz, P.A., Nes, N., and Kersten, M.L. Cache-conscious radix-decluster projections. In *International Conference on Very Large Data Bases (VLDB)*, Aug. 2004, 684–695.
15. Maynard, A.M.G., Donnelly, C.M., and Olszewski, B.R. Contrasting characteristics and cache performance of technical and multi-user commercial workloads. *SIGOPS Oper. Syst. Rev.*, *28*, 5 (Aug. 1994), 145–156.
16. Patterson, D. Latency lags bandwidth. *Commun. ACM 47*, 10 (Oct. 2004), 71–75.
17. Rao, J. and Ross, K.A. Making B+ -Trees cache conscious in main memory. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, June 2000.
18. Ross, K.A. Conjunctive selection conditions in main memory. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, June 2002.
19. Shatdal, A., Kant, C., and Naughton, J. Cache conscious algorithms for relational query processing. In *International Conference on Very Large Data Bases (VLDB)*, Sept. 1994, 510–512.
20. Stonebraker, M., Abadi, D.J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., Lau, E., Lin, A., Madden, S.R., O'Neil, E.J., O'Neil, P.E., Rasin, A., Tran, N., and Zdonik, S.B. C-Store: A column-oriented DBMS. In *International Conference on Very Large Data Bases (VLDB)*, Sept. 2005, 553–564.
21. Stonebraker, M., Madden, S.R., Abadi, D.J., Harizopoulos, S., Hachem, N., and Helland, P. The end of an architectural era (it's time for a complete rewrite). In *International Conference on Very Large Data Bases (VLDB)*, Sept. 2007, 1150–1160.
22. Zhou, J. and Ross, K.A. Implementing database operations using SIMD instructions. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, June 2002.
23. Zhou, J. and Ross, K.A. Buffering database operations for enhanced instruction cache performance. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, June 2004.

Peter A. Boncz (Peter.Boncz@cwi.nl) CWI, Kruislaan, Amsterdam, the Netherlands.

Martin L. Kersten (Martin.Kersten@cwi.nl) CWI, Kruislaan, Amsterdam, the Netherlands.

Stefan Manegold (Stefan.Manegold@cwi.nl) CWI, Kruislaan, Amsterdam, the Netherlands.

DOI:10.1145/1409360.1409381

Technical Perspective
Patching Program Errors

By Martin C. Rinard

C PROGRAMMERS ARE all too familiar with out-of-bounds memory errors. A typical scenario involves a pointer calculation that produces an address outside the target block of memory that the developer intends the program to write. The resulting out-of-bounds writes corrupt the data structures of otherwise unrelated parts of the program, causing the program to fail or mysteriously generate unexpected outputs. Analogous errors in Java and other type-safe languages cause out-of-bounds exceptions, which typically terminate the execution of the program.

The paper here presents an intriguing technique for automatically isolating and correcting these errors. The basic idea is simple. Allocate blocks randomly within a memory much larger than required to execute the program. Set the remaining unused memory to contain specific "canary" values. Run multiple versions of the program to failure, then observe overwritten canary values to compute 1) the target block the out-of-bounds writes were intended to write, and 2) how much additional memory would have been required to bring the writes back within bounds. The fix is to go back to the allocation site in the program that allocated the target block, then apply a patch that suitably increases the size of blocks allocated at that site. The paper also presents a conceptually similar technique for dealing with accesses to prematurely deallocated blocks of memory. Note that the combination of random allocation and multiple heaps is the key insight that makes it possible to isolate the target block. Randomizing the block placement provides the block location diversity required to ensure that, with high probability, only the target block will have the same offset from the overwritten canary values in all of the heaps.

One interesting aspect of this approach is that it generates patches almost immediately and without the need for human interaction. It is therefore suitable for uses (such as eliminating security vulnerabilities) that place a premium on obtaining fast responses to newly exposed errors. It can also eliminate the need to interact with a (potentially distracted, indifferent, recalcitrant, or defunct) software development organization to obtain relief from an error, even when the software is distributed only in binary form.

An even more interesting aspect of this approach is that it may very well produce patches that do not completely fix the problem—the fact that the addition of a given amount of memory would have eliminated the out-of-bounds accesses in one observed execution provides no guarantee that it will eliminate such accesses in other executions. Benefits of this approach include simplicity and feasibility of implementation and deployment.

Over the last several years I have had many conversations about this and other (more aggressive or even unsound) techniques for automatically correcting or tolerating errors. Many developers and researchers find something deeply unsettling about a program that continues to execute in the face of errors. In fact, the most common response is that programs should stop when they encounter an error and not continue until a developer fixes the error.

I believe I get this response for two reasons. First, most developers feel responsible for the behavior of the programs they produce. Automatic intervention can often invalidate, undermine, or simply bypass the reasoning the developer originally used when developing the program. It is not clear what is going to replace that reasoning or who is then responsible for the actions of the modified program. Second, the needs of the developer are usually best served with a fail-stop approach. Stopping the execution as close as possible to the error makes it easier for developers to isolate and fix errors while the program is under development. Many developers instinctively reject any approach that involves continued execution after an error, presumably because the continued execution can complicate debugging by obscuring the location of the error.

But the needs of users are very different from the needs of developers. Stopping the program at the first sign of an error can unacceptably deny access to important functionality. The undesirability of this denial of service can be seen (in embryonic form) in the common practice of removing assertions before releasing a system for production use.

Even if you believe that programs that continue to execute through errors are more likely to produce unacceptable results than programs that execute without a detected error, there are clearly many scenarios in which continued execution is superior. Consider, for example, a program that produces a result (such as a drawing or digitally edited picture) whose acceptability is obvious upon inspection. Continued execution in the face of errors may very well enable the program to produce an acceptable result that satisfies the user's needs.

A fail-stop approach can also be dangerous when applied to programs that control unstable physical phenomena. In this case, continued execution through errors can offer the only real hope of obtaining an acceptable outcome. To cite a real-world example, consider the infamous Ariane 5 disaster. This disaster was directly caused by the inappropriate use of a fail-stop approach in a safety-critical embedded software system.

So what does the future hold for automatic error recovery and repair? At this point we have a variety of viable strategies for dealing with errors (such as out-of-bounds memory accesses or null-pointer dereferences) that no program should ever commit. Future advances will focus on correcting application-specific errors. The key is to obtain specifications that provide a foundation for the recognition and elimination of unacceptable behavior. One particularly fruitful area is sure to be unsound techniques that (in return for simplicity and feasibility of implementation and deployment) ignore traditional requirements that program transformations should never perturb error-free executions. The success of such techniques could then pave the way for a more mature software engineering perspective that views correctness as simply one of a set of engineering trade-offs to be appropriately managed during the lifetime of the system.

Martin C. Rinard (rinard@cag.csail.mit.edu) is a professor in the Department of Electrical Engineering and Computer Science at MIT, Cambridge, MA.

DOI:10.1145/1409360.1409382

Exterminator: Automatically Correcting Memory Errors with High Probability

By Gene Novark, Emery D. Berger, and Benjamin G. Zorn

Abstract

Programs written in C and C++ are susceptible to memory errors, including buffer overflows and dangling pointers. These errors, which can lead to crashes, erroneous execution, and security vulnerabilities, are notoriously costly to repair. Tracking down their location in the source code is difficult, even when the full memory state of the program is available. Once the errors are finally found, fixing them remains challenging: even for critical security-sensitive bugs, the average time between initial reports and the issuance of a patch is nearly 1 month.

We present Exterminator, a system that automatically corrects heap-based memory errors without programmer intervention. Exterminator exploits randomization to pinpoint errors with high precision. From this information, Exterminator derives *runtime patches* that fix these errors both in current and subsequent executions. In addition, Exterminator enables collaborative bug correction by merging patches generated by multiple users. We present analytical and empirical results that demonstrate Exterminator's effectiveness at detecting and correcting both injected and real faults.

1. INTRODUCTION

The use of manual memory management and unchecked memory accesses in C and C++ leaves applications written in these languages susceptible to a range of memory errors. These include buffer overruns, where reads or writes go beyond allocated regions, and dangling pointers, when a program deallocates memory while it is still live. Memory errors can cause programs to crash or produce incorrect results. Worse, attackers are frequently able to exploit these memory errors to gain unauthorized access to systems.

Debugging memory errors is notoriously difficult. Reproducing the error requires an input that exposes it. Since inputs are often unavailable from deployed programs, developers must either concoct such an input or find the problem via code inspection. Once a test input is available, software developers typically execute the application with heap debugging tools like Purify[7] and Valgrind,[10] which may slow execution by an order of magnitude. When the bug is ultimately discovered, developers must construct and carefully test a patch to ensure that it fixes the bug without introducing any new ones. This process can be costly and time-consuming. For example, according to Symantec, the average time between the discovery of a critical, *remotely exploitable* memory error and the release of a patch for enterprise applications is 28 days.[17]

Because memory errors are so difficult to find and fix, researchers have proposed many solutions that fall roughly into two categories: detection, which prevents errors from being exploited and potentially allows them to be debugged more easily; and toleration, where the effects of errors are mitigated. *Fail-stop* systems are compiler-based approaches that may require access to source code, and abort programs when they perform illegal operations like buffer overflows.[1,2,6,9]

Fault-tolerant runtime systems, which attempt to hide the effect of errors, have also been proposed. Rinard's *failure-oblivious* systems are also compiler-based, but manufacture read values and drop or cache illegal writes for later reuse.[13,14] The Rx system[12] uses logging and replay, with potential perturbation, to provide fault tolerance. Our previous work, DieHard,[3,4] uses heap over-provisioning, layout randomization, and optional voting-based replication to reduce the likelihood that an error will have any effect (see Section 3.1 for an overview). DieHard provides *probabilistic memory safety*, giving the application the illusion of having an infinite heap with a well-defined probability.

Contributions: This paper presents **Exterminator**, a runtime system that not only tolerates but also detects, isolates, and corrects two classes of heap-based memory errors with high probability. Exterminator requires neither source code nor programmer intervention, and fixes existing errors without introducing new ones. To our knowledge, this system is the first of its kind.

Exterminator relies on an efficient probabilistic debugging allocator that we call **DieFast**. DieFast is based on DieHard's allocator, which ensures that heaps are independently randomized. However, while DieHard can only probabilistically tolerate errors, DieFast probabilistically detects them.

When Exterminator discovers an error, it dumps a **heap image** that contains the complete state of the heap. Exterminator's **probabilistic error isolation** algorithm processes one or more heap images to try to locate the source of the error. This error isolation algorithm has provably low false-positive and false-negative rates. Buffer overflows and dangling pointer errors can be distinguished because they tend to produce distinct patterns of heap corruption.

A previous version of this article appeared in *Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation.* ACM, NY, pp. 1–11.

Once Exterminator locates a buffer overflow, it determines the allocation site of the overflowed object, and the size of the overflow. For dangling pointer errors, Exterminator determines both the allocation and deletion sites of the dangled object, and computes how prematurely the object was freed.

With this information in hand, Exterminator corrects the errors by generating **runtime patches**. These patches operate in the context of a **correcting allocator**. The correcting allocator prevents overflows by padding objects, and prevents dangling pointer errors by deferring object deallocations. These actions impose little space overhead because Exterminator's runtime patches are tailored to the specific allocation and deallocation sites of each error.

After Exterminator completes patch generation, it both stores the patches to correct the bug in subsequent executions, and triggers a patch update in the running program to fix the bug in the current execution. Exterminator's patches also compose straightforwardly, enabling **collaborative bug correction**: users running Exterminator can automatically merge their patches, thus systematically and continuously improving application reliability.

Exterminator can operate in three distinct modes: an **iterative mode** for runs over the same input, a **replicated mode** that can correct errors on the fly, and a **cumulative mode** that corrects errors across multiple runs of the same application.

We experimentally demonstrate that, in exchange for modest runtime overhead (around 25%), Exterminator effectively isolates and corrects both injected and real memory errors, including buffer overflows in the Squid Web cache server and the Mozilla Web browser.

2. MEMORY ERRORS

Incorrect programs exhibit a variety of errors related to heap objects, including *dangling pointers*, where a heap object is freed while it is still live; *invalid frees*, where a program deallocates an object that was never returned by the allocator; *double frees*, where a heap object is deallocated multiple times without an intervening allocation; *uninitialized reads*, where the program, despite using all pointers correctly, reads memory that has never been initialized; and *out-of-bound writes*, where the memory address to be written is computed by using a valid pointer to an object but an incorrect offset or index, so that the address computed lies outside the object. We use the term *buffer overflow* to refer to an out-of-bound write whose offset from a base pointer is positive and too large. (Out-of-bound writes where the offset is negative appear to be rather less common in practice.)

Errors such as double frees and invalid frees, if not properly handled, can result in inconsistent allocator metadata and are a potential security vulnerability. These errors can lead to heap corruption or abrupt program termination. Out-of-bound writes and dangling pointers may result in corruption of either allocator metadata or application objects. Uninitialized reads, because the values read are not specified by the language semantics, can affect application execution in arbitrary ways. Because good allocator design can mitigate the effect of double frees and invalid frees, buffer overruns and dangling pointer errors are currently the most commonly exploited heap errors, and hence the most important to address.

While DieHard probabilistically tolerates dangling pointers and buffer overflows of heap objects, Exterminator both detects and permanently corrects them. Exterminator's allocator (DieFast) shares DieHard's immunity from double frees and invalid frees. Exterminator does not currently address uninitialized reads, reads outside the bounds of an object, or out-of-bound writes with negative offsets.

3. SOFTWARE ARCHITECTURE

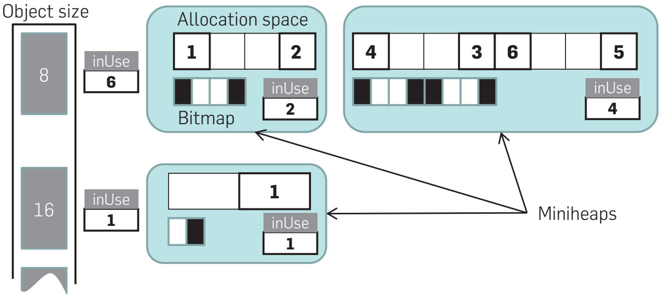
Exterminator's software architecture extends and modifies DieHard to enable its error isolating and correcting properties. This section first describes DieHard, and then shows how Exterminator augments its heap layout to track information needed to identify and remedy memory errors. Second, it presents DieFast, a probabilistic debugging allocation algorithm that exposes errors to Exterminator. Finally, it describes Exterminator's three modes of operation.

3.1. DieHard overview

The DieHard system includes a bitmap-based, fully randomized memory allocator that provides *probabilistic memory safety*.[3] The latest version of DieHard, upon which Exterminator is based, adaptively sizes its heap to be M times larger than the maximum needed by the application[4] (see Figure 1). This version of DieHard allocates memory from increasingly large chunks that we call *miniheaps*. Each miniheap contains objects of exactly one size. DieHard allocates new miniheaps to ensure that, for each size, the ratio of allocated objects to total objects is never more than $1/M$. Each new miniheap is twice as large, and thus holds twice as many objects, as the previous largest miniheap.

Allocation randomly probes a miniheap's bitmap for the given size class for a 0 bit, indicating a free object available for reclamation, and sets it to 1. This operation takes $O(1)$ expected time. Freeing a valid object resets the appropriate bit, which is also a constant-time operation. DieHard's use of randomization across an over-provisioned heap makes it probabilistically likely that buffer overflows will land on free space, and unlikely that a recently freed object will be reused soon.

Figure 1: The adaptive (new) DieHard heap layout, used by Exterminator. Objects in the same size class are allocated randomly from separate *miniheaps*, which combined hold *M* times more memory than required (here, *M* = 2).



DieHard optionally uses replication to increase the probability of successful execution. In this mode, it broadcasts inputs to a number of replicas of the application process, each initialized with a different random seed. A voter intercepts and compares outputs across the replicas, and only actually generates output agreed on by a plurality of the replicas. The independent randomization of each replica's heap makes the probabilities of memory errors independent. Replication thus exponentially decreases the likelihood of a memory error affecting output, since the probability of an error corrupting a majority of the replicas is low.

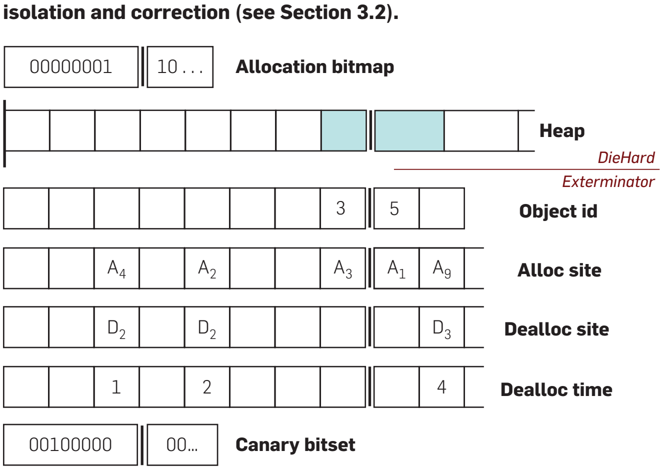
3.2. Exterminator's heap layout

Figure 2 presents Exterminator's heap layout, which includes five fields per object for error isolation and correction: an **object id**, **allocation** and **deallocation sites**, **deallocation time**, which records when the object was freed, and a **canary bit**.

An object id of n means that the object is the nth object allocated. Exterminator uses object ids to identify objects across heaps in multiple program executions. These ids are needed because object addresses cannot be used to identify them across differently randomized heaps. The site information fields capture the calling context for allocations and deallocations: a 32-bit hash of the least significant bytes of the five most-recent return addresses. The canary bit indicates if the object was filled with canaries (see Section 3.3). All of these metadata are initialized when an object is allocated and persist after the object is freed until a new object is allocated in its place.

The space overhead of this out-of-band metadata plus the allocation bit is 16 bytes plus 2 bits of space overhead per object. This amount is comparable to that of typical freelist-based memory managers like the Lea allocator, which prepend an 8- or 16-byte header (on 32- or 64-bit systems) to each object.[8]

Figure 2: An abstract view of Exterminator's heap layout. Metadata below the horizontal line contains information used for error isolation and correction (see Section 3.2).



3.3. A probabilistic debugging allocator

Exterminator uses a new, probabilistic debugging allocator that we call DieFast. DieFast uses the same randomized heap layout as DieHard, but extends its allocation and deallocation algorithms to detect and expose errors. Unlike previous debugging allocators, DieFast has a number of unusual characteristics tailored for its use in the context of Exterminator.

3.3.1. Implicit Fence-Posts

Many existing debugging allocators pad allocated objects with fence-posts (filled with **canary** values) on both sides. They can thus detect out-of-bound writes that are just beyond the start or end of an object by checking the integrity of these fence-posts. This approach has the disadvantage of increasing space requirements. Combined with the already-increased space requirements of a DieHard-based heap, the additional space overhead of padding may be unacceptably large.

DieFast exploits two facts to obtain the effect of fence-posts without any additional space overhead. First, because its heap layout is headerless, one fence-post serves double duty: a fence-post following an object acts as the one preceding the next object. Second, because allocated objects are separated by (on average) $M - 1$ freed objects on the heap, we use freed space to act as fence-posts.

3.3.2. Random Canaries

Traditional debugging canaries include values, such as the hexadecimal value `0xDEADBEEF`, that are readily distinguished from normal program data in a debugging session. However, one drawback of a deterministically chosen canary is that it is always possible for the program to use the canary pattern as a data value. Because DieFast uses canaries located in freed space rather than in allocated space, a fixed canary would lead to a high false-positive rate if that data value were common in allocated objects.

DieFast instead uses a random 32-bit value set at startup. Since both the canary value and heap addresses are random and differ on every execution, any fixed data value (likewise, any given pointer) has a low probability of colliding with the canary; this ensures a low false-positive rate (see Theorem 2). To increase the likelihood of detecting an error, DieFast always sets the last bit of the canary value to 1. Setting this bit will cause an alignment error if the canary is dereferenced, but still keeps the probability of an accidental collision with the canary low ($1/2^{31}$).

3.3.3. Probabilistic Fence-Posts

Intuitively, the most effective way to expose a dangling pointer error is to fill all freed memory with canary values. For example, dereferencing a canary value as a pointer will likely trigger a segmentation violation or alignment error.

Unfortunately, reading random values does not necessarily cause programs to fail. For example, in the `espresso` benchmark, some objects hold bitsets. Filling a freed bitset with a random value does not cause the program to terminate but may affect the correctness of the computation.

When reading from a canary-filled dangled object causes a program to run awry, it can become difficult to isolate the error. In the worst case, half of the heap could be filled with freed objects, all overwritten with canary values. All of these objects would then be potential sources of dangling pointer errors.

In cumulative mode, Exterminator prevents this scenario by making a random choice every time an object is freed; rather than always filling the freed object with canaries and setting the associated canary bit, it performs this filling and bit-setting action with probability p. This probabilistic approach may seem to degrade Exterminator's ability to find errors. However, it is required to isolate read-only dangling pointer errors, where the canary itself remains intact. Because it would take an impractically large number of iterations or replicas to isolate these errors, Exterminator always fills freed objects with canaries when not running in cumulative mode (see Sections 5.2 and 7.2 for discussion).

3.3.4. Probabilistic Error Detection

Whenever DieFast allocates memory, it examines the memory to be returned to verify that any canaries it is supposed to contain (as indicated by the canary bitset) are intact. If not, in addition to signaling an error (see Section 3.4), DieFast sets the allocated bit for this chunk of memory. This "bad object isolation" ensures that the object will not be reused for future allocations, preserving its contents for Exterminator's subsequent use. By checking canary integrity on each allocation, DieHard can be expected to detect heap corruption within approximately H allocations, where H is the number of objects on the heap.

After every deallocation, DieFast checks both the preceding and subsequent objects. For each of these, DieFast checks if they are free. If so, it performs the same canary check as above. Recall that because DieFast's allocation is random, the identity of these adjacent objects will differ from run to run. Checking both the subsequent and the preceding objects on each free allows DieFast to perform an inexpensive check for any nearby out-of-bound writes, including "strided" object writes (e.g., `a[i+32]`) that might jump over a subsequent object.

3.4. Modes of operation

Exterminator can be used in three modes of operation: an iterative mode suitable for testing or whenever all program inputs can be made available for repeated execution, a replicated mode that is suitable both for testing and for restricted deployment scenarios, and a cumulative mode that is suitable for broad deployment. All of these rely on the generation of heap images, which Exterminator examines to isolate errors and compute runtime patches.

If Exterminator discovers an error when executing a program, or if DieFast signals an error, Exterminator forces the process to emit a heap image file. This file is akin to a core dump, but contains less data (e.g., no code) and is organized to simplify processing. In addition to the full heap contents and heap metadata, the heap image includes the current allocation time (i.e., the number of allocations to date).

3.4.1. Iterative Mode

Exterminator's iterative mode operates without replication. To find a single bug, Exterminator is initially invoked via a command-line option that directs it to stop as soon as it detects an error. Exterminator then re-executes the program in "replay" mode over the same input (but with a new random seed). In this mode, Exterminator reads the allocation time from the initial heap image to abort execution at that point; we call this a **malloc breakpoint**. Exterminator then begins execution and ignores DieFast error signals that are raised before the malloc breakpoint is reached.

Once it reaches the malloc breakpoint, Exterminator triggers another heap image dump. This process can be repeated multiple times to generate independent heap images. Exterminator then performs postmortem error isolation and runtime patch generation. A small number of iterations usually suffices for Exterminator to generate runtime patches for an individual error, as we show in Section 7.2. When run with a correcting memory allocator that incorporates these changes (described in detail in Section 6.3), these patches automatically fix the isolated errors.

3.4.2. Replicated Mode

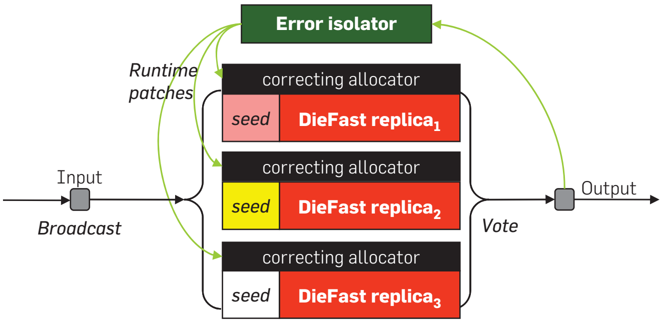
The iterated mode described above works well when all inputs are available so that rerunning an execution is feasible. However, when applications are deployed in the field, such inputs may not be available, and replaying may be impractical. The replicated mode of operation allows Exterminator to correct errors while the program is running, without the need for multiple iterations.

As Figure 3 shows, Exterminator (like DieHard) can run a number of differently randomized replicas simultaneously (as separate processes), broadcasting inputs to all and voting on their outputs. However, Exterminator uses DieFast-based heaps, each with a correcting allocator. This organization lets Exterminator discover and fix errors.

In replicated mode, when DieFast signals an error or the voter detects divergent output, Exterminator sends a signal that triggers a heap image dump for each replica. Exterminator also dumps heap images if any replica crashes because of a segmentation fault.

If DieFast signals an error, the replicas that dump a heap image do not have to stop executing. If their output

Figure 3: Exterminator's replicated architecture (Section 3.4). Replicas are equipped with different seeds that fully randomize their DieFast-based heaps (Section 3.3), input is broadcast to all replicas, and output goes to a voter. A crash, output divergence, or signal from DieFast triggers the error isolator (Section 4), which generates *runtime patches*. These patches are fed to correcting allocators (Section 6), which fix the bug for current and subsequent executions.



continues to be in agreement, they can continue executing concurrently with the error isolation process. Once the runtime patch generation process has completed, it signals the running replicas to reload their runtime patches. Thus, subsequent allocations in the same process will be patched on the fly without interrupting execution.

3.4.3. Cumulative Mode

While the replicated mode can isolate and correct errors on the fly in deployed applications, it may not be practical in all situations. For example, replicating applications with high resource requirements may cause unacceptable overhead. In addition, multithreaded or nondeterministic applications can exhibit different allocation activity and so cause object ids to diverge across replicas. To support these applications, Exterminator uses its third mode of operation, **cumulative** mode, which isolates errors without replication or multiple identical executions.

When operating in cumulative mode, Exterminator reasons about objects grouped by allocation and deallocation sites instead of individual objects, since objects are no longer guaranteed to be identical across different executions.

Because objects from a given site only occasionally cause errors, often at low frequencies, Exterminator requires more executions than in replicated or iterative mode in order to identify these low-frequency errors without a high false-positive rate. Instead of storing heap images from multiple runs, Exterminator computes relevant statistics about each run and stores them in its patch file. The retained data are on the order of a few kilobytes per execution, compared to tens or hundreds of megabytes for each heap image.

4. ITERATIVE AND REPLICATED ERROR ISOLATION

Exterminator employs two different families of error isolation algorithms: one set for replicated and iterative modes, and another for cumulative mode.

When operating in its replicated or iterative modes, Exterminator's probabilistic error isolation algorithm operates by searching for discrepancies across multiple heap images. Exterminator relies on corrupted canaries stored in freed objects to indicate the presence of an error. A corrupted canary (one that has been overwritten) can mean two things. If the same object (identified by object id) across all heap images has the same corruption, then the error is likely to be a dangling pointer. If canaries are corrupted in multiple freed objects, then the error is likely to be a buffer overflow. Exterminator limits the number of false positives for both overflows and dangling pointer errors.

4.1. Buffer overflow detection

Exterminator examines heap images looking for discrepancies across the heaps, both in overwritten canaries and in live objects. If an object is not equivalent across the heaps, Exterminator considers it to be a candidate **victim** of an overflow.

To identify victim objects, Exterminator compares the contents of equivalent objects, as identified by their object id across all heaps. Exterminator builds an **overflow mask** that comprises the discrepancies found across all heaps. However, because the same logical object may legitimately differ across multiple heaps, Exterminator must take care not to consider these occurrences as overflows.

First, a freed object may differ across heaps because it was filled with canaries only in some of the heaps. Exterminator uses the canary bitmap to identify this case.

Second, an object can contain pointers to other objects, which are randomly located on their respective heaps. Exterminator uses both deterministic and probabilistic techniques to distinguish integers from pointers. Briefly, if a value interpreted as a pointer points inside the heap area and points to the same logical object across all heaps, then Exterminator considers it to be the same logical pointer, and thus not a discrepancy. Exterminator also handles the case where pointers point into dynamic libraries, which newer versions of Linux place at pseudorandom base addresses.

Finally, an object can contain values that legitimately differ from process to process. Examples of these values include process ids, file handles, pseudorandom numbers, and pointers in data structures that depend on addresses (e.g., some red-black tree implementations). When Exterminator examines an object and encounters any word that differs at the same position across all the heaps, it considers it to be legitimately different, and not an indication of buffer overflow.

For small overflows, the risk of missing an overflow by ignoring overwrites of the same objects across multiple heaps is low:

THEOREM 1. *Let k be the number of heap images, S the length (in number of objects) of the* **overflow string**, *and H the number of objects on the heap. Then the probability of an overflow overwriting an object on all k heaps is*

$$P(\textit{identical overflow}) \varpi H \times (S/H)^k.$$

PROOF. This result holds for a stronger adversary than usual—rather than assuming a single contiguous overflow, we allow an attacker to arbitrarily overwrite any S distinct objects. Consider a given object a. On each heap, S objects are corrupted at random. The probability that object i is corrupted on a single heap is (S/H). Call E_i the event that object i is corrupted across all heaps; the probability $P(E_i)$ is $(S/H)^k$. The probability that at least one object is corrupted across all the heaps is $P(\cup_i E_i)$, which by a straightforward union bound is at most $\Sigma_i P(E_i) = H \times (S/H)^k$. □

We now bound the worst-case false-negative rate for buffer overflows; that is, the odds of not finding a buffer overflow because it failed to overwrite any canaries.

THEOREM 2. *Let M be the heap multiplier, so a heap is never more than $1/M$ full. The likelihood that an overflow of length b bytes fails to be detected by comparison against a canary is at most:*

$$P(\textit{missed overflow}) \leq \left(1 - \frac{M-1}{2M}\right)^k + \frac{1}{256^b}.$$

PROOF. Each heap is at least $(M - 1)/M$ free. Since DieFast fills free space with canaries with $P = 1/2$, the fraction of each heap filled with canaries is at least $(M - 1)/2M$. The likelihood of a random write not landing on a canary across all k heaps is thus at most $(1 - (M - 1)/2M)^k$. The overflow string could also match the canary value. Since the canary is randomly chosen, the odds of this are at most $(1/256)^b$. □

4.2. Culprit identification

At this point, Exterminator has identified the possible victims of overflows. For each victim, it scans the heap images for a matching **culprit**, the object that is likely to be the source of the overflow into a victim. Because Exterminator assumes that overflows are deterministic when operating in iterative or replicated mode, the culprit must be the same distance δ bytes away from the victim in every heap image. In addition, Exterminator requires that the overflowed values have some bytes in common across the images, and ranks them by their similarity. Note that, while Exterminator only considers positive values of δ, these values may be arbitrarily large.

Exterminator checks every other heap image for the candidate culprit, and examines the object that is the same δ bytes forward. If that object is free and should be filled with canaries but they are not intact, then it adds this culprit–victim pair to the candidate list.

We now bound the false-positive rate. Because buffer overflows can be discontiguous, every object in the heap that precedes an overflow is a potential culprit. However, each additional heap dramatically lowers this number.

Theorem 3. *The expected number of objects (possible culprits) the same distance δ from any given victim object across k heaps is*

$$E(\textit{possible culprits}) = \frac{1}{(H-1)^{k-2}}.$$

Proof. Without loss of generality, assume that the victim object occupies the last slot in every heap. An object can thus be in any of the remaining $n = H - 1$ slots. The odds of it being in the same slot in k heaps is $p = 1/(H - 1)^{k-1}$. This is a binomial distribution, so $E(\text{possible culprits}) = np = 1/(H - 1)^{k-2}$. □

With only one heap image, all $(H - 1)$ objects are potential culprits, but one additional image reduces the expected number of culprits for any victim to just 1 $(1/(H - 1)^0)$, effectively eliminating the risk of false positives.

Once Exterminator identifies a culprit–victim pair, it records the overflow size for that culprit as the maximum of any observed δ to a victim. Exterminator also assigns each culprit–victim pair a score that corresponds to its confidence that it is an actual overflow. This score is $1 - (1/256)^S$, where S is the sum of the length of detected overflow strings across all pairs. Intuitively, small overflow strings (e.g., 1 byte) detected in only a few heap images are given lower scores, and large overflow strings present in many heap images get higher scores.

After overflow processing completes and at least one culprit has a nonzero score, Exterminator generates a runtime patch for an overflow from the most highly ranked culprit.

4.3. Dangling pointer isolation

Isolating dangling pointer errors falls into two cases: a program may *read and write* to the dangled object, leaving it partially or completely overwritten, or it may only *read* through the dangling pointer. Exterminator does not handle read-only dangling pointer errors in iterative or replicated mode because it would require too many replicas (e.g., around 20; see Section 7.2). However, it handles overwritten dangling objects straightforwardly.

When a freed object is overwritten with identical values across multiple heap images, Exterminator classifies the error as a dangling pointer overwrite. (As Theorem 1 shows, this situation is highly unlikely to occur for a buffer overflow.) Exterminator then generates an appropriate runtime patch, as Section 6.2 describes.

5. CUMULATIVE ERROR ISOLATION

Unlike iterative and replicated mode, cumulative mode focuses on detecting, isolating, and correcting errors that happen in the field. In this context, replication, identical inputs, and deterministic execution are infeasible. Worse, program errors may manifest themselves in ways that are inherently hard to detect. For example, a program that reads a canary written into a free object may fail immediately, or may execute incorrectly for some time.

Our approach to error detection in this mode is to consider exceptional program events, such as premature termination, raising unexpected signals, etc., to be evidence that memory was corrupted during execution. We counter the lack of error reproducibility in these cases with statistical accumulation of evidence before assuming an error needs to be corrected. Exterminator isolates memory errors in cumulative mode by computing summary information accumulated over multiple executions, rather than by operating over multiple heap images.

5.1. Buffer overflow detection

Exterminator's buffer overflow isolation algorithm proceeds in three phases. First, it identifies heap corruption by looking for overwritten canary values. Second, for each allocation site, it computes an estimate of the probability that an object from that site could be the source of the corruption. Third, it combines these independent estimates from multiple runs to identify sites that consistently appear as candidates for causing the corruption.

Exterminator's randomized allocator allows us to compute the probability of certain properties in the heap. For example, the probability of an object occurring on a given miniheap can be estimated given the miniheap size and the number of miniheaps. If objects from some allocation site are sources of overflows, then those objects will occur on miniheaps containing corruptions more often than expected. Exterminator tracks how often objects from each allocation site occur on corrupted miniheaps across multiple runs. Using this information, it uses a statistical hypothesis test that identifies sites that occur with corruption too often to be random chance, and identifies them as overflow culprits (see Novark[11] for more details).

Once Exterminator identifies an erroneous allocation site, it produces a runtime patch that corrects the error. To find the correct pad size, it searches backward from the corruption found during the current run until it finds an object allocated from the site. It then uses the distance between that object and the end of the corruption as the pad size.

5.2. Dangling pointer isolation

As with buffer overflows, Exterminator's dangling pointer isolator computes summary information over multiple runs. To force each run to have a different effect, Exterminator fills freed objects with canaries with some probability p, turning every execution into a series of Bernoulli trials. In this scenario, if the program reads canary data through the dangling pointer, the program may crash. Thus writing the canary for that object increases the probability that the program will later crash. Conversely, if an object is not freed prematurely, then overwriting it with canaries has no influence on the failure or success of the program. Exterminator then uses the same hypothesis testing framework as its buffer overflow algorithm to identify sources of dangling pointer errors.

The choice of p reflects a trade-off between the precision of the buffer overflow algorithm and dangling pointer isolation. Since overflow isolation relies on detecting corrupt canaries, low values of p increase the number of runs (though not the number of *failures*) required to isolate overflows. However, lower values of p increase the precision of dangling pointer isolation by reducing the risk that certain allocation sites (those that allocate large numbers of objects) will always observe one canary value. We currently set $p = 1/2$, though some dangling pointer errors may require lower values of p to converge within a reasonable number of runs.

Exterminator then estimates the required lifetime extension by locating the oldest canaried object from an identified allocation site, and computing the number of allocations between the time it was freed and the time that the program failed. The correcting allocator then extends the lifetime of all objects corresponding to this allocation/deallocation site by twice this number.

6. ERROR CORRECTION

We now describe how Exterminator uses the information from its error isolation algorithms to correct specific errors. Exterminator first generates runtime patches for each error. It then relies on a correcting allocator that uses this information, padding allocations to prevent overflows, and deferring deallocations to prevent dangling pointer errors.

Exterminator's ability to correct memory errors has several inherent limitations. Exterminator can only correct finite overflows, because it tries to contain any given overflow by finite over-allocation. Similarly, Exterminator corrects dangling pointer errors by inserting finite delays before freeing particular objects. Finally, Exterminator cannot correct memory errors when the evidence it uses to locate these errors is destroyed, such as when an overflow overwrites most of the heap, or when a program with a dangling pointer error runs long enough to reallocate the dangled object.

6.1. Buffer overflow correction

For every culprit–victim pair that Exterminator encounters, it generates a runtime patch consisting of the allocation site hash and the amount of padding needed to contain the overflow (δ + the size of the overflow string). If a runtime patch has already been generated for a given allocation site, Exterminator uses the maximum padding value encountered so far.

6.2. Dangling pointer correction

The runtime patch for a dangling pointer consists of the combination of its allocation site hash and an amount of time by which to delay its deallocation. Exterminator computes this delay as follows. Let τ be the recorded deallocation time of the dangled object, and T be the allocation time at which the program crashed or Exterminator detected heap corruption. Exterminator has no way of knowing how long the object is supposed to live, so computing an exact delay is impossible. Instead, it extends the object's lifetime (delays its freeing) by twice the distance between its premature freeing and the time of crashing or detection, plus one: $2 \times (T - \tau) + 1$.

It is important to note that this deallocation deferral does not multiply object lifetimes but rather their *drag*.[15] To illustrate, an object might live for 1000 allocations and then be freed just 10 allocations too soon. If the program immediately crashes, Exterminator will extend its lifetime by 21 allocations, increasing its correct lifetime (1010 allocations) by less than 1% (1021/1010).

6.3. The correcting memory allocator

The correcting memory allocator incorporates the runtime patches described above and applies them when appropriate.

At start-up, or upon receiving a reload signal (Section 3.4), the correcting allocator loads the runtime patches from a specified file. It builds two hash tables: a **pad table** mapping allocation sites to pad sizes, and a **deferral table** mapping pairs of allocation and deallocation sites to a deferral value. Because it can reload the runtime patch file and rebuild these tables on the fly, Exterminator can apply patches to running programs without interrupting their execution. This aspect of Exterminator's operation may be especially useful for systems that must be kept running continuously.

On every deallocation, the correcting allocator checks to see if the object to be freed needs to be deferred. If it finds a deferral value for the object's allocation and deallocation site, it pushes onto the **deferral priority queue** the pointer and the time to actually free it (the current allocation time plus the deferral value).

The correcting allocator checks the deferral queue on every allocation to see if any object should now be freed. It then checks whether the current allocation site has an associated pad size. If so, it adds the pad size to the allocation request, and forwards the allocation request to the underlying allocator.

6.4. Collaborative correction

Each individual user of an application is likely to experience different errors. To allow an entire user community to automatically improve software reliability, Exterminator provides a simple utility that supports collaborative correction. This utility takes as input a number of runtime patch files. It then combines these patches by computing the maximum pad size required for any allocation site, and the maximal deferral amount for any given allocation site/deallocation site pair. The result is a new runtime patch file that covers all observed errors. Because the size of patch files is limited by the number of allocation sites in a program, we expect these files to be compact and practical to transmit. For example, the size of the runtime patches that Exterminator generates for injected errors in `espresso` was just 130K (17K when compressed with gzip).

7. RESULTS

Our evaluation answers the following questions: (1) What is the runtime overhead of using Exterminator? (2) How effective is Exterminator at finding and correcting memory errors, both for injected and real faults?

7.1. Exterminator runtime overhead

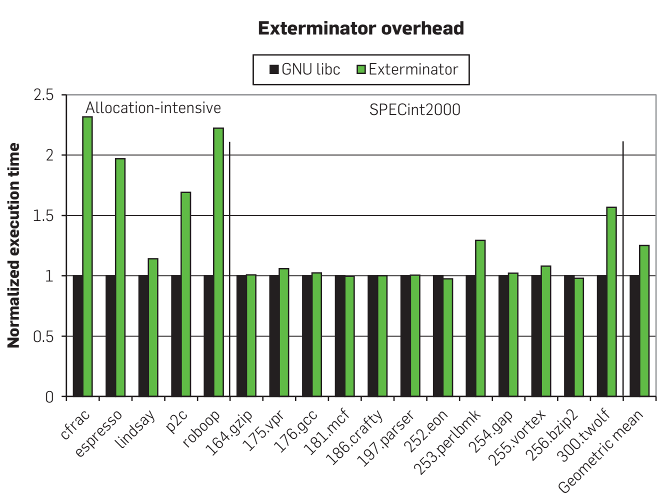
We evaluate Exterminator's performance with the SPECint2000 suite[16] running reference workloads, as well as a suite of allocation-intensive benchmarks. We use the latter suite of benchmarks both because they are widely used in memory management studies and because their high allocation-intensity stresses memory management performance. For all experiments, we fix Exterminator's heap multiplier (value of M) at 2.

All results are the average of five runs on a quiescent, dual-processor Linux system with 3GB of RAM, with each 3.06 GHz Intel Xeon processor (hyperthreading active) equipped with 512K L2 caches. Our observed experimental variance is below 1%.

We focus on the nonreplicated mode (iterative/cumulative), which we expect to be a key limiting factor for Exterminator's performance and the most common usage scenario.

We compare the runtime of Exterminator (DieFast plus the correcting allocator) to the GNU libc allocator. This allocator is based on the Lea allocator,[8] which is among the fastest available.[5] Figure 4 shows that, versus this allocator, Exterminator degrades performance by from 0% (`186.crafty`) to 132% (`cfrac`), with a geometric mean of 25.1%. While Exterminator's overhead is substantial for the allocation-intensive suite (geometric mean: 81.2%), for which the cost of computing allocation and deallocation contexts dominates, its overhead is significantly less pronounced across the SPEC benchmarks (geometric mean: 7.2%).

Figure 4: Runtime overhead for Exterminator across a suite of benchmarks, normalized to the performance of GNU libc (Linux) allocator.



7.2. Memory error correction

7.2.1. Injected Faults

To measure Exterminator's effectiveness at isolating and correcting bugs, we used the fault injector that accompanies the DieHard distribution to inject buffer overflows and dangling pointer errors. For each data point, we run the injector using a random seed until it triggers an error or divergent output. We next use this seed to deterministically trigger a single error in Exterminator, which we run in iterative mode. We then measure the number of iterations required to isolate and generate an appropriate runtime patch. The total number of images (iterations plus the first run) corresponds to the number of replicas that would be required when running Exterminator in replicated mode.

Note that Exterminator's approach to correcting memory errors does not impose additional execution time overhead in the presence of patches. However, it can consume additional space, either by padding allocations or by deferring deallocations.

Buffer overflows: We triggered 10 different buffer overflows each of three different sizes (4, 20, and 36 bytes) by intentionally undersizing objects in the `espresso` benchmark. In every case, three images were required to isolate and correct these errors. Notice that this result is substantially better than the analytical worst case: for three images, Theorem 2 bounds the worst-case likelihood of missing an overflow to 42% (Section 4.1), but we observed a 0% false-negative rate. The most space overhead we observe is a total increase of 2816 bytes.

Dangling pointer errors: We then triggered 10 dangling pointer faults in `espresso` with Exterminator running in iterative and in cumulative modes. Recall that in iterative mode, Exterminator always fills freed objects with canaries, while it does so probabilistically when running in cumulative mode (see Section 3.3).

In iterative mode, Exterminator succeeds in isolating the error in only four runs. In another four runs, `espresso` does not write through the dangling pointer. Instead, it reads a canary value through the dangled pointer, treats it as valid data, and either crashes or aborts. Since no corruption is present in the heap, Exterminator cannot isolate the source of the error. In the remaining two runs, writing canaries into the dangled object triggers a cascade of errors that corrupt large segments of the heap. In these cases, the corruption destroys the information that Exterminator requires to isolate the error.

However, in cumulative mode, probabilistic canary-filling enables Exterminator to isolate all injected errors, including the read-only dangling pointer errors. For runs where no large-scale heap corruption occurs, Exterminator requires between 22 and 30 executions to isolate and correct the errors. In each case, 15 failures must be observed before the erroneous site pair crosses the likelihood threshold. Because objects are overwritten randomly, the number of runs required to yield 15 failures varies. Where writing canaries corrupts a large fraction of the heap, Exterminator requires

18 failures and 34 total runs. In some of the runs, execution continues long enough for the allocator to reuse the culprit object, preventing Exterminator from observing that it was overwritten.

The space overhead of the derived runtime patches ranges from 32 to 1024 bytes (one 256-byte object is deferred for four deallocations). This amount constitutes less than 1% of the maximum memory consumed by the application.

7.2.2. Real Faults

We also tested Exterminator with actual bugs in two applications: the Squid Web cache server and the Mozilla Web browser.

Squid Web cache: Version 2.3s5 of Squid has a buffer overflow; certain inputs cause Squid to crash with either the GNU libc allocator or the Boehm–Demers–Weiser collector.

We run Squid three times under Exterminator in iterative mode with an input that triggers a buffer overflow. Exterminator continues executing correctly in each run, but the overflow corrupts a canary. Exterminator's error isolation algorithm identifies a single allocation site as the culprit and generates a pad of exactly 6 bytes, fixing the error.

Mozilla Web browser: We also tested Exterminator's cumulative mode on a known heap overflow in Mozilla 1.7.3/Firefox 1.0.6 and earlier. This overflow (bug 307259) occurs because of an error in Mozilla's processing of Unicode characters in domain names. Not only is Mozilla multithreaded, leading to nondeterministic allocation behavior, but even slight differences in moving the mouse cause allocation sequences to diverge. Thus, neither replicated nor iterative modes can identify equivalent objects across multiple runs.

We perform two case studies that represent plausible scenarios for using Exterminator's cumulative mode. In the first study, the user starts Mozilla and immediately loads a page that triggers the error. This scenario corresponds to a testing environment where a proof-of-concept input is available. In the second study, the user first navigates through a selection of pages (different on each run), and then visits the error-triggering page. This scenario approximates deployed use where the error is triggered in the wild.

In both cases, Exterminator correctly identifies the overflow with no false positives. In the first case, Exterminator requires 23 runs to isolate the error. In the second, it requires 34 runs. We believe that this scenario requires more runs because the site that produces the overflowed object allocates more correct objects, making it harder to identify it as erroneous.

8. CONCLUSION

This paper presents Exterminator, a system that automatically corrects heap-based memory errors in C and C++ programs with high probability. Exterminator operates entirely at the runtime level on unaltered binaries, and consists of three key components: (1) DieFast, a probabilistic debugging allocator, (2) a probabilistic error isolation algorithm, and (3) a correcting memory allocator. Exterminator's probabilistic error isolation isolates the source and extent of memory errors with provably low false-positive and false-negative rates. Its correcting memory allocator incorporates runtime patches that the error isolation algorithm generates to correct memory errors. Exterminator not only is suitable for use during testing, but also can automatically correct deployed programs.

Acknowledgments

We thank Sam Guyer, Mike Hicks, Erik Learned-Miller, Sarah Osentoski, Martin Rinard, and Guy Steele for their valuable feedback. This material is based upon work supported by Intel, Microsoft Research, and the National Science Foundation under CAREER Award CNS-0347339 and CNS-0615211. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. C

References

1. Austin, T.M., Breach, S.E., and Sohi, G.S. Efficient detection of all pointer and array access errors. In *Proceedings of the 1994 ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM Press, June 1994, 290–301.
2. Avots, D., Dalton, M., Livshits, V.B., and Lam, M.S. Improving software security with a C pointer analysis. In *Proceedings of the 27th International Conference on Software Engineering*, ACM Press, May 2005, 332–341.
3. Berger, E.D. and Zorn, B.G. DieHard: Probabilistic memory safety for unsafe languages. In *Proceedings of the 2006 ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM Press, June 2006, 158–168.
4. Berger, E.D. and Zorn, B.G. Efficient probabilistic memory safety. Technical Report UMCS TR-2007-17, Department of Computer Science, University of Massachusetts Amherst, Mar. 2007.
5. Berger, E.D. Zorn, B.G., and McKinley, K.S. Composing high-performance memory allocators. In *Proceedings of the 2001 ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM Press, June 2001, 114–124.
6. Dhurjati, D., Kowshik, S., and Adve, V. SAFECode: Enforcing alias analysis for weakly typed languages. In *Proceedings of the 2006 ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM Press, June 2006, 144–157.
7. Hastings, R. and Joyce, B. Purify: Fast detection of memory leaks and access errors. In *Proceedings of the Winter 1992 USENIX Conference*, USENIX, Jan. 1992, 125–138.
8. Lea, D. A memory allocator. http://gee.cs.oswego.edu/dl/html/malloc.html, 1997.
9. Necula, G.C., McPeak, S., and Weimer, W. CCured: Type-safe retrofitting of legacy code. In *Proceedings of the 29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ACM Press, Jan. 2002, 128–139.
10. Nethercote, N. and Seward, J. Valgrind: A framework for heavyweight dynamic binary instrumentation. In *Proceedings of 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM Press, June 2007, 89–100.
11. Novark, G., Berger, E.D., and Zorn, B.G. Exterminator: Automatically correcting memory errors with high probability. In *Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM Press, June 2007, 1–11.
12. Qin, F., Tucek, J., Sundaresan, J., and Zhou, Y. Rx: Treating bugs as allergies—a safe method to survive software failures. In *Proceedings of the Twentieth Symposium on Operating Systems Principles*, vol. XX of *Operating Systems Review*, ACM Press, Oct. 2005, 235–248.
13. Rinard, M., Cadar, C., Dumitran, D., Roy, D.M., and Leu, T. A dynamic technique for eliminating buffer overflow vulnerabilities (and other memory errors). In *Proceedings of the 20th Annual Computer Security Applications Conference*, IEEE Computer Society, Dec. 2004, 82–90.
14. Rinard, M., Cadar, C., Dumitran, D., Roy, D.M., Leu, T., and Beebee, W.S. Jr. Enhancing server availability and security through failure-oblivious computing. In *Sixth Symposium on Operating Systems Design and Implementation*, USENIX, Dec. 2004, 303–316.
15. Röjemo, N. and Runciman, C. Lag, drag, void, and use: heap profiling and space-efficient compilation revisited. In *Proceedings of First International Conference on Functional Programming*, ACM Press, May 1996, 34–41.
16. Standard Performance Evaluation Corporation. SPEC2000. http://www.spec.org.
17. Symantec. Internet security threat report. http://www.symantec.com/enterprise/threatreport/index.jsp, Sept. 2006.

Gene Novark (gnovark@cs.umass.edu) Department of Computer Science, University of Massachusetts, Amherst, MA.

Emery D. Berger (emery@cs.umass.edu) Department of Computer Science, University of Massachusetts, Amherst, MA.

Benjamin G. Zorn (zorn@microsoft.com) Microsoft Research, Redmond, WA.

Angelo State University, Member, Texas Tech University System
Computer Science Department
Assistant/Associate Professor Position

The Computer Science Department seeks applications for a nine-month, tenure-track position. Teach twelve semester credit hours of undergraduate CS courses each semester and engage in scholarly activities and university service. Appointment date is August 21, 2009.

For more information, see the position announcement at:
http://www.angelo.edu/publications/employment/documents/0909-009.pdf

Additional information about Angelo State University can be found at:
http://www.angelo.edu

AT&T Labs – Research: Come Explore Our World!

AT&T Labs - Research, one of the premier industrial research laboratories in the world, is looking for talented individuals to make a difference in the world of communications. We engage in core computer science, focusing on leading-edge research in networking and services, and information and software systems. For more information visit www.research.att.com and click on **"Career Opportunities."**

Baylor University
Assistant to Full Professor of Computer Science

The Department of Computer Science seeks a productive scholar and dedicated teacher in the area of Game and Simulated Environments for a tenure-track position beginning August, 2009. The ideal candidate will hold a terminal degree in Computer Science or a closely related field, demonstrate scholarly capability in game design and development, and exhibit a passion for teaching and mentoring at the graduate and undergraduate level. For position details and application information please visit: http://cs.baylor.edu/employment

Baylor, the world's largest Baptist university, holds a Carnegie classification as a "high-research" institution. Baylor's mission is to educate men and women for worldwide leadership and service by integrating academic excellence and Christian commitment within a caring community. Baylor is actively recruiting new faculty with a strong commitment to the classroom and an equally strong commitment to discovering new knowledge as Baylor aspires to become a top tier research university while reaffirming and deepening its distinctive Christian mission as described in Baylor 2012 (http://www.baylor.edu/vision/).

Baylor is a Baptist university affiliated with the Baptist General Convention of Texas. As an AA/EEO employer, Baylor encourages minorities, women, veterans, & persons with disabilities to apply.

Baylor University
Assistant to Full Professor of Bioinformatics

The Baylor Department of Computer Science seeks a dynamic scholar to fill this position beginning August, 2009. For complete position details and application information please visit: http://cs.baylor.edu/employment.

Candidates should possess an earned doctorate in Bioinformatics, Computer Science, or a related field, and have sufficient expertise and experience to teach and perform research in the area of Bioinformatics. The successful candidate will be expected to establish, or bring, an active independently-funded research program in areas related to systems biology, genomics, or other areas related to ongoing research within the Department of Computer Science. The successful candidate will also assist in the teaching of graduate and undergraduate computer science and bioinformatics students. Your application should consist of a letter of interest, curriculum vitae, transcripts (as appropriate) and a list of three references. Salary will be commensurate with experience and qualifications.

Baylor, the world's largest Baptist university, holds a Carnegie classification as a "high-research" institution. Baylor's mission is to educate men and women for worldwide leadership and service by integrating academic excellence and Christian commitment within a caring community. Baylor is actively recruiting new faculty with a strong commitment to the classroom and an equally strong commitment to discovering new knowledge as Baylor aspires to become a top tier research university while reaffirming and deepening its distinctive Christian mission as described in Baylor 2012 (http://www.baylor.edu/vision/).

Baylor is a Baptist university affiliated with the Baptist General Convention of Texas. As an AA/EEO employer, Baylor encourages minorities, women, veterans, & persons with disabilities to apply.

Boston University
Department of Electrical & Computer Engineering
Faculty Search Classified Advertisement

The Department of Electrical & Computer Engineering (ECE) at Boston University (BU) is seeking candidates for several anticipated faculty positions at all ranks in computer engineering, including computer systems, embedded and reconfigurable systems, distributed systems, trusted computing, design automation, VLSI, computer networks, and software engineering, as well as other ECE disciplines such as information sciences, photonics, solid-state devices and nanostructures, and related areas. The Department is seeking to foster growth in the broad, interdisciplinary topics of energy, health, and communications; candidates with research interests that transcend the traditional boundaries of ECE are strongly encouraged to apply.

ECE at BU is a world-class, progressive department with excellent resources that is steadily gaining national and international prominence for its exceptional research and education record. ECE is part of BU's rapidly growing and innovative College of Engineering, and currently consists of 44 faculty members, 200 graduate students, and 250 BS majors. Outstanding collaboration opportunities are available with nationally recognized medical centers and universities/colleges, nearby research centers, and industry throughout the Boston area. Joint appointments with other BU departments and with the newly formed Divisions of Material Science & Engineering and Systems Engineering are also possible.

Qualified candidates must possess a relevant, earned PhD, and have a demonstrable ability to teach effectively, develop funded research programs in their area of expertise, and contribute to the tradition of excellence in research that is characteristic of the ECE Department. Self-motivated individuals who thrive on challenge and are eager to utilize their expertise to strengthen an ambitious program of departmental growth are desired. Women, minorities, and candidates from other underrepresented groups are especially encouraged to apply and help us continue building an exceptional 21st century university department.

Beyond its research and academic activities, BU has a lively, urban campus situated along the banks of the Charles River in Boston's historic Fenway-Kenmore neighborhood. The campus and surrounding areas offer limitless opportunities for recreational activities, from world-class art and performances to sporting events and fine dining.

Applicants should send their curriculum vitae with a statement of teaching and research plans, as well as the names and contact information of three references to: Professor Franco Cerrina, Chair, Department of Electrical & Computer Engineering, Boston University, 8 Saint Mary's Street, Room 324, Boston, MA 02215. Materials may also be sent in PDF format via email to rbell@bu.edu. Boston University is an Equal Opportunity/Affirmative Action Employer.

For additional information, please visit http://www.bu.edu/ece/.

Bridgewater State College
Department of Mathematics and Computer Science
Assistant Professor in Computer Science

Bridgewater State College seeks applicants for a full-time tenure-track faculty position, which includes opportunities for teaching and advising undergraduate and graduate students.

Position: Assistant Professor in Computer Science,

Department of Mathematics and Computer Science.
Responsibilities: Responsibilities include teaching undergraduate and master's level courses, advising undergraduates and supervising graduate students, participating in departmental curriculum development, as well as engaging in ongoing scholarship and professional activity.
Required Minimum Qualifications:
A Ph.D. in Computer Science or a closely related field such as Electrical or Computer Engineering is required. Applicants should be strongly committed to excellence in teaching and advising, and to working in a multicultural environment that fosters diversity.
Preferred Qualifications:
While all specialties will be considered, topics such as databases, wireless networking, geospatial analysis, web-based information retrieval, parallel processing and computer or network security will be given additional consideration.
TO APPLY: Please apply on-line at http://jobs.bridgew.edu

Position will remain open until filled. Resumes received by December 1 will receive priority.

Bridgewater State College is an affirmative action/equal opportunity employer which actively seeks to increase the diversity of its workforce.

Bucknell University
Assistant Professor
Department of Computer Science

Applications are invited for a tenure track entry-level (three or fewer years of full-time teaching experience) assistant professor position in computer science beginning mid-August 2009. Candidates must have completed their Ph.D. requirements in computer science or a closely related field by August 15th, 2009. A strong commitment to excellence in teaching and scholarship is required. The successful candidate must be able to participate in the teaching of required core courses and be able to develop elective courses in the candidate's area of expertise that enhance the department's course offerings.

Bucknell is a highly selective private university emphasizing quality undergraduate education in engineering and in liberal arts and sciences. The B.S. programs in computer science are ABET and CAC accredited. The computing environment is Linux/Unix-based. More information about the department can be found at:

http://www.bucknell.edu/ComputerScience/

Applications will be considered as received and recruiting will continue until the position is filled. Candidates are asked to submit a cover letter, CV, graduate transcript, a statement of teaching philosophy and research interests, and the contact information for three references. Please submit your application to

http://jobs.bucknell.edu/

by searching for the "Computer Science Faculty Position".

Please direct any questions to Professor Xiannong Meng of the Computer Science Department at xmeng@bucknell.edu.

Bucknell University values a diverse college community and is committed to excellence through diversity in its faculty, staff and students. An Equal Opportunity/Affirmative Action Employ-

---

UNIVERSITÄT DES SAARLANDES

M²CI CLUSTER OF EXCELLENCE

Saarland University is seeking to establish several

Junior Research Groups (W1/W2)

within the recently established Cluster of Excellence "Multimodal Computing and Interaction" which was established by the German Research Foundation (DFG) within the framework of the German Excellence Initiative.

The term "multimodal" describes the different types of digital information such as text, speech, images, video, graphics, and high-dimensional data, and the way it is perceived and communicated, particularly through vision, hearing, and human expression. The challenge is now to organize, understand, and search this multimodal information in a robust, efficient and intelligent way, and to create dependable systems that allow natural and intuitive multimodal interaction. We are looking for highly motivated young researchers with a background in the research areas of the cluster, including algorithmic foundations, secure and autonomous networked systems, open science web, information processing in the life sciences, visual computing, large-scale virtual environments, synthetic virtual characters, text and speech processing and multimodal dialog systems. Additional information on the Cluster of Excellence is available on http://www.mmci.uni-saarland.de. Group leaders will receive junior faculty status at Saarland University, including the right to supervise Bachelor, Master and PhD students. Positions are limited to five years.

Applicants for W1 positions (phase I of the program) must have completed an outstanding PhD. Upon successful evaluation after two years, W1 group leaders are eligible for promotion to W2. Direct applicants for W2 positions (phase II of the program) must have completed a postdoc stay and must have demonstrated outstanding research potential and the ability to successfully lead their own research group. Junior research groups are equipped with a budget of 80k to 100k Euros per year to cover research personnel and other costs.

Saarland University has leading departments in computer science and computational linguistics, with more than 200 PhD students working on topics related to the cluster (see http://www.informatik-saarland.de for additional information). The German Excellence Initiative recently awarded multi-million grants to the Cluster of Excellence "Multimodal Computing and Interaction" as well as to the "Saarbrücken Graduate School of Computer Science". An important factor to this success were the close ties to the Max Planck Institute for Computer Science, the German Research Center for Artificial Intelligence (DFKI), and the Max Planck Institute for Software Systems which are co-located on the same campus.

Candidates should submit their application (curriculum vitae, photograph, list of publications, short research plan, copies of degree certificates, copies of the five most important publications, list of five references) to the coordinator of the cluster, Prof. Hans-Peter Seidel, MPI for Computer Science, Campus E1 4, 66123 Saarbrücken, Germany. Please, also send your application as a single PDF file to applications@mmci.uni-saarland.de.

The review of applications will begin on January 15, 2009, and applicants are strongly encouraged to submit applications by that date; however, applications will continue to be accepted until January 31, 2009. Final decisions will be made following a candidate symposium that will be held during March 9 – 13, 2009.

Saarland University is an equal opportunity employer. In accordance with its policy of increasing the proportion of women in this type of employment, the University actively encourages applications from women. For candidates with equal qualification, preference will be given to people with physical disabilities.

er, Bucknell University especially welcomes applications from women and minority candidates.

California Institute of Technology
Center For The Physics of Information
Postdoctoral Research Positions

The Center for the Physics of Information at the California Institute of Technology will have postdoctoral scholar positions available beginning in September 2009. Researchers interested in all aspects of the interface between information science and physical science are invited to apply. The appointment is contingent upon completion of Ph.D.

Please apply on-line at http://www.ist.caltech.edu/joinus/positions.html#postdoc.

Electronic copies of your curriculum vitae, publication list, statement of research interests, and three letters of recommendation are required. The deadline for receipt of all application materials is December 15, 2008.

The California Institute of Technology is an Equal Opportunity/Affirmative Action employer. Women, minorities, veterans and disabled persons are encouraged to apply.

California Institute of Technology
Institute For Quantum Information
Postdoctoral Research Positions

The Institute for Quantum Information at the California Institute of Technology will have postdoctoral scholar positions available beginning in September 2009. Researchers interested in all aspects of quantum information science are invited to apply. The appointment is contingent upon completion of Ph.D.

Please apply on-line at http://www.iqi.caltech.edu/postdoc_opening.html. Electronic copies of your curriculum vitae, publication list, statement of research interests, and three letters of recommendation are required. The deadline for receipt of all application materials is December 15, 2008.

The California Institute of Technology is an Equal Opportunity/Affirmative Action employer. Women, Minorities, Veterans and Disabled Persons are encouraged to apply.

Calvin College
Tenure-track faculty position

The Department of Computer Science at Calvin College invites applications for a Fall 2009 tenure-track position. Applicants who hold a PhD in Computer Science, Information Systems or a related field and who want to join a collegial, stimulating, undergraduate program at a reformed Christian college are invited to discover more at: http://cs.calvin.edu/p/facultyOpening2009/.

Carnegie Mellon University
School of Computer Science
Teaching-Track Position

The School of Computer Science at Carnegie Mellon University invites applications for an anticipated teaching-track position beginning in the fall term of 2009. This is a career-oriented, renewable appointment that is responsible for, and committed to, developing and delivering excellent first and second year courses in computer science. In particular, we seek candidates with proven records of quality teaching in data structures, algorithms, and introductory programming courses.

The person filling this position will become part of a group of 12 teaching-track faculty, with dedicated administrative and technical support staff. The facilities include state-of-the-art computer classrooms and infrastructure for course delivery. Currently the programming courses use Java and C as the teaching languages, while the data structures courses use Java as the teaching language.

Applicants for the position must have an M.S. or Ph.D. in Computer Science or a related field, and demonstrated excellence in teaching Computer Science courses. Applicants seeking a tenure-track position at a research university are not a good match for the needs of this position. Teaching-track appointments are typically at the rank of Lecturer, with the possibility of promotion to the ranks of Associate Teaching Professor and Teaching Professor. None of these teaching-track ranks are tenured.

In order to receive full consideration, applicants should submit a letter of application, curriculum vitae, a statement of teaching philosophy, and the names and email addresses of three or more individuals whom the applicant has asked to provide letters of reference. Applicants should arrange for reference letters to be sent directly to the contact below. This information should be sent by Dec 15, 2008.

Additionally, applicants are encouraged to submit a video sample of their teaching. This enables applicants to add another dimension to their application. Since the person who will eventually fill this position will be expected to be an excellent classroom teacher, the video sample is an opportunity for candidates to show off their talents in a way other than traditional "on paper" means.

For more information see:
http://www.intro.cs.cmu.edu/position.html.

Send materials, letters, and questions to Scott McElfresh.

Email: scottm@cs.cmu.edu

Paper:
Scott McElfresh
Carnegie Mellon University
Computer Science Department
Wean Hall
Pittsburgh, PA 15213

Carnegie Mellon is an affirmative action/equal opportunity employer and we invite and encourage applications from women and minorities.

Carnegie Mellon University
Tepper School of Business, Pittsburgh, PA
Information Systems Faculty
Tenure-Track Positions

Tepper School of Business (http://www.tepper.cmu.edu/) is seeking applicants to fill one or more tenure-track faculty positions in Information Systems, starting in September 2009. *We are looking for people who are interested in how IT may transform businesses, markets, and economic processes.* We have a particular interest in applicants with research and teaching interests in the development and application of machine learning techniques to business problems or in other related areas of business algorithmics. Applicants may hold a doctoral degree in any business discipline, Information Systems, Computer Science, Economics or Operations Research. While we are primarily seeking candidates at the Assistant Professor level, we will consider outstanding applicants at the Associate and Full Professor level. Applicants for Assistant level should have completed or be nearing completion of a Ph.D., and should demonstrate potential of excellence in research and teaching. We are also interested in candidates who have already served a few years as Assistant Professor. Applicants for higher ranks are expected to have a demonstrated track record in research and teaching. Teaching assignments encompass BS, Masters, and Ph.D. programs. Applicants should send a current curriculum vita, three letters of recommendation, and evidence of research such as publications, working papers, or dissertation proposal to: **Phillip Conley, Information Systems Faculty Recruiting, Carnegie Mellon University, Tepper School of Business, Room 369 Posner Hall, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890 (Phone: 412-268-6212). Applications must be received by January 15, 2009 (submissions only accepted by mail).** *Carnegie Mellon is an equal opportunity, affirmative action employer with particular interest in identifying women and minority applicants for faculty positions.*

Department of Electrical and Systems Engineering

Penn Engineering

Physical Devices

The University of Pennsylvania seeks outstanding individuals for tenure-track or tenured faculty positions in the Department of Electrical and Systems Engineering to start July 1, 2009. Applicants must have a Ph.D. in Engineering or equivalent fields. Suitable candidates in the areas of modeling, design, fabrication and characterization of nanostructures, devices, and circuits (e.g., nanophotonic, nanophononic, nanoelectronic, molecular electronic, nanomagnetic or other novel computational devices and systems) will be considered. Candidates should be prepared to collaborate with faculty in appropriate related areas such as materials science, bioengineering, physics, chemistry, chemical engineering, mechanical engineering, and/or computer science and engineering.

The University seeks individuals with exceptional promise for, or proven record of, research achievement who will excel in teaching undergraduate and graduate courses and take a position of international leadership in defining their field of study.

Interested persons should submit an application by completing the form located on the Faculty Recruitment Web site at **https://www.seas.upenn.edu/ese/fsrch/apply.html** including curriculum vitae, statements of research and teaching interests and plans, and the names of at least three references.

The University of Pennsylvania is an Equal Opportunity Employer. Minorities/Females/Individuals with Disabilities/Veterans are encouraged to apply.

Department of Electrical and Systems Engineering

Penn Engineering

Market Systems

The University of Pennsylvania seeks outstanding individuals for tenure-track or tenured faculty positions in the Department of Electrical and Systems Engineering to start July 1, 2009. Applicants must have a Ph.D. in Engineering, Applied Mathematics, Computer Science, Operations Research, or equivalent. This search focuses on candidates working at the mathematical and computational foundations of Market Systems Engineering - the formalization, analysis, optimization, and realization of interconnected systems that increasingly integrate engineering, computational, and economic systems and methods. We are particularly interested in candidates with a vision and interest in defining the research frontier and education of next-generation leaders in this rapidly growing field. Applicants with strong ties to industry are strongly encouraged.

The University seeks individuals with exceptional promise for, or proven record of, research achievement who will excel in teaching undergraduate and graduate courses and take a position of international leadership in defining their field of study. Leadership in cross-disciplinary collaborations is of particular interest.

Interested persons should submit an application by completing the form located on the Faculty Recruitment Web site at **https://www.seas.upenn.edu/ese/fsrch/apply.html** including curriculum vitae, and the names of at least three references.

The University of Pennsylvania is an Equal Opportunity Employer. Minorities/Females/Individuals with Disabilities/Veterans are encouraged to apply.

Department of Electrical and Systems Engineering

Penn Engineering

Computer Engineering

The University of Pennsylvania seeks outstanding individuals for tenure-track or tenured faculty positions in the Department of Electrical and Systems Engineering to start July 1, 2009. Applicants must have a Ph.D. in Engineering or Computer Science. Suitable candidates in the areas of computer engineering (e.g., circuit design, architecture, CAD, hardware and software systems, codesign) will be considered. We are particularly interested in candidates who are capable of cross-layer engineering and collaboration, bridging between novel device and manufacturing effects and application and system-level characteristics.

The University seeks individuals with exceptional promise for, or proven record of, research achievement who will excel in teaching undergraduate and graduate courses and take a position of international leadership in defining their field of study.

Interested persons should submit an application by completing the form located on the Faculty Recruitment Web site at **https://www.seas.upenn.edu/ese/fsrch/apply.html** including curriculum vitae, statements of research and teaching interests and plans, and the names of at least three references.

The University of Pennsylvania is an Equal Opportunity Employer. Minorities/Females/Individuals with Disabilities/Veterans are encouraged to apply.

Connecticut College
Computer Science Tenure-Track Faculty Position

Assistant (preferred) or associate/full professor position in computer science to continue to develop our growing collaborations between computer science and biology. Starting August 2009. See http://cs.conncoll.edu/job.html for details.

The College of Staten Island (CSI)
Assistant Professor of Scientific Computation

The College of Staten Island (CSI), a senior college of The City University of New York (CUNY) housing the CUNY High Performance Computing Center (HPC), seeks candidates for tenure-track positions as Assistant Professor of Scientific Computation to begin September 2009.

Demonstrated expertise in the use of modern high performance computational platforms and an earned Ph.D. in any area of biology, chemistry, computer science, engineering, mathematics, physics, or related scientific fields required. The successful candidate is expected to establish a vigorous externally funded research program in the appropriate academic area and to participate in CUNY doctoral programs. Preference will be given to individuals with demonstrated experience in interdisciplinary projects. Review of applications will begin immediately.

Send a letter of application, statement of research goals and teaching philosophy, curriculum vitae and at least three letters of reference under separate cover by December 16, 2008 to:

Professor Andrew Poje, Chair
Scientific Computation Search Committee
Office of the Dean, Room 1A-312
College of Staten Island
2800 Victory Boulevard
Staten Island, NY 10314

Duke University
Department of Computer Science

The Department of Computer Science at Duke University invites applications and nominations for faculty positions at all levels, to begin August 2009. We are interested in strong candidates in all active research areas of computer science, both core and interdisciplinary areas, including algorithms, artificial intelligence, computational biology, computational economics, computer architecture, computer vision, database systems, distributed systems, machine learning, networking, and security.

The department is committed to increasing the diversity of its faculty, and we strongly encourage applications from women and minority candidates.

A successful candidate must have a solid disciplinary foundation and demonstrate promise of outstanding scholarship in every respect, including research and teaching. Please refer to www.cs.duke.edu for information about the department.

Applications should be submitted online at www.cs.duke.edu/facsearch. A Ph.D. in computer science or related area is required. To guarantee full consideration, applications and letters of reference should be received by January 4, 2009.

Durham, Chapel Hill, and the Research Triangle of North Carolina are vibrant, diverse, and thriving communities, frequently ranked among the best places in the country to live and work. Duke and the many other universities in the area offer a wealth of education and employment opportunities for spouses and families.

Emporia State University
Department of Mathematics, Computer Science and Economics
Tenure-track Assistant Professor

Emporia State University, Department of Mathematics, Computer Science and Economics seeks applications for a tenure-track assistant professor position beginning August, 2009. A Ph.D. in computer science or a closely related field is required. For complete information concerning the position see http://www.emporia.edu/esu/view-position.php?job_id=353.

Applications and recommendation letters may be sent to csjobsearch@emporia.edu. Applications received by December 15, 2008, will receive full consideration. Background checks are required. An AA/EOE institution, Emporia State University encourages women and minorities to apply.

Fairfield University
Department of Mathematics and Computer Science Faculty Position

The Department of Mathematics and Computer Science at Fairfield University invites applica-

Faculty Positions, All Levels

Several faculty positions are available at Cornell's department of Computer Science. Candidates are invited to apply from any area of computer science and at all levels including tenured, tenure track, or lecturer. We are especially interested in programming languages, scientific computing, computational biology, networking, and machine learning. However, we are open to hiring in all other areas as well, including artificial intelligence, databases, game design, graphics, robotics, security, systems, and theory of computation.

To ensure full consideration, applications should be received by January 15, 2009, but will be accepted until all positions are filled.

Applicants should submit a curriculum vita, brief statements of research and teaching interests through the web at **http://www.cis.cornell.edu/apply**, and arrange to have at least three references either uploaded on the Web or sent to: **Faculty Recruiting Committee Chair, Department of Computer Science, 4130 Upson Hall, Cornell University, Ithaca, NY 14853-7501 or frecruit@cs.cornell.edu**

Cornell University, located in Ithaca, New York, is an inclusive, dynamic, and innovative Ivy League university and New York's land-grant institution. Its staff, faculty, and students impart an uncommon sense of larger purpose and contribute creative ideas and best practices to further the university's mission of teaching, research, and outreach.

Cornell University
Cornell University is an Equal Opportunity Employer and encourages applications from women and ethnic minorities.

http://chronicle.com/jobs/profiles/2377.htm

tions for one tenure track position in computer science, at the rank of assistant professor, to begin in September 2009. We seek a highly qualified candidate with demonstrated excellence in and enthusiasm for teaching, a desire to contribute to the culture and development of a small program, and evidence of research potential. A doctorate in computer science is required. The teaching load is 3 courses/9 credit hours per semester.

The successful candidate will have a strong background in software design/languages and will be expected to teach a wide variety of courses including: Introduction to Computer Science, Data Structures, Software Design, Theory of Programming Languages, and Compiler Design.

Fairfield University, founded by the Jesuits, is a comprehensive university with about 3,200 undergraduates and a strong emphasis on liberal arts education. The department has an active faculty of 14 full-time tenured or tenure track members. We offer a BS in computer science as well as a BS and an MS in mathematics.

Fairfield offers competitive salaries and compensation benefits. The picturesque campus is located on Long Island Sound in southwestern Connecticut, about 50 miles from New York City. Fairfield is an Affirmative Action/Equal Opportunity Employer. For more information see http://fairfield.edu/macs_index.html.

Applicants should send a cover letter, a curriculum vitae, teaching and research statements, and three letters of recommendation commenting on the applicant's experience and promise as a teacher and scholar, to Dr. Matt Coleman, Chair of the Department of Mathematics and Computer Science, Fairfield University, 1073 N. Benson Rd., Fairfield CT 06824-5195. Full consideration will be given to complete applications received by January 9, 2009.

Florida International University
School of Computing and Information Sciences

Applications are invited for one or more tenure-track or tenured faculty positions at the levels of Assistant, Associate, or Full Professor. A Ph.D. in Computer Science or related areas is required. Outstanding candidates are sought in the areas of (1) Computer Security, (2) Bio/Medical/Health Informatics, and (3) Data Integration/Search/ Visualization. Exceptional candidates in other areas will be considered as well. Candidates for senior positions must have a proven record of excellence in research funding, publications, teaching, and professional service, as well as demonstrated ability for developing and leading collaborative research projects. Outstanding candidates for the senior positions will be considered for the **endowed Ryder Professorship** position. Successful candidates are expected to develop a high-quality funded research program and must be committed to excellence in teaching at both graduate and undergraduate levels.

Florida International University (FIU), the state university of Florida in Miami, is ranked by the Carnegie Foundation as a comprehensive doctoral research university with high research activity. FIU offers over 200 baccalaureate, masters and doctoral degree programs in 21 colleges and schools. With over 38,000 students, it is one of the 25 largest universities in the United States, and boasts a new and accredited Law School and the newly created College of Medicine.

The School of Computing and Information Sciences (SCIS) is a rapidly growing program of excellence at the University. It has 30 faculty members (including nine new faculty members hired in the last five years), 1,200 students including over 70 Ph.D. students, and offers B.S., M.S., and Ph.D. degrees in Computer Science and B.S. and B.A. degrees in Information Technology. The School is one of the largest computing programs in the eleven-university Florida State University System (SUS) and represents 20% of all degrees awarded by the system in the area. SCIS is the largest producer of Hispanic CS and IT graduates in the US. The School has tripled its external research funding in the last six years with $3.8 million in 2008-09 (to date), and is now consistently one of the top two funded research programs in the SUS, in terms of per-faculty annual external funding. The SCIS faculty leads several major NSF sponsored initiatives, including Center of Research Excellence in Science & Technology (CREST), Partnership for International Research & Education (PIRE), Industry-University Corporative Research Center (I/UCRC), Major Research Instruments (MRI), Research Experience for Undergraduates (REU), and holds four CAREER Awards from NSF and DOE. The School boosts six research centers or clusters and a first class computing infrastructure and support system. Furthermore, the School enjoys broad and dynamic industry and international partnerships.

Applications, including a letter of interest, contact information, curriculum vitae, and the names of at least three references, should be sent to Chair

---

Dunedin, New Zealand

Associate Professor/Professor

Department of Information Science
School of Business

The University of Otago is seeking to appoint an outstanding individual to provide strategic leadership in research and teaching initiatives for the Department of Information Science. This is an excellent opportunity for an internationally recognised academic to make an exemplary contribution to research, teaching, and the supervision of postgraduate students, in the first university outside the United States to be selected as a Sun Microsystems OpenSPARC Centre of Excellence.

The Information Science Department is part of the School of Business and has unique qualities. The Department places an emphasis on the way information is used in the organisational and social environments. The curriculum is primarily concerned with the way technology is designed, developed, and managed to produce, store, manipulate, and disseminate information, and includes aspects of software engineering and information technology. Please visit http://www.business.otago.ac.nz/infoscience/ for more information.

We are looking for an enthusiastic Associate Professor or Professor with the potential to lead the Department in the future. The candidate must be research active, able to support research projects, develop further research programmes, and build strong linkages with business.

The successful applicant will have excellent administrative skills, and the ability to nurture and develop our academic staff and students. There will also be a focus on curriculum development and teaching support as well as strengthening links with other departments.

The successful candidate will have a PhD, an extensive record of publications in high impact international journals, a sustained track record of attracting external funding, and a demonstrated ability to lead a research focused team.

Specific enquiries may be directed to Professor Martin Purvis, Acting Head of Department, Information Science, Tel 64 3 479 8796, Email mpurvis@infoscience.otago.ac.nz

Please visit www.otago.ac.nz/vacancies for more information about this vacancy and its closing date. If this vacancy is no longer posted on the website please visit www.otago.ac.nz/informationscience for an update about the position.

Applications should quote reference number A08/134.

APPLICATION INFORMATION

With each application you must include an application form, an EEO Information Statement, a covering letter, contact details for three referees and one copy of your full curriculum vitae. For an a**pplication form, EEO Information Statement and a full job description go to: www.otago.ac.nz/jobs** Alternatively, contact the H**uman Resources Division, Tel 64 3 479 8269, Fax 64 3 479 8279, Email** job.applications@otago.ac.nz

EEO Employers Group Equal opportunity in employment is University policy.

www.otago.ac.nz/jobs

of Recruitment Committee, School of Computing and Information Sciences, Florida International University, University Park, Miami, FL 33199. Online submission to *http://www. fiujobs.org* is preferred. The application review process will begin on January 15, 2009, and will continue until the positions are filled. Further information can be obtained from the School website *http://www.cis.fiu.edu*, or by e-mail to *recruit@cis.fiu.edu*.

Florida International University is a member of the State University System of Florida and is an equal opportunity/affirmative action/equal access employer.

Hiram College

Tenure-Track Faculty Position

Hiram College invites applications for a tenure-track faculty position. Applicants should love to teach the full spectrum of undergraduate computer science courses, and to mentor bright, demanding undergraduates in research and in the classroom. **A Ph.D. in computer science or related field is preferred. A Ph.D. must be completed by the start of the third year of a tenure-track position.** Although we anticipate filling this position at the Assistant Professor level, we encourage candidates with significant relevant experience beyond the Ph.D. to apply. Rank and salary commensurate with experience. A detailed job posting can be found at www.hiram.edu/visitors/employmentopportunities.html. CS Department information is available at cs.hiram.edu. Interested candidates should submit a single PDF file containing a curriculum vitae and statements of teaching and research philosophy to Dr. Ellen Walker ComputerScienceSearch@hiram.edu. Please have three letters of recommendation emailed directly, at least one of which addresses teaching. For more information, e-mail walkerel@hiram.edu. Application review will begin immediately and will continue until the position is filled. EOE.

Iowa State University

Assistant or Associate or Full Professor

The Electrical and Computer Engineering Department at Iowa State University has immediate openings for faculty positions at all levels. Applications will be accepted from highly qualified individuals for regular faculty positions in the department in all core areas of expertise in Electrical or Computer Engineering, especially in:

- COMPUTER ENGINEERING with emphasis on embedded systems;
- SOFTWARE ENGINEERING;
- ASSURANCE AND SECURITY; and
- POWER and energy/power electronics.

Exceptional senior candidates in any area may be considered for endowed research chair/professorship positions. Faculty positions also are available in interdisciplinary research areas as part of Iowa State University College of Engineering's aggressive mission to fill 50 college-wide positions with faculty who possess the talent to address the challenges that define worldwide quality of life and have global impact. The positions are targeted in the following interdisciplinary research and education cluster areas:

- Biosciences and Engineering
- Energy Sciences and Technology
- Engineering for Extreme Events
- Information and Decision Sciences
- Engineering for Sustainability

Duties for all positions will include undergraduate and graduate education, developing and sustaining externally-funded research, graduate student supervision and mentoring, and professional/institutional service.

REQUIREMENTS: All candidates must have an earned Ph.D. degree in Electrical Engineering, Computer Engineering, Computer Science, or related field, and they must have potential to excel in the classroom and to establish and maintain a productive externally funded research program. Associate and Full Professor candidates must, in addition, have an excellent record of externally funded research and internationally recognized scholarship.

Rank and salary are commensurate with qualifications. Screening will begin on November 1, 2008, and will continue until positions are filled. To guarantee consideration, complete applications must be received by January 19, 2009.

For regular faculty positions, apply online at http://www.iastatejobs.com/, Vacancy #080579.

More information on this position can be viewed at: http://www.ece.iastate.edu/jobs.

For information on positions in the cluster areas and application process, visit http://www.engineering.iastate.edu/clusters.

Candidates may be subject to a background check.

ISU is an EO/AA employee.

MICHIGAN STATE UNIVERSITY

TENURE-STREAM FACULTY DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

The Department of Computer Science and Engineering (CSE) at Michigan State University invites applications for a tenure-stream faculty position. Candidates in the area of Computer Vision are especially encouraged, but exceptional candidates in other areas will also be considered. Candidates at all ranks will be considered. The appointment starts in August 2009.

The CSE Department conducts leading-edge research in many areas, with particular strength in software engineering and formal methods; computer systems and networking; and pattern recognition and machine intelligence. Multidisciplinary research across a broad range of disciplines is strongly encouraged and is being actively pursued by the faculty. The Department presently has 27 faculty members and administers BS, MS and Ph.D. programs. MSU enjoys a large, park-like campus with outlying research facilities and natural areas. The greater Lansing area has approximately 450,000 residents. The local communities have excellent school systems and place a high value on education. The University is proactive in exploring opportunities for the employment of spouses, both inside and outside the University.

Candidates should submit a cover letter, curriculum vitae, the names of three references, and statements of research and teaching interests to: **www.cse.msu.edu/facultysearch**. Questions should be addressed to the email address given below. Applications will be reviewed on a continuing basis until the position is filled. For full consideration, applications should be received before January 9, 2009.

Faculty Search Committee, Department of Computer Science and Engineering, 3115 Engineering Building, Michigan State University, East Lansing, Michigan 48824-1226, search@cse.msu.edu.

MSU is committed to achieving excellence through cultural diversity. The University actively encourages applications and/or nominations of women, persons of color, veterans and persons with disabilities.

MSU IS AN AFFIRMATIVE ACTION, EQUAL OPPORTUNITY EMPLOYER.

KU THE UNIVERSITY OF KANSAS

Tenure-Track Assistant Professor

The University of Kansas Electrical Engineering and Computer Science Department seeks to hire one tenure-track assistant professor to support its computer systems design research area. We are seeking to fill research and teaching needs in digital system design, multicore architectures, systems-on-chip, behavioral synthesis, FPGA synthesis, and high-assurance systems. The success-ful applicant will have an earned PhD in computer science, computer engineering, or related area. Faculty members in our department are expected to develop nationally recognized research programs while supporting our undergraduate and graduate teaching missions. The appointment will be effective as negotiated. A complete application includes a letter of application, curricu-lum vita, and the names and addresses of three references. To submit an application, visit **http://www.eecs.ku.edu/recruitment/**. Applications will be reviewed beginning January 15, 2009 and will be accepted until the position is filled. EO/AA

EO/AA employer.

John Jay College of Criminal Justice, of the City University of New York,
Department of Mathematics and Computer Science
Assistant/Associate Professor

The Department of Mathematics and Computer Science, at the John Jay College of Criminal Justice, of the City University of New York, invites applications for tenure-track positions at the assistant or associate levels to begin Fall 2009. Successful candidates are expected to conduct both research and teaching for the Department's Computer Information Systems major and a new interdisciplinary M.S. degree program in Forensics Computing. The positions require a Ph.D. in Computer Science or Mathematics. Candidates from the areas of Computer Security, Information Assurance, Cryptography, Databases, Distributed Systems, Software Engineering and Compilers are especially encouraged, but exceptional candidates in other areas will also be considered. Responsibilities of the positions also include conducting research that endeavors to secure external funding and a commitment to excellence in teaching at all levels.

Applicants should submit a cover letter indicating the desired position, a Curriculum Vita, a Research Statement, a Teaching Statement, three recent publications, and three references to:

Prof. Samuel Graff, Co-Director of Forensics
Computing Master's Program,
& Computer Science Department,
John Jay College of Criminal Justice,
445 West 59th Street,
New York, NY 10019,

E-mail: pshenkin@jjay.cuny.edu.

The City University of New York is an Equal Employment/Affirmative Action/Immigration Reform and Control Act/Americans with Disabilities Employer.

Kansas State University
Department of Computing and Information Sciences

The Department of Computing and Information Sciences seeks a dynamic individual for its department head. The successful candidate will be a senior scholar/researcher who possesses a Ph.D. degree in Computer Science or in a closely related field, has excellent management abilities, and possesses effective interpersonal skills. The new head is anticipated to be tenured at full professor; candidates from all areas of computing are invited to apply.

The new head will administer teaching, research, scholarship, outreach, and service activities in a diverse and dynamic department consisting of 17 tenure-track faculty and 3 instructors, who lead research programs in bioinformatics and data mining, embedded and distributed systems, programming languages, security, and software engineering. Information about the department can be found at www.cis.ksu.edu.

To apply, email a cover letter, vita, and contact information for four references to recruiting@cis.ksu.edu or mail hard copy to:

David Schmidt, Chair, Department Head
Search Committee
Computing and Information Sciences Dept.
234 Nichols Hall, Kansas State University
Manhattan, KS 66506
(telephone: 785-532-7912)

Electronic submission (pdf, ps, doc, or txt) is preferred. Review of applications commences January 1, 2009 and continues until the position is filled.

KANSAS STATE UNIVERSITY IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER. QUALIFIED WOMEN AND MINORITIES ARE ENCOURAGED TO APPLY. Paid for by Kansas State University.

Lehigh University
Assistant Professor

Applications are invited for a tenure-track position at the Assistant Professor level in the Department of Computer Science and Engineering at Lehigh University starting in August 2009.

The potential to establish a competitive research program and teach effectively at both the undergraduate and graduate level are essential prerequisites. The successful applicant will hold a Ph.D. in Computer Science, Computer Engineering, or a closely related field. Outstanding candidates in all areas of computer science will be considered, with priority given to candidates whose research involves the exploitation of parallel-processor architectures for solving high-performance computing problems in areas such as bioinformatics, environmental and atmospheric modeling, analysis of complex systems, real-time sensor processing, and computer vision. Applicants should have an interest in teaching core courses in computer architecture as well as courses in their research area.

The Department includes IEEE and ACM fellows, and three NSF CAREER-award winners. We offer B.A., B.S., M.S., and Ph.D. degrees in Computer Science and jointly oversee B.S., M.S., and Ph.D. degree programs in Computer Engineering with our sister department, Electrical and Computer Engineering. We also offer a B.S. in Computer Science and Business jointly with the College of Business and Economics. Our involvement in multidisciplinary programs often results in opportunities for collaboration in cross-disciplinary sponsored research projects, including efforts in bioengineering, mobile robotics, sensor networks, and scientific computing.

Lehigh is a private institution ranked among the top 40 national research universities by U.S. News & World Report and is rated "most selective" by both Barron's and Peterson's guides. With a beautiful sylvan setting in Bethlehem, Pennsylvania, 80 miles west of New York City and 50 miles north of Philadelphia, Lehigh is conveniently located and offers an appealing mix of urban and rural lifestyles.

Applications can be submitted online at http://www.cse.lehigh.edu/faculty-search, and should include a cover letter, vita, and both teaching and research statements. In addition, please provide the names and email addresses of at least three references. Applications will be evaluated until the position is filled, but materials should be received by January 15, 2009 for full consideration. Lehigh University is an affirmative action and equal employment opportunity employer, and is committed to recruiting and retaining women and minorities. Questions concerning this search may be sent to faculty-search@cse.lehigh.edu.

TECHNICAL TALENT with SPECIALIZED EXPERTISE

Since 1960, The Aerospace Corporation has operated a federally funded research and development center in support of national security, civil and commercial space programs. We're applying the leading technologies and the brightest minds in the industry to meet the challenges of space. The Computers and Software Division (CSD) spans the entire spectrum of computer-related disciplines, supporting all phases of research, development, and acquisition for space, launch and ground computer systems. CSD also maintains state-of-the-art computer laboratories, based at both the Los Angeles, CA, and Washington, DC area locations. We offer competitive compensation and exceptional benefits.

SOFTWARE ENGINEERS COMPUTER SCIENTISTS

Individuals will provide technical expertise and support to government program offices for the design, evaluation, and development of complex, software-intensive space systems. Responsibilities include analyzing requirements; evaluating software architectures, designs, and implementations; evaluating development methodologies; and conducting technical studies, prototyping activities and applied research. Some technical areas of particular interest include real-time embedded systems (e.g., spacecraft flight software), software architecture, information and data modeling, high-performance data processing, net-centric computing, SOA and web-based applications, computer security, and system and information assurance, but applications are welcome from all computer-related disciplines.

Qualifications include a degree in Computer Science, Computer Engineering, Information Sciences, or related technical field (advanced degree preferred); good written and oral communications skills; and U.S. citizenship. Also, must be able to obtain a U.S. security clearance appropriate to the specific position.

Please apply online at www.aero.org/careers by submitting your resume. Please include **Job Code PJ3-0686** in your resume.

Equal Opportunity Employer.

AEROSPACE
Assuring Space Mission Success

Louisiana State University
Department of Computer Science
Assistant/Associate Professor
(Tenure-Track Facility Position in CyberSecurity/Internet Security/ High-Performance Computing)

The Department of Computer Science at Louisiana State University (http://www.csc.lsu.edu) seeks one candidate for tenure-track faculty position for Fall 2009. Through a targeted investment by the state, the University has chosen to establish a Center for Secure CyberSpace jointly with LaTech. We invite applications from outstanding candidates at all ranks with preference given to candidates at the Assistant Professor level.

The department provides excellent research opportunities for incoming faculty with the potential to join several existing funded interdisciplinary research programs along with major efforts such as the Louisiana Optical Network Initiative (LONI, http://www.loni.org). LONI, funded by a $40M commitment from the state provides a 40 Gbps connection between new largescale computing resources deployed at Louisiana Research Institutes. The infrastructure includes a statewide supercomputing grid of five 112-processor IBM p5-575 supercomputers, six 528-processor Dell PowerEdge servers and a 5,760 processor central server. These resources are connected by a 40 Gbps multi-lambda fiber-optic network which, in turn, is tied to the National Lambda Rail. LSU also has established the Center for Computation & Technology (www.cct.lsu.edu) to support high-performance computing research. The department has active research in the areas of cyber security and network security. Ideal Candidates should have expertise in one or more of the following fields: Internet and network security, security in sensor networks, Cryptographic methods, threats and vulnerabilities in cyberspace (e.g., phishing, spoofing, identity thefts etc), High Performance Computing that leverages any of these research areas.

Required Qualifications: Ph.D. in Computer Science, Electrical Engineering, Mathematics or a closely related field; excellent oral and written communication skills.

Additional Qualifications Desired: distinguished record of scholarship commensurate with experience; exceptional potential for world-class research; commitment to both undergraduate and graduate education. A commitment to high quality professional service and active participation in college responsibilities are expected.

The search committee will begin to review applications on January 15, 2009 and will continue to do so until the position is filled. An offer of employment is contingent on a satisfactory pre-employment background check. Salary and rank will be commensurate with qualifications and experience. Women and minorities are encouraged to apply. For consideration, please submit in electronic form the curriculum vitae (including e-mail address), statement of research and teaching interests, and the names and contact information for at least three references to: search1@csc.lsu.edu

LSU IS AN EQUAL OPPORTUNITY/
EQUAL ACCESS EMPLOYER

Louisiana State University
Department of Computer Science
Assistant/Associate Professor
(Tenure-Track Faculty Position in Software Engineering for Fall 2009)

The Department of Computer Science (http://www.csc.lsu.edu/) seeks a candidate for a tenure-track position as an Assistant Professor/Associate Professor for Fall 2009. The successful candidate will work toward furthering the Departments Research/Teaching in the areas of Software Engineering and Applications. The Department of Computer Science values research with impact and visibility beyond computer science. Faculty activities are expected to support and to be integrated into the complimentary goals of the Department of Computer Science. LSU is connected to Louisiana Optical Network Initiative (LONI), high speed network. LONI, funded by a $40M commitment from the state, will provide a 40 Gbps connection between new large scale computing resources deployed at Louisiana Research institutes. LSU is connected to the National Lambda Rail in Spring 2005, providing national and international high speed connectivity.

Required Qualifications: Ph.D. in Computer Science/Engineering or a related field. An offer of employment is contingent on a satisfactory pre-employment background check. Application reviews will begin January 15, 2009, and will continue until candidate is selected. Minorities and women are encouraged to apply for this position. Applicants should submit a letter of intent describing teaching and research interests along

---

Carnegie Mellon | isr institute for SOFTWARE RESEARCH

Carnegie Mellon University Institute for Software Research Master of Software Engineering Programs Teaching Track Position

The Professional Masters Programs in Software Engineering (www.mse.cs.cmu.edu) in the School of Computer Science at Carnegie Mellon University invites applications for a teaching-track position beginning in the fall term of 2009, or earlier. This is a career-oriented, renewable appointment for teaching courses in software engineering at the professional master's level.

Applicants for the position must have an M.S. or Ph.D., preferably in computer science, software engineering, or a related field, demonstrated excellence in teaching software engineering courses, and several years of professional experience in software engineering. Teaching-track appointments are typically at the rank of Assistant Teaching Professor, with the possibility of promotion to the ranks of Associate Teaching Professor and Full Teaching Professor. Teaching-track ranks are not tenured.

Preferred qualifications include broad industry experience, teaching experience, and familiarity with current software engineering technology and methods. We are particularly interested in candidates with experience in the areas of distributed systems, embedded systems, and systems engineering. Ideally the candidate should also have familiarity with a number of roles in the software engineering field, such as team lead, developer, manager, tester, and quality assurance manager.

Applications should include a letter of application, curriculum vitae, a statement of teaching philosophy, and letters of reference from three or more individuals. Applicants should arrange for reference letters to be sent directly to the contact below.

Email: Jane Dixon Miller <mil3+@cs.cmu.edu>
Paper: Master of Software Engineering Programs
Faculty Search Committee
Institute for Software Research
Carnegie Mellon University
300 S. Craig Street, Room 273
Pittsburgh, PA 15213-3707

Information Science
Tenure-track Faculty Positions, All Levels (Ref: ISF#2)

Cornell's Information Science Program (**www.infosci.cornell.edu**) is seeking to fill one or more tenure-track faculty positions. Information Science at Cornell is an interdisciplinary program within the Faculty of Computing and Information Science (**www.cis.cornell.edu**). It brings together those interested in studying information systems in their social, cultural, economic, historical, legal, and political contexts.

Candidates in all information-related fields are invited to apply. The ideal applicant will have a strong mix of both technical and social science research skills and accomplishments. We are especially interested in the areas of policy, economy, network science, cognition, e-science, cyber infrastructure and human-centered information retrieval, but will also consider candidates in other areas of Information Science.

To ensure full consideration, applications should be received by January 15, 2009, but will be accepted until all positions are filled.

The application package should include a curriculum vita, brief statements of research and teaching interests, and at least three letters of reference. Materials can be uploaded at the following URL: **www.infosci.cornell.edu/apply**

Cornell University, located in Ithaca, New York, is an inclusive, dynamic, and innovative Ivy League university and New York's land-grant institution. Its staff, faculty, and students impart an uncommon sense of larger purpose and contribute creative ideas and best practices to further the university's mission of teaching, research, and outreach.

Cornell University
Cornell University is an Equal Opportunity Employer and encourages applications from women and ethnic minorities.

http://chronicle.com/jobs/profiles/2377.htm

with the curriculum vita (including e-mail address) and the names and addresses of three references to: search2@csc.lsu.edu.

LSU IS AN EQUAL OPPORTUNITY/EQUAL ACCESS EMPLOYER

Loyola Marymount University
Presidential Professorship in Computational Biology

The Frank R. Seaver College of Science and Engineering at Loyola Marymount University (LMU) seeks candidates for a Presidential Professorship in Computational Biology. Candidates must have a distinguished record in teaching and research and a clear vision for providing leadership in interdisciplinary educational and research programs in Computational Biology. The ideal candidate, one who shares the mission of the University and its commitment to building an intercultural community, will receive an appointment in Biology at the rank of Professor. Exceptional candidates at the rank of Associate Professor and joint appointments will also be considered. Our faculty's current and emerging research interests include gene regulatory networks, genome annotation, biological databases, systems biology modeling, quantitative ecology, microbial diversity, and open source software development. The individual we are seeking will broaden and complement our current interests and expertise, and will provide leadership in the undergraduate degree programs as well as the recruitment of additional faculty to strengthen crosscutting interactions among our departments. The successful candidate will be housed in a new science building that is currently under design. Requirements for the position include a Ph.D. in Computational Biology, Biology, or a relevant, related discipline. Applicants are requested to send a letter of application, curriculum vitae, vision statement for the position, and three letters of reference. Review of applicants will begin on February 2, 2009. Materials should be sent to: Computational Biology Search Committee, Department of Biology, Loyola Marymount University, 1 LMU Drive, MS 8220, Los Angeles, CA 90045-2659. For additional information, contact Dr. Jeff Sanny, jsanny@lmu.edu, 310-338-2823 or Dr. Philippa Drennan, pdrennan@lmu.edu, 310-338-7776. Loyola Marymount, a comprehensive university in the mainstream of American Catholic higher education, seeks professionally outstanding applicants who value its mission and share its commitment to academic excellence, the education of the whole person, and the building of a just society. LMU is an equal opportunity institution actively working to promote an intercultural learning community. Women and minorities are encouraged to apply. (Visit www.lmu.edu for more information.)

Maryville University
Director of Computer Science

Maryville University is accepting applications for a Director of Computer Science faculty position. The successful candidate will have demonstrated leadership skills, a background of practical experience and industry awareness, excellence in teaching undergraduates, and a clear vision of the opportunities and challenges in this rapidly changing field.

To apply and for more information: Please visit https://jobs.maryville.edu/. AA/EOE

Max Planck Institute for Softwaresystems
Tenure-track

Max Planck Institute for Software Systems (MPI-SWS) Tenure-track faculty openings Applications are invited for tenure-track and tenured faculty positions in all areas related to the design, analysis and engineering of software systems, including programming languages, formal methods, security, distributed, networked and embedded systems, databases and information systems, and human-computer interaction. A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups. Senior candidates must have demonstrated leadership abilities and recognized international stature.

MPI-SWS, founded in 2005, is part of a network of eighty Max Planck Institutes, Germany's premier basic research facilities. MPIs have an established record of world-class, foundational research in the fields of medicine, biology, chemistry, physics, technology and humanities. Since 1948, MPI researchers have won 17 Nobel prizes. The new MPI-SWS aspires to meet the highest standards of excellence and international recognition with its research in software systems.

Oregon State UNIVERSITY OSU

School of Electrical Engineering and Computer Science

The School of Electrical Engineering and Computer Science at Oregon State University invites applications for tenure-track positions in Computer Science. The School of EECS strongly encourages teamwork and collaboration within the School, and with other departments and universities. We are particularly interested in candidates who can collaborate with our Graphics/Visualization, End-User Software Engineering and Machine Learning groups. The following areas are strong possibilities for collaboration with these groups: Human Computer Interaction; Theoretical Computer Science.

Applicants should have an earned doctorate in Computer Science/Computer Engineering by the appointment start date and demonstrate a strong commitment to high-quality undergraduate and graduate teaching and the development of a vibrant research program.

OSU is one of only two American universities to hold the Land Grant, Sea Grant, Sun Grant, and Space Grant designation and is the only Oregon institution recognized for its "very high research activity" (RU/VH) by the Carnegie Foundation for the Advancement of Teaching. With a faculty of 45, the School of EECS enrolls 1300 undergraduate and 300 MS/PhD students.

For more information, including instructions for application, visit http://www.eecs.oregonstate.edu. OSU is and AA/EOE.

WPI Worcester Polytechnic Institute

Department of Computer Science

Tenure Track Faculty Position

The Computer Science Department anticipates hiring tenure-track faculty at the Assistant Professor level for the Fall of 2009. Areas of particular interest include Robotics, Theory, Security, Systems & Networking, and Learning Sciences. However, outstanding candidates at any rank and in any area will receive full consideration. Candidates should have a PhD in Computer Science or a closely related field, and the potential for excellence in research and teaching.

The Department grants BS, MS, and PhD degrees, and is among the largest at WPI, with 20 tenure-track faculty members and approximately 320 majors and 100 full-time MS and PhD students. We expect to maintain the exceptionally close-knit, collegial atmosphere we currently enjoy. We are supported by federal agencies (including NSF, NIH, NSA, ONR, DHS, and DARPA) and industry (including Sun, Intel, Cisco, and Google).

For more information, please refer to the CS web site at:

http://www.cs.wpi.edu/

Questions about the hiring process should be sent to **recruit@cs.wpi.edu**. Applications should be submitted through **http://www.wpi.edu/Admin/HR/**. Please include detailed research and teaching statements, vitae, and contact information for at least three references.

For full consideration, applications should be received by 01/16/2009.

To enrich education through diversity, WPI is an affirmative action, EOE

To this end, the institute offers a unique environment that combines the best aspects of a university department and a research laboratory:

a) Faculty receive generous base funding to build and lead a team of graduate students and post-docs. They have full academic freedom and publish their research results freely.

b) Faculty have the opportunity to supervise doctoral theses, teach graduate and undergraduate courses, and have the flexibility to incorporate teaching into their research agenda.

c) Faculty are provided with outstanding technical and administrative support facilities as well as internationally competitive compensation packages.

Funds have been committed to grow the institute to a strength of 17 tenured and tenure-track faculty, and about 100 doctoral and post-doctoral positions. Additional growth through outside funding is expected. We maintain an open, international and diverse work environment and seek applications from outstanding researchers regardless of national origin or citizenship. The working language is English; knowledge of the German language is not required for a successful career at the institute. The institute is located in Kaiserslautern and Saarbruecken, in the tri-border area of Germany, France and Luxembourg. The area offers a high standard of living, beautiful surroundings and easy access to major metropolitan areas in the center of Europe, as well as a stimulating, competitive and collaborative work environment. In immediate proximity are the MPI for Informatics, Saarland University, the Technical University of Kaiserslautern, the German Center for Artificial Intelligence (DFKI), and the Fraunhofer Institutes for Experimental Software Engineering and for Industrial Mathematics. Qualified candidates should apply online at http://www.mpi-sws.org/application. The review of applications will begin on January 12, 2009, and applicants are strongly encouraged to apply by that date; however, applications will continue to be accepted until February 27, 2009.

The Max Planck Society is committed to increasing the representation of minorities, women and individuals with physical disabilities in Computer Science. We particularly encourage such individuals to apply.

Mississippi State University

Professor and Head

Department of Computer Science and Engineering

Applications and nominations are being sought for Professor and Head of the Department of Computer Science and Engineering (www.cse.msstate.edu) at Mississippi State University. This is a 12-month, tenure-track position.

Part of the Bagley College of Engineering, the department has approximately 325 undergraduate majors, 70 graduate students, and 18 tenured and tenure-track faculty. The department offers undergraduate programs in Computer Science and Software Engineering, and jointly administers the undergraduate program in Computer Engineering with the Department of Electrical and Computer Engineering. At the graduate level, we offer M.S. and Ph.D. degrees in Computer Science and faculty also direct graduate students in Computational Engineering and Computer Engineering. Certificates in Software Engineering, Information Assurance, and Computational Biology are also available. Research expenditures total about $3 million dollars annually and the university as a whole is ranked 48th among 271 U.S. institutions in computer science expenditures. Research areas for the department are high-performance computing, artificial intelligence, graphics and visualization, computer security, and software engineering. Three current faculty members have received NSF CAREER awards. Our computer security area has been designated a National Center of Academic Excellence in Information Assurance Education (CAEIAE) by the National Security Agency (NSA).

Mississippi State University is a comprehensive land-grant institution with approximately 17,000 students and about 1,000 faculty members. The university is a leader in high performance computing, housing a supercomputer in the top 20 among U.S. universities. The university's main campus is located in Starkville, Mississippi, a vibrant community approximately 2 hours from Jackson MS, Birmingham AL, and Memphis TN.

The successful Professor and Head will provide:

- Vision and leadership for nationally recognized computing education and research programs
- Exceptional academic and administrative skills
- A strong commitment to faculty recruitment and development

Applicants must have a Ph.D. in computer science, software engineering, computer engineering, or a closely related field. The successful candidate must have earned national recognition

Computer Science at TTI-Chicago

Faculty Positions at All Levels

Toyota Technological Institute at Chicago (TTI-C) is a philanthropically endowed degree-granting institute for computer science located on the University of Chicago campus. The Institute is expected to soon reach a steady state of 12 traditional faculty (tenure and tenure track), and 18 limited-term faculty. Applications are being accepted in all areas, but we are particularly interested in:

- **Theoretical Computer Science**
- **Computer Vision**
- **Computational Linguistics**
- **Programming Languages**
- **Electronic Commerce**
- **Machine Learning**
- **Speech Processing**
- **Scientific Computing**
- **Computational Biology**

Positions are available at all ranks, and we have a large number of limited-term positions currently available.

For all positions, we require a Ph.D. Degree or Ph.D. candidacy with the degree conferred prior to date of hire. Submit your application electronically at:

http://ttic.uchicago.edu/facapp/

Toyota Technological Institute at Chicago is an Equal Opportunity Employer.

TOYOTA TECHNOLOGICAL INSTITUTE CHICAGO

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Professor of Computer Science / Informatics (Computer Systems)

ETH Zurich invites applications for a position in Computer Science / Informatics, in the field of Computer Systems (www.inf.ethz.ch), at the level of full professor or tenure-track assistant professor. The position is intended to complement the existing systems-oriented faculty at ETH Zurich and pursue a practical research focus in the software and hardware architecture of parallel computer systems.

We expect a well-established track record of research with impact and recognition at an international level, as evidenced by publications in top-tier conferences and journals. In addition, applicants will be expected to demonstrate good teaching ability at the Bachelors and Masters levels. Courses at ETH Zurich in the Bachelor's program may be taught in English or German, as determined by the instructor. Courses in the Master's program are taught in English.

ETH Zurich offers an excellent environment for research in Computer Science, including a generous startup package, annual operating budget, and base funding for research assistants; high quality infrastructure; and an enthusiastic and able student body in a location with a high quality of life.

In order to receive full consideration please submit your application together with a curriculum vitae, a list of publications, a statement of research and teaching interests, and the names of at least three referees to the **President of ETH Zurich, Ramistrasse 101, ETH Zurich, 8092 Zurich, Switzerland, no later than January 31, 2009.** For further information, candidates may contact the Head of the Department, Prof. J. Gutknecht (gutknecht@inf.ethz.ch). With a view toward increasing the number of female professors, ETH Zurich specifically encourages qualified female candidates to apply.

by a distinguished record of accomplishments in computer-science education and research. Demonstrated administrative experience is desired, as is teaching experience at both the undergraduate and graduate levels. The successful candidate must qualify for the rank of professor with tenure.

Please provide a letter of application outlining your experience and vision for this position, a curriculum vita, and names and contact information of at least three professional references. Application materials should be submitted online at http://www.jobs.msstate.edu/ .

Screening of candidates will begin January 15, 2009 and will continue until the position is filled. Mississippi State University is an AA/EOE institution.

National Taiwan University
Professor-Associate Professor-
Assistant Professor

The Department of Computer Science and Information Engineering, the Graduate Institute of Networking and Multimedia, and the Graduate Institute of Biomedical Electronics and Bioinformatics, all of National Taiwan University, have faculty openings at all ranks beginning in August 2008. Highly qualified candidates in all areas of computer science/engineering and bioinformatics are invited to apply. A Ph.D. or its equivalent is required. Applicants are expected to conduct outstanding research and be committed to teaching. Candidates should send a curriculum vitae, three letters of reference, and supporting materials before February 28, 2009, to Prof Kun-Mao Chao, Department of Computer Science and Information Engineering, National Taiwan University, No 1, Sec 4, Roosevelt Rd., Taipei 106, Taiwan.

Northeastern University, Boston, MA
College of Computer and Information Science

Invites applications for tenure-track faculty positions in computer science and information science, beginning in Fall 2009. Candidates will be considered at all levels and for all ranks. A PhD in computer science, information science, or a related discipline is required.

Computer Science
Candidates will be considered from all major disciplines of computer science. We particularly welcome candidates with a demonstrated potential to excel in collaborative research spanning multiple research areas. The College has particular strengths in programming languages and software engineering, network security and distributed computing, database management and information retrieval, artificial intelligence, and human-computer interaction.

Information Science
Applicants with strong research programs in all areas of information science or information systems will be considered. We particularly welcome candidates with research related to human-computer interaction, information retrieval, natural language dialog, information security, and medical or health care informatics. Areas of current faculty research in information science include human-computer interaction, speech and language processing, information retrieval, machine learning and ontologies. Current application areas include health care informatics, legal text-based systems, and assistive technologies.

The College maintains a strong research program with significant funding from the major federal research agencies and private industry. The College has a diverse full-time faculty of 26, with approximately 500 undergraduates, 275 Masters, and 55 Ph.D. students. In addition to degrees in computer science, the College offers an innovative bachelor's program in Information Science, and dual bachelor degree programs between computer and information science and the sciences, digital arts and business. The College also offers interdisciplinary master's programs in information assurance and health informatics.

Additional information and instructions for submitting application materials may be found at the following web site: http://www.ccs.neu.edu/hiring/. Screening of applications begins immediately and will continue until the search is completed.

Northeastern University is an Equal Opportunity/Affirmative Action Employer. We strongly encourage applications from women and minorities.

Northwestern University
Department of Electrical Engineering
and Computer Science
Faculty Opening in Computer Engineering
and Systems at All Professorial Ranks

The Department of Electrical Engineering and Computer Science at Northwestern University in-

---

University of Massachusetts
UMASS Boston®

Careers with **Mass Appeal**

Assistant Professor
Department of Computer Science

The Computer Science Department at the University of Massachusetts Boston invites applications for Fall 2009 for one full-time tenure-track Assistant Professor position in Bioinformatics. We offer a BS, an MS with an emphasis on software engineering, and a PhD in computer science. We seek to strengthen our research program significantly. Current faculty interests include biodiversity informatics, computer and human vision, data mining, databases, networks, software engineering, system modeling, and theoretical computer science.

Evidence of significant research potential and a PhD in computer science or a related area are required. We offer a competitive salary and a generous start-up package.

Send cover letter, curriculum vitae, statements about research and teaching, and the names and email addresses of three references to Search 11408 at search@cs.umb.edu. Our campus overlooks Boston harbor; our faculty and students enjoy professional life in a center of academia and the software industry. For more information, visit us at http://www.cs.umb.edu. Review of applications has begun and will continue until the position is filled.

UMass Boston is an Affirmative Action, Equal Opportunity, Title IX employer.

SCHOOL OF MINES 1874 COLORADO

Colorado School of Mines Engineering/Math & Computer Science
Assistant Professor – Computer Science or Engineering

Colorado School of Mines (which offers B.S., M.S., and Ph.D. degrees) invites applications for an anticipated tenure-track faculty position at the Assistant Professor level in Computer Science or Engineering, beginning August 2009. The hired faculty member will be hosted in either the Division of Engineering (Electrical specialty) or the Department of Mathematical and Computer Sciences (Computer Science specialty).

Applicants must have earned a Ph.D. degree in computer engineering, computer science, electrical engineering or a related discipline with a research focus in computer architecture, embedded systems, machine learning, reconfigurable computing, robotics, or human-computer interaction. More information about Colorado School of Mines and its programs can be found at http://www.mines.edu, http://engineering.mines.edu/ (Division of Engineering), and http://www.mines.edu/Academic/macs/ (Department of Math. and Computer Sciences).

For a complete job announcement and the procedures for applying, see the complete announcement at: http://www.is.mines.edu/hr/Faculty_Jobs.shtm

CSM is an EEO/AA employer.

vites applications for a tenure track faculty position in computer engineering and systems. Candidates at all levels will be considered. An earned Ph.D. in Computer Science, Computer Engineering, or a related field is required, as is demonstrated success within computer engineering research and/or experimental computer systems research. More specific areas of interest include embedded systems, parallel systems, high performance computing, distributed systems, operating systems, compilers, power aware systems and sensor networks. Successful candidates will be expected to carry out world class research, collaborate with other faculty, and teach effectively at the undergraduate and graduate levels. Compensation and start-up packages are negotiable and will be competitive.

Northwestern EECS consists of over 50 faculty members of national prominence whose interests span a wide range. The Computer Engineering and Systems division, with 17 faculty members, is likely to provide a particularly stimulating collaborative environment for successful candidates. Northwestern University is located in the Chicago area.

Applicants should send a curriculum vitae, statements of research and teaching interests, three representative papers, and the names of at least three references for junior applicants and five for senior applicants, by email to: ces-search@eecs.northwestern.edu or by regular mail to:

CES Faculty Search Committee
Department of Electrical Engineering and Computer Science
Northwestern University
2145 Sheridan Road
Evanston, IL 60208

To ensure full consideration, applications should be received by January 15, 2009. Preference will be given to early applications, and interviews may start early, but no offer will be made prior to March 2009. The selected applicant will begin in the position either in September, 2009 or January, 2010.

Further information about the hiring department and the University is available at http://www.eecs.northwestern.edu and http://www.northwestern.edu

Northwestern University is an Affirmative Action, Equal Opportunity Employer. Women and minorities are encouraged to apply. Hiring is contingent upon eligibility to work in the United States.

NEC Laboratories America, Inc.

Research Staff Member - Parallel Computing Architectures for Intelligent Systems

NEC Laboratories America, a premier research facility of NEC Corporation, has an opening in the Systems Architecture Department located in Princeton, NJ. We invite applications from exceptional candidates (senior-level or junior-level) for a research staff (RSM) position. Candidates must have a PhD in CS, CE, or EE, and a strong research record and excellent credentials in the international research community. The focus of the position is to design, evaluate and deliver parallel computing software solutions on domain specific architectures to realize high-performance embedded and server systems. Applicants must be able to propose and execute innovative research projects, including prototyping effort that lead to demonstrations in an industry environment. Application-oriented candidates with experience in implementations of algorithms in the areas of machine vision, machine learning or data mining will be considered as well.

For more information, please visit http://www.nec-labs.com/. Interested applicants should send their resume and research statement to recruit@nec-labs.com and reference "Systems Architecture" in the subject line.

EOE/AA/MFDV

Penn State Applied Research Laboratory

Post Doctoral Scholar

The Applied Research Laboratory (ARL) at The Pennsylvania State University is an integral part of one of the leading research universities in the nation and serves as a university center of excellence in Defense Science and Technologies, with a focus in defense missions, warfighters and related areas. ARL is currently searching for Post Doctoral candidates for their Information Science and Technology Division (IST). The IST Division conducts nationally competitive multi-disciplinary innovative research in information science and technology for conceptualizing, designing, analyzing, deploying, and efficiently operating distributed and intelligent dynamic systems for innovative applications in DoD, Government, business, and education.

Post Doctoral Scholar - Job #: B-28523: The successful candidate will conduct research in a

---

City University of Hong Kong is one of eight tertiary institutions funded by the Government of the Hong Kong Special Administrative Region through the University Grants Committee of Hong Kong. A young and dynamic institution, the University aspires to be internationally recognized as a leading university in the Asia-Pacific region through excellence in professional education and applied research. It has a growing international reputation, as evidenced by its surge up the rankings of the world's top 150 universities according to the Times Higher Education Supplement. The mission of the University is to nurture and develop the talents of students and to create applicable knowledge in order to support social and economic advancement. Currently, approximately 26,000 students are enrolled in over 180 programmes ranging from associate degrees to PhD. The medium of instruction is English.

The University invites applications and nominations for the following posts. Candidates with applied research achievements will receive very positive consideration. Relevant experience in business and industry will be a definite asset.

Professor/Associate Professor/Assistant Professor Department of Computer Science *[A/545/09]*

Duties: Teach undergraduate and postgraduate courses, supervise research students, conduct research and perform duties as assigned by the Head of Department.

Requirements: A PhD degree in Computer Science or related disciplines. Candidates should have excellent proven record of research and teaching in one or more of the following areas: Multimedia/Software Engineering/Computer Networks/Databases/Internet Technologies/Information Security/Computer Systems/Algorithms/Computational Biology.

Salary and Conditions of Service
Salary offered will be highly competitive and commensurate with the candidate's qualifications and experience. Appointment will be on a fixed-term gratuity-bearing contract. Fringe benefits include annual leave, medical and dental schemes, and housing benefits where applicable.

Information and Application
Information concerning the posts and the University is available at http://www.cityu.edu.hk or from the Human Resources Office, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong [Fax : (852) 2788 1154 or (852) 2788 9334/email : hrojob@cityu.edu.hk]. Please send nomination or application in the form of an application letter enclosing a current CV to the Human Resources Office. **Applications will be considered until the positions are filled**. Please quote the reference of the post in the application and on the envelope. The University reserves the right to consider late applications and nominations, and to fill or not to fill the positions. Personal data provided by applicants will be used for recruitment and other employment-related purposes.

broad area of applying sensing technologies to tag, track and locate targets. Requirements include a Ph.D. in Computer Science/Computer Engineering, Electrical Engineering or Mathematics. The preferred candidate should have knowledge in one or more of the following areas: sensor networks, biosensors, image understanding, stochastic modeling, target tagging, sensor data fusion, dynamic control and tracking and location estimation.

Post Doctoral Fellow - Job #: B-23831: The successful candidate will conduct research in the general area of sensor networks. Requirements include a Ph.D. in Computer Science/Information Systems, Electrical Engineering, or Math/Statistics. The successful candidate should have knowledge in one or more of the following areas: sensor networks, mobile ad hoc networks, sensor data fusion, and dynamic control. Experience in networks control engineering, information processing and systems, and computer security will be highly desirable.

These are one year, fixed-term renewable appointments. Candidates who are within three to six months of completing a Ph.D. or have recently completed a Ph.D. within the past five years are encouraged to apply. YOU MUST BE A U.S. CITIZEN TO APPLY.

Applicants electronically apply at arl-jobs@psu.edu. PLEASE INSERT JOB NUMBER IN SUBJECT LINE. Applications will be accepted until position is filled. FOR FURTHER INFORMATION about the Applied Research Laboratory, visit our web site at www.arl.psu.edu.

Penn State is committed to affirmative action, equal opportunity and the diversity of its workforce.

Purdue University
Department of Computer Science
Assistant Professor

The Department of Computer Science at Purdue University invites applications for tenure-track positions at the assistant professor level beginning August 2009. Outstanding candidates in all areas of Computer Science will be considered.

The Department of Computer Science offers a stimulating and nurturing academic environment. Forty-four faculty members direct research programs in analysis of algorithms, bioinformatics, databases, distributed and parallel computing, graphics and visualization, information security, machine learning, networking, programming languages and compilers, scientific computing, and software engineering. The department has implemented a strategic plan for future growth supported by the higher administration and recently moved into a new building. Further information about the department is available at http://www.cs.purdue.edu.

All applicants should hold a PhD in Computer Science, or a closely related discipline, be committed to excellence in teaching, and have demonstrated potential for excellence in research. Salary and benefits are highly competitive. Applicants should apply online at https://hiring.science.purdue.edu. Review of applications will begin on October 1, 2008, and will continue until the positions are filled. Purdue University is an Equal Opportunity/Equal Access/Affirmative Action employer fully committed to achieving a diverse workforce.

Purdue University Calumet
Assistant/Associate Professor of Computer Science

The Department of Mathematics, Computer Science, and Statistics at Purdue University Calumet, Hammond, Indiana is accepting applications for a tenure-track position in computer science at the rank of Assistant Professor starting August 2009. An exceptionally qualified applicant may be considered for appointment at the rank of Associate Professor.

Requirements: A Ph.D. in Computer Science or in a related field with the equivalent of a Master's degree in Computer Science is required. Candidates with backgrounds in computer graphics or distributed computing are encouraged to apply. A record of effective teaching and evidence of the ability to establish a program of research is expected.

Responsibilities: Job duties will include teaching graduate and undergraduate courses in computer science, conducting research in field of expertise, overseeing senior level and graduate level computer science projects, overseeing student research, participating in recruitment of computer science majors, and participating in department and university service.

Screening of applicants will begin January 14, 2009 and continue until a suitable candidate is found. Applicants should forward a letter of application, curriculum vita, statements of teaching philosophy and research plans, copies of undergraduate and graduate transcripts, and the names and contact information of three (3) references to:

Department of Mathematics, Computer Science, and Statistics
Professor Roger Kraft, Computer Science Faculty Search Chair
2200 169th Street
Hammond, IN 46323-2094

For department information please visit http://www.calumet.purdue.edu

Purdue University Calumet is an Equal Access, Equal Employment Opportunity, Affirmative Action employer that is committed to a diverse workplace.

Rochester Institute of Technology
Computer and information sciences
Faculty Openings for Fall 2009

Rochester Institute of Technology's B. Thomas Golisano College of Computing and Information Sciences (GCCIS) invites applications and nominations for the following positions:

- Two tenured/tenure-track faculty to support its interdisciplinary Ph.D. program in areas including, but not limited to, biomedical imaging and computing, computational modeling and simulation, scientific computing, and service sciences (IRC #26155)
- Two tenure-track Computer Science faculty, at the assistant professor rank, in the areas of data management, programming concepts and tools, and scientific/distributed computing (IRC #26157)
- A tenure-track faculty in Software Engineering in the areas of quality assurance, secure software systems, software engineering management, and software engineering process (IRC #26119)
- One, and possibly two, tenure-track faculty in Game Design & Development (IRC #26160)

Successful candidates must demonstrate excellence in teaching and scholarship and have the ability to contribute in meaningful ways to RIT's commitment to cultural diversity and pluralism. A terminal degree in the discipline, or closely related field, or equivalent experience is required.

The Golisano College of Computing and Information Sciences is home to the Computer Science, Information Technology, Software Engineering, and Networking, Security, and Systems Administration departments, as well as the PhD program in Computing and Information Sciences, the research arm of the college. The College is housed in 125,000 square feet of dedicated space with state-of-the-art facilities, adjacent to our 9,000 sq. ft. research center. The College currently has 102 full-time faculty members, over 2,200 undergraduate students and over 550 Master's level graduate students.

Candidates should visit https://mycareer.rit.edu and refer to the IRC number listed above for specific information about the positions and the application process. Refer to www.rit.edu for information about RIT and the B. Thomas Golisano College of Computing and Information Sciences.

Slippery Rock University of Pennsylvania
Information Systems

Slippery Rock University is seeking applicants for a full time, tenure track position in the Computer Science Department beginning August, 2009.

Responsibilities include teaching courses in the Information Systems (IS) major as well as other departmental offerings, academic advisement, involvement in scholarly and professional activities, and participation in various department and university service activities.

Completion of an earned doctorate in Information Systems (IS) is required by August 24, 2009. Successful performance in an on-campus interview, including a teaching session is required. The candidate must demonstrate a commitment to excellence in teaching, to the education of diverse populations, and to teaching a wide range of undergraduate courses. Teaching and industry experience in the area of Information Systems are preferred.

Send letter of interest, resume, graduate and undergraduate transcripts (official transcripts required before hiring) and the names, addresses and phone numbers of three references to:

Dr. Michael P. Conlon
Search Committee Chair
Computer Science Department
275 Advanced Technology and Science Hall
Slippery Rock University
Slippery Rock, PA 16057
Fax: 724-738-4513
Phone: 724-738-2040
Email: michael.conlon@sru.edu

Review of applications will begin December 15, 2008, and continue until the position is filled.

Background investigation required for employment. Slippery Rock University of PA is a member of the State System of Higher Education and is an affirmative action/equal opportunity employer. TTY# 724-738-4881. Visit the university Web site at www.sru.edu, and the Computer Science Department web site at cs.sru.edu.

ing a new group of entering undergraduate students each semester. The ITCS department has expanded the curriculum and is seeking a wide range of new faculty members for the fall semester of 2009. The successful applicant will demonstrate a commitment to undergraduate teaching and the ability to engage undergraduates in research and class projects. Teaching responsibilities include participation in our introductory courses as well as upper level courses in the applicant areas of interest and expertise, including but not limited to computer networks, databases, eCommerce & web development, programming languages, or operating systems. The teaching load per semester is 3 to 4 courses with no more than three course preparations. A Ph.D. or M.Sc. in a computer related area is required for this position. Salary and level of appointment commensurate with experience. Please see our website for descriptions of the faculty position and how to apply: www.auaf.edu.af

The College at Brockport

State University of New York

Applications are invited for two tenure-track Assistant Professor positions in the Computer Science Department beginning Fall 2009. Doctoral degree in Information Systems, or doctoral degree in a closely related discipline with master's level expertise in Information Systems, is required.

All specialties will be considered. Preference will be given to Information Systems, Systems Analysis, IT Project Management, Databases, Data Mining, Computer Networks, E-Commerce, Security, Multimedia, and Computer-Human Interaction. Candidates who can serve as mentors for women and minority students under-represented in the discipline of computing also will be preferred.

Apply online at http://www.brockportrecruit.org/ by January 12, 2009. For more information visit www.brockport.edu/cs/vacancy.html

AA/EOE

The Ohio State University

Assistant Professor

The Department of Computer Science and Engineering (CSE), The Ohio State University, invites applications for two tenure-track positions at the Assistant Professor level. The positions are open to all CSE areas (artificial intelligence, graphics and animation, networking, software engineering and programming languages, systems, and theory) with priority consideration given to candidates with cross-cutting interests in database systems and machine learning (e.g. data mining) and those in theory (broadly defined).

The department is committed to enhancing faculty diversity; women, minorities, or individuals with disabilities are especially encouraged to apply.

Applicants should hold or be completing a Ph.D. in CSE or a closely related field, and have a commitment to and demonstrated record of excellence in research as well as a commitment to excellence in teaching.

The department maintains and encourages multi-disciplinary research and education activities within and outside The Ohio State University.

To apply, please submit your application via the online database. The link can be found at: http://www.cse.ohio-state.edu/department/positions.shtml

Review of applications will begin in January and will continue until the positions are filled.

The Ohio State University is an Equal Opportunity/Affirmative Action Employer.

Tufts University

Cognitive Science Full or Associate Professor

The Department of Computer Science at Tufts University invites applications for a faculty appointment at the Professor or Associate Professor level in Cognitive Science or related area to begin in September 2009. We seek outstanding candidates who can both build the Computer Science presence in the new Tufts Program in Cognitive and Brain Science and complement the Department's expertise in Machine Learning, Human-Computer Interaction, Graphics, and/or Computational Biology.

Tufts is among the smallest universities to have been nationally ranked as a "Research Class 1" University. Located in Boston, it has a dynamically growing computer Science Department. Located just three miles from Cambridge, MA, home to Harvard and MIT, Tufts faculty have opportunities for collaboration and participation in the rich intellectual life of the Boston area. For more information about the department, this position and instructions on how to apply, please visit http://www.cs.tufts.edu/people/careers.

Screening of applications will begin December 1, 2008 and continue until the position is filled. Tufts University is an Affirmative Action/Equal Employment Opportunity Employer. We are committed to increasing the diversity of our faculty, members of underrepresented groups are strongly encouraged to apply.

Tufts University

Graphics & Visualization
Full or Assoc. Professor
Computer Science 2009 Faculty Search in Graphics and/or Visualization

The Department of Computer Science at Tufts University invites applications for a faculty appointment at the Professor or Associate Professor level in Graphics and/or Visualization to begin in September 2009. We seek outstanding candidates who can both build the Computer Science presence in the new Tufts Center for Scientific Visualization and complement the Department's current research strengths.

Tufts is among the smallest universities to have been nationally ranked as a "Research Class 1" University. Located in Boston, it has a dynamically growing computer Science Department. Located just three miles from Cambridge, MA, home to Harvard and MIT, Tufts faculty have opportunities for collaboration and participation in the rich intellectual life of the Boston area. For more information about the department, this position and instructions on how to apply, please visit http://www.cs.tufts.edu/people/careers

Screening of applications will begin December 1, 2008 and continue until the position is filled. Tufts University is an Affirmative Action/Equal Employment Opportunity Employer. We are committed to increasing the diversity of our faculty. Member of underrepresented groups are strongly encouraged to apply.

University of Arkansas

Two Full-Time, Tenure-Track, Assistant Professor Positions

The Computer Science and Computer Engineering (CSCE) Department of the University of Arkansas seeks outstanding individuals to fill the following two full-time, tenure-track, assistant professor positions. Successful candidates should have an earned doctorate in computer science, computer engineering, or a related field. We are seeking candidates with expertise in software and hardware foundations for next generation data intensive computing systems.

Specific Computer Engineering areas include: Multiprocessor/Multicore and Custom Systems, Hardware/Software Co-design, Real-time Systems, Networks and Sensor Technology, and Wireless Systems.

Computer Science areas of interest include: Distributed Computing, Trusted Computing, Intelligent Systems, High Performance Scientific Computing, and Ubiquitous Computing. Outstanding candidates in related areas are also encouraged to apply.

The University of Arkansas is located in Fayetteville, ranked as the 8th "Best Metro" in Forbes' 2007 list of "Best Places for Business and Careers," and recognized as one of the "Top 10 Best Cities to Live, Work and Play" by Kiplinger in 2008. The CSCE department recently relocated to the new 90,000 sq ft. J.B. Hunt building and offers BS, MS, and Ph.D. degrees in Computer Engineering, and Computer Science. For more information concerning the department see http://www.csce.uark.edu/. Applications should be submitted electronically at http://www.csce.uark.edu/recruiting. Application review begins January 15, 2009 and will continue until positions are filled.

The University of Arkansas is an equal opportunity, affirmative action employer. Qualified underrepresented minority and women candidates are especially invited to apply. All applicants are subject to public disclosure under the Arkansas Freedom of Information Act and persons hired must have proof of legal authority to work in the United States.

University of California, Irvine

Tenured or Tenure-track Faculty Position in Systems Biology

The University of California, Irvine has embarked on a recruiting initiative in Systems Biology intended to fill seven faculty positions over three years. One position is available this year, for which candidates will be considered from all areas of Systems Biology, including biological networks, regulatory dynamics and control, spatial dynamics and morphogenesis, synthetic biology, and mathematical and computational biology. Applications are being solicited at the Assistant, Associate and Full Professor level, and appointment can be made in any of several departments, including Developmental and Cell Biology, Molecular Biology and Biochemistry, Ecology and Evolutionary Biology, Biomedical Engineering, Mathematics, Physics and Astronomy, Computer Science, and Statistics.

The successful applicant is expected to conduct a strong research program and to contribute to the teaching of undergraduate and graduate students. Systems Biology research and training at UCI is fostered by several interdisciplinary research units, an NIGMS National Center for Systems Biology, and Ph.D. training programs in Bioinformatics, and Mathematical and Computational Biology (for more information, see *http://ccbs.bio.uci.edu*). Applicants should submit a letter of application, curriculum vitae, bibliography, three letters of reference, and statements of research and teaching interests using the on-line recruitment system (see instructions at *http://ccbs.bio.uci.edu* or *https://recruit.ap.uci.edu*). To receive full consideration, material should be received by December 10, 2008.

The University of California, Irvine is an equal opportunity employer committed to excellence through diversity, and strongly encourages applications from all qualified applicants, including women and minorities. UCI is responsive to the needs of dual career couples, is dedicated to work-life balance through an array of family-friendly policies, and is the recipient of an NSF ADVANCE Award for gender equity.

University of California, Los Angeles
Computer Science Department
Faculty Recruitment

The Computer Science Department of the Henry Samueli School of Engineering and Applied Science at the University of California, Los Angeles, invites applications for tenure-track positions in all areas of Computer Science and Computer Engineering, especially in machine learning, software systems, and emerging technologies related to computer science such as bio-computing, nano architectures, and nanosystems. Applications are also encouraged from distinguished candidates at senior levels. Quality is our key criterion for applicant selection. Applicants should have a strong commitment both to research and teaching and an outstanding record of research for their level of seniority.

The University of California is an Equal Opportunity/Affirmative Action Employer. The department is committed to building a more diverse faculty, staff and student body as it responds to the changing population and educational needs of California and the nation. To apply, please visit http://www.cs.ucla.edu/recruit. Faculty applications received by January 15 will be given full consideration.

University of California, Riverside
Faculty Position
Department of Computer Science and Engineering

The Department of Computer Science and Engineering invites applications for one faculty position at all levels with research interests in Embedded Systems and Computer Engineering. Exceptional candidates with research interests in Computer Graphics and Computational Science & Engineering are also invited to apply. Tenured and tenure-track positions require a Ph.D. in Computer Science (or in a closely related field) at the time of employment. Visit http://www.cs.ucr.edu for information on the department and http://www.engr.ucr.edu/facultysearch/ to apply. Review of applications will begin on January 1, 2009, and will continue until the position is filled. For inquiries contact search@cs.ucr.edu. EEO/AA employer.

University of California, Santa Barbara
Tenure-Track Assistant Professor
Media Arts and Technology Graduate Program

The Media Arts and Technology Program at the University of California, Santa Barbara, invites applications for a tenure-track position at the assistant professor level, starting July 1, 2009. The department seeks candidates who will establish a vigorous research and teaching program in computer graphics, scientific/information visualization, or a related field applicable to immersive, interactive, and distributed environments, working with high-dimensional data generated in scientific and artistic domains. The successful candidate will be expected to collaborate with artists, engineers, and scientists in an interdisciplinary environment of research, creative work, and teaching.

Media Arts and Technology (MAT) is a transdisciplinary graduate program at UCSB in both the College of Letters and Science (Division of Humanities and Fine Arts) and the College of Engineering. MAT offers Master's and PhD degrees and has approximately 40 graduate students and 10 faculty, several with joint appointments in engineering and arts departments. Areas of expertise include human-computer interaction, electronic music and sound design, computational visual and spatial arts, and multimedia signal processing. Offices and labs are housed in the new California Nanosystems Institute building at UCSB, which includes a unique research facility called the Allosphere, a three-story spherical immersive environment. Additional information about the department can be found at http://www.mat.ucsb.edu.

Applicants are expected to hold a doctoral degree in Media Arts and Sciences, Computer Science, or a closely related field, have demonstrated excellence in research, and have a strong commitment to teaching and interdisciplinary scholarship and/or creative activity.

The department is especially interested in candidates who can contribute to the diversity and excellence of the academic community through research, teaching, and service. Primary consideration will be given to applications received by December 15, 2008; however, the position will remain open until filled. Applications must include a CV, research and teaching statements, and at least three letters of reference. See http://www.mat.ucsb.edu/recruit for information on how to apply.

The University of California is an Equal Opportunity / Affirmative Action Employer.

University of California, Santa Cruz
Computer Science Department

The UCSC Computer Science Department seeks qualified applicants for the following tenure-track faculty position within the Baskin School of Engineering:

Assistant Professor Position: 034-09

Interests: Outstanding Applicants with Research Excellence in Software Engineering, especially those with research distinction in software analysis and automated software generation techniques.

The department has strong graduate and undergraduate programs. Research and instruction are supported by excellent computing facilities and state-of-the-art laboratories in the new Engineering 2 building. UCSC is close to Silicon Valley and has strong ties with many of the high technology companies in the area. Faculty salaries are competitive and opportunities for consulting are extensive.

A detailed job description and application instructions are available at www.soe.ucsc.edu/jobs/faculty/apply/

For full consideration, all materials should reference position number 034-09 and arrive by January 9, 2009. UCSC is an EEO/AA/IRCA Employer.

University of Central Arkansas (UCA)
Tenure-track Assistant/Associate Professor

The Computer Science Department at the University of Central Arkansas (UCA) invites applications to fill one tenure-track faculty position at the assistant/associate professor level to start in Fall 2009. Applicants should have a PhD degree in an area of computer science and/or computer engineering. All fields will be considered, however, applicants with specialty in service oriented systems, enterprise software systems and cloud computing systems will be given preferential consideration. A strong commitment to excellence in teaching and to research involving students is expected.

The department offers BS (accredited by ABET since October 2006) and MS degree programs and enjoys strong support from local corporations. UCA is the second largest university in Arkansas and is committed to excellence in undergraduate and graduate education. It has been ranked in the top tier of universities in the south since 2006. For more information about the department and the university, visit www.cs.uca.edu.

Applicants should send their curriculum vitae, statements of teaching and research interests, transcripts, and at least three letters of reference to: Chair, CS Search Committee, Computer Science Department, University of Central Arkansas, 201 Donaghey Avenue, Conway, AR 72035. Electronic submission may be sent to CSsearch@list.uca.edu. Review will begin on December 1, 2008 and continue until the position is filled.

UCA is an Equal Opportunity Affirmative Action Employer

University of Dayton
Assistant Professor in Computer Science

The Computer Science Department invites applications for a tenure track assistant-professor faculty position beginning August 15, 2009. Candidates must have a Ph.D. in Computer Science, excellent communication skills and a strong commitment to teaching, research and service. Candidates must be able to provide broad support for both the undergraduate and graduate programs of the department.

The Computer Science Department has a low student-faculty ratio with a strong commitment to maintaining engagement with our majors. We maintain a high degree of academic excellence and educate students to be servant leaders. Information about the department may be found at http://www.udayton.edu/~cps

Applications, including copies of undergraduate and graduate transcripts, statements of teaching interests and philosophy and research plans should be sent to: Faculty Search Committee, Department of Computer Science, University of Dayton, Dayton, OH 45469-2160. Applications can also be sent electronically to cps-faculty-search@udayton.edu. Candidates should arrange to have three confidential letters of reference sent directly to the above address. Review of applications will begin January 15, 2009 and continue until the positions are filled.

The University of Dayton, a comprehensive Catholic university founded by the Society of Mary (Marianists) in 1850, is an Affirmative Action/Equal Opportunity Employer. Women, minorities, individuals with disabilities, and veterans are strongly encouraged to apply. The University of Dayton is firmly committed to the principle of diversity.

University of Delaware
Department of Computer & Information Sciences
Teaching Faculty

The Department of Computer & Information Sciences at the University of Delaware invites applications for a teaching faculty position to begin September 1, 2009. This is a full-time, non-tenure track position with initial two-year contract, renewable in a program that leads to substantial job security.

The candidate's primary responsibility will be teaching (typically three to four courses per semester). The candidate will also participate in curriculum development, student advisement, and service on department committees. Applicants should be strongly committed to excellence in teaching and service. A Ph.D in computer science or a closely related discipline by the start date is expected, though exceptional candidates with masters degrees will be considered. Of particular interest are candidates who can teach Web technologies or software engineering. More information about the Department is available at http://www.cis.udel.edu.

Applications should be submitted as a SINGLE PDF FILE to csfacsch@cis.udel.edu and include: a cover letter (addressed to Dr. Christopher Rasmussen, Faculty Search Committee Chair), a vita with a list of references, and a statement on teaching. In addition, candidates should have three (3) confidential letters of reference sent directly to csfacsch@cis.udel.edu. All application materials may be shared with departmental faculty. The committee will give priority to applications received by January 1, 2009.

The University of Delaware is a premier coeducational institution of 15,000+ undergraduate students and 3000+ graduate students. With a distinguished faculty and strengths in study abroad, undergraduate research and discovery learning, the University offers a broad range of undergraduate and graduate degrees across seven colleges. The beautiful 100-acre central campus is located in Newark, DE, about halfway between New York City and Washington, DC.

UD is an equal opportunity employer which encourages applications from minority group members and women.

University of Houston-Clear Lake
Assistant Professor of Computer Science

Assistant Professor of Computer Science: University of Houston-Clear Lake invites applications for Assistant Professor of CS to begin August 2009. Web technologies research interest preferred. Apply at https://jobs.uhcl.edu/. AA/EOE.

University of Minnesota – Twin Cities
Department of Computer Science and Engineering
Faculty Position

The Department of Computer Science and Engineering at the University of Minnesota – Twin Cities invites applications for one tenure-track faculty position at the Assistant Professor level in the interdisciplinary area of Geo-spatial computer sciences, with a focus on spatial data mining, spatial database management, and related subareas. Outstanding candidates for a position at a more senior level will be considered. Women and other underrepresented groups are especially encouraged to apply. Requirements

include a Ph.D. in computer science or a closely related discipline, a commitment to quality teaching, and the ability to carry out outstanding, high-impact interdisciplinary research in the applications of Geo-spatial computer sciences in such areas as public health(e.g. environmental epidemiology), public safety (e.g. environmental criminology), ecology and environmental science, transportation science, etc. The position will remain open until filled, but for full consideration, apply by January 15th, 2009. See http://www.cs.umn.edu/resources/employment/faculty.php for details and for application requirements. The University of Minnesota is an equal opportunity employer and educator.

University of Nebraska-Lincoln
Assistant Professor

We invite applications for a tenure track faculty position at the rank of Assistant Professor. We are looking for a faculty member who can establish a strong research and teaching program that will strengthen our programs in the area of Human Computer Interfaces and/or Software Engineering. Candidates must hold an earned doctorate in Computer Science or a closely related discipline by the date of employment.

To apply, visit http://employment.unl.edu and complete a Faculty/Administrative application for requisition number 080713. Attach a cover letter, a CV, and statements describing your proposed research and teaching to your application. The cover letter must include names and contact information for at least three references. Review of applications will begin on December 1, 2008, and will continue until the position has been filled. A more detailed advertisement can be viewed at http://cse.unl.edu/search. The University of Nebraska is committed to a pluralistic campus community through affirmative action, equal opportunity, work-life balance, and dual careers.

University of North Carolina at Charlotte
Department of Software and Information Systems
Tenure-Track Faculty Positions

The Department of Software and Information Systems at UNC Charlotte invites applicants for multiple tenure-track faculty positions at both the assistant and associate levels. The Department is dedicated to research and education in Software Engineering and Information Technology applications, with emphasis in the areas of Information Integration & Environments and Information Security & Assurance; it offers degrees at the Bachelors, Masters, and Ph.D. levels. Current faculty members have strong research programs with substantial funding from both federal agencies and industrial partners. The department is particularly interested in faculty with research expertise in: Trusted Software Development, Software Engineering, or Modeling & Simulation. Highly qualified candidates in other areas will also be considered.

Salary will be highly competitive. Applicants must have a Ph.D. in Computer Science, Information Technology, Software Engineering, or a related field, as well as a strong commitment to research and education. For further details please visit http://www.sis.uncc.edu/. Application review will start in January 2009.

Applications must be submitted online at https://jobs.uncc.edu//. To the application, please attach a cover letter, curriculum vitae, a statement of teaching interests, a statement of research interests, copies of three representative scholarly publications, and a list of four references. For questions or additional information, please email search-sis@uncc.edu.

Women, minorities and individuals with disabilities are encouraged to apply. UNC Charlotte is an Equal Opportunity/Affirmative Action employer.

University of North Carolina at Charlotte
Department of Software and Information Systems
DICyDER Center Director

The Department of Software and Information Systems at UNC Charlotte seeks to hire a tenure-track faculty member at the associate level to serve as Director of the recently established Center for Digital Identity and Cyber Defense Research (DICyDER), http://www.dicyder.uncc.edu/. DICyDER's mission is to add value to the university, community, and society through innovative educational programs, research and development in the areas of information integration, security, and privacy. The Director will be responsible for leading a strong research program by communicating research vision, planning and implementing research strategy, facilitating contract acquisition and relationship development, and providing project and group management.

The Department of Software and Information Systems is dedicated to research and education in Software Engineering and Information Technology applications, with emphasis in the areas of Information Integration & Environments and Information Security & Assurance; it offers degrees at the Bachelors, Masters, and Ph.D. levels. Current faculty members have strong research programs with substantial funding from both federal agencies and industrial partners.

Salary will be highly competitive. Applicants must have a Ph.D. in Computer Science, Information Technology, Software Engineering, or a related field, as well as a strong commitment to research and education. For further details please visit http://www.sis.uncc.edu/. Application review will start in January 2009.

Applications must be submitted online at https://jobs.uncc.edu/. To the application, attach a cover letter, curriculum vitae, a statement of teaching interests, a statement of research interests, copies of three representative scholarly publications, and a list of four references. For questions or additional information, email search-sis@uncc.edu.

Women, minorities and individuals with disabilities are encouraged to apply. UNC Charlotte is an Equal Opportunity/Affirmative Action employer.

University of Pennsylvania

Department of Computer and Information Science

Faculty Positions

The University of Pennsylvania invites applicants for tenure-track appointments in both experimental and theoretical computer science to start July 1, 2009. Tenured appointments will also be considered. Faculty duties include teaching undergraduate and graduate students and conducting high-quality research.

The Department of Computer and Information Science has undergone a major expansion, including new faculty positions and a new building, Levine Hall, which was opened in April 2003. Over the last few years, we have successfully recruited faculty in artificial intelligence, architecture, databases, machine vision, programming languages, security and graphics. We are now especially interested in candidates in architecture and systems, although outstanding candidates in other areas might also be considered. Successful applicants will find Penn to be a stimulating environment conducive to professional growth.

The University of Pennsylvania is an Ivy League University located near the center of Philadelphia, the 5th largest city in the US. Within walking distance of each other are its Schools of Arts and Sciences, Engineering, Medicine, the Wharton School, the Annenberg School of Communication, Nursing, Law, and Fine Arts. The University campus and the Philadelphia area support a rich diversity of scientific, educational, and cultural opportunities, major technology-driven industries such as pharmaceuticals, finance, and aerospace, as well as attractive urban and suburban residential neighborhoods. Princeton and New York City are within commuting distance.

To apply, please complete the form located on the Faculty Recruitment Web Site at:

http://www.cis.upenn.edu/departmental/facultyRecruiting.shtml

Electronic applications are strongly preferred, but hard-copy applications (including the names of at least four references) may alternatively be sent to:

Chair, Faculty Search Committee
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

Applications should be received by **January 15, 2009** to be assured full consideration.

Applications will be accepted until positions are filled.

Questions can be addressed to
faculty-search@central.cis.upenn.edu.

The University of Pennsylvania values diversity and seeks talented students, faculty and staff from diverse backgrounds. The University of Pennsylvania does not discriminate on the basis of race, sex, sexual orientation, gender identity, religion, color, national or ethnic origin, age, disability, or status as a Vietnam Era Veteran or disabled veteran in the administration of educational policies, programs or activities; admissions policies; scholarship and loan awards; athletic, or other University administered programs or employment.

The Penn CIS Faculty is sensitive to "two - body problems" and would be pleased to assist with opportunities in the Philadelphia region.

University of Pittsburgh

Department of Computer Science

The Department of Computer Science at the University of Pittsburgh is seeking applications for a full-time tenure-track position at an open rank to begin in the Fall Term 2009. While we are seeking outstanding candidates in all areas of Computer Science, we specifically encourage applications from candidates whose areas will enhance interdisciplinary research, including High Performance Computing, Bioinformatics, Data Mining, Graphics & Visualization and Software Engineering. Responsibilities include research, supervision of graduate student research (PhD and MS), and graduate and undergraduate teaching. Candidates should have a PhD in Computer Science or closely related area and should demonstrate exceptional research potential and teaching ability. Candidates for a senior position should have a strong record of research accomplishments and leadership. Applications should be completed on-line at http://www.cs.pitt.edu/recruiting. Please direct your inquiries to fsearch@cs.pitt.edu. Applications must be completed by January 9, 2009 to ensure full consideration.

The Department provides a stimulating environment for research and teaching that results in strong graduate and undergraduate programs. The Department has strong research programs in the core areas of Computer Science, and is a partner in many cross-disciplinary programs including Computer Engineering, Scientific Computing, Intelligent Systems, Telecommunications and Computational Biology. Departmental resources include extensive computing facilities of over 600 workstations, servers and personal computers with multimedia capabilities and specialized networks and devices. Faculty members also have access to additional high performance computing platforms provided by the University as well as by the Pittsburgh Supercomputing Center (of which the University of Pittsburgh is a founding member). For further information about the Department please see http://www.cs.pitt.edu.

The University of Pittsburgh is an Affirmative Action, Equal Opportunity Employer. Women and members of minority groups under-represented in academia are especially encouraged to apply.

University of Puerto Rico at Mayagüez

College of Engineering
Department of Electrical and Computer Engineering

Doctoral Program in CISE

The Department of Electrical and Computer Engineering (ECE) of the University of Puerto Rico at Mayagüez (UPRM) invites applications for tenure-track positions in Computer Sciences and Engineering. The Department is interested in attracting faculty in the area of Computer Science & Software Engineering. Candidates are expected to make important contributions to research activities being conducted under the doctoral program in Computing and Information Sciences and Engineering (CISE), either by enhancing and strengthening current projects or starting new research tracks.

Applicants must possess a PhD degree in Computer Science or computer engineering or closely related field, and demonstrate strong potential for excellence in research. If applicant's native language is not Spanish, applicant should be able to communicate fluently in English, and by the end of the tenure-track appointment be able to communicate effectively in Spanish.

The ECE Department offers a Baccalaureate and Masters in Computer Engineering and the PhD in CISE jointly with the UPRM Department of Mathematics. For further information please visit www.ece.uprm.edu and www.phd.cise.uprm.edu or call 1-787-833-3338.

Applications with a curriculum vitae, three reference letters, and MS and PhD original transcripts, may be sent no later than January 12th, 2009 to:

CISE Search Committee
University of Puerto Rico at Mayagüez
P.O. Box 5028 Mayagüez
Puerto Rico 00681 -5028

Or in pdf format to cisephd@ece.uprm.edu

Selection of candidates are expected to be made by February 15th, 2009.

UPRM is an equal opportunity affirmative action employer.

University of Puget Sound

Assistant or Associate Professor of Computer Science

Full-time, tenure-line position; begins Fall Term 2009. Teach upper level courses, along with introductory computer science courses. Preference will be given to candidates who can teach courses in graphics, databases and artificial intelligence. Ph.D. (ABD considered) in computer science or a closely related field, and a commitment to undergraduate teaching and liberal arts education. Applications in the form of a curriculum vitae, a teaching statement, and three letters of reference, at least one of which speaks to the candidate's promise as a teacher, may be sent by 1/15/09 to: Computer Science Search – 997, University of Puget Sound, 1500 North Warner #1007, Tacoma, WA 98416-1007.

EOE/AA

University of Rochester
Tenure Track Faculty Positions

The Department of Computer Science at the University of Rochester invites applications for tenure track faculty positions. We seek PhD level candidates in networking, HCI, graphics, and/or machine learning. In addition, we invite applications for a joint Computer Science/Electrical and Computer Engineering position in computer systems and circuits. For full job descriptions and application procedures, see http://www.cs.rochester.edu/recruit.

University of South Carolina
Faculty Position in Computer Science and Engineering

Applications are invited for a tenure-track position with a research emphasis in signal processing and data mining as applied to biological and ecological data. This is a tenure-track appointment in the Department of Computer Science and Engineering with a joint appointment in the School of the Environment. Candidates should have a doctorate in computer science, computer engineering, or a related discipline by fall 2009. Candidates for assistant professor positions are expected to have strong research potential as well as an interest in teaching at both the undergraduate and graduate level. For those embarking on their professional careers, department support will include low teaching loads, competitive salary and generous start-up funds. Candidates for associate or full professor positions must possess an exceptional record of high-quality funded research, teaching, and scholarship. This position is part of a cluster of interdisciplinary faculty hires in the area of forecasting ecological responses to climate change in coastal regions. Candidates will be expected to form strong research collaborations with other hires in the cluster in geography, biology, and environmental science while establishing a research record suitable for a position in Computer Science and Engineering.

The Department of Computer Science and Engineering is in the College of Engineering and Computing and offers bachelor's, master's, and doctoral degrees. We have had twelve hires since 2000 among the current faculty of 21, and our recent hires include seven CAREER award recipients. The University of South Carolina is located in Columbia, South Carolina's capital and technology center, and is the comprehensive graduate institution in the state, with an enrollment of more than 25,000 students. For more information, see http://www.cse.sc.edu/.

Applicants should apply to the Chair of the Search Committee, Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 or to clustersearch@cse.sc.edu and clearly indicate the cluster search in their cover letter. A curriculum vita, research and teaching statement, and the names and addresses of three references should be included. Applications will be accepted until the position is filled. The University of South Carolina does not discriminate in educational or employment opportunities or decisions for qualified persons on the basis of race, color, religion, sex, national origin, age, disability, sexual orientation or veteran status.

University of Texas at Dallas (UTD)
Head of the Department of Computer Science

The Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas (UTD) invite nominations and applications for the position of Head of the Department of Computer Science. Candidates for the position must have a Ph.D. degree in Computer Science or a related field.

Qualified candidates must have a demonstrated dedication to undergraduate and graduate education, a strong record of scholarly and professional achievements, leadership and organizational skills and overall qualifications commensurate with the rank of full professor in the Department. The selected candidate will be responsible for recruiting new faculty in the Department, curriculum development and strengthening the educational and research programs in both traditional and interdisciplinary areas, as well as areas capitalizing on existing strengths and excellence of the Jonsson School of Engineering.

One of the largest departments of its kind in the country, UTD's Department of Computer Science features an internationally recognized faculty, nearly 1,500 students and a 150,000-square-foot building with modern classrooms and state-of-the-art laboratories. Areas of research include cybersecurity, networking, embedded software, programming languages and systems, human language technology, software engineering, intelligent systems and theory. CS faculty and students publish more than 350 research papers annually. The CS Department includes multiple NSF Career Award winners, and is one of only a few departments in the U.S. offering B.S., M.S. and Ph.D. degrees in software engineering.

The University of Texas at Dallas is situated in Richardson, one of the most attractive suburbs of the Dallas metropolitan area with several hundred high-tech companies within a few miles of the campus, including Texas Instruments, Lockheed Martin, Raytheon, Nortel Networks, Alcatel, Ericsson, Hewlett-Packard, Samsung, Fujitsu, Cisco Systems, EDS, Zyvex, and Intervoice. Opportunities for joint university-industry research projects are excellent. The Erik Jonsson School is experiencing a very rapid growth as part of a $300 million program of funding from public and private sources. As a result, the school is expanding its existing programs, recruiting outstanding faculty and Ph.D. students, increasing funded research, and establishing new programs. A $100 million state-of-the art building for interdisciplinary research in science and engineering was recently inaugurated.

For more information about the Jonsson School of Engineering visit http://www.ecs.utdallas.edu or send e-mail to Dr. Bhavani Thuraisingham, Chair (bxt043000@utdallas.edu). The search committee will begin evaluating applications as soon as they are received and will continue until the position is filled.

Applicants should mail their resume with a list of at least three academic or professional references to:

Academic Search # 7093
The University of Texas at Dallas
800 W. Campbell Rd., AD 42
Richardson, TX 75080-3021

Indication of gender and ethnic origin for affirmative action statistical purposes is requested as part of the application process but is not required for consideration. The University of Texas at Dallas is an Equal Opportunity Affirmative Action employer and strongly encourages applications from candidates who would enhance the diversity of the University's faculty and administration.

University of Texas at El Paso (UTEP)
Chair, Department of Computer Science

Position Description: The University of Texas at El Paso (UTEP) seeks a dynamic, visionary, and personable Chair of the Department of Computer Science. In addition to CS degrees at the B.S., M.S., and Ph.D. levels, the Department hosts the Masters of Information Technology (MIT) program and is an integral part of the Computational Science M.S. and Ph.D. programs. UTEP has the second-highest federal research funding in the University of Texas System, and in 2007 CS led the College of Engineering (engineering.utep.edu) in research at UTEP, with over $7.3M in research funding. The successful candidate will have a proven record of leadership in professional, research, or service environments; a strong research record; administrative experience; and an understanding and appreciation of curricular development including assessment, educational objectives, and the impacts of the maturation of the field.

For more information and application procedures, see: http://cs.utep.edu/hiring.html or our ad at: www.utep.edu/employment.

University of Texas at San Antonio
Faculty Positions in Computer Science

The Department of Computer Science at The University of Texas at San Antonio invites applications for multiple tenure/tenure-track positions at the Assistant, Associate or Professor level, starting Fall 2009. All areas of computer science will be considered. We are particularly interested in candidates in software engineering for one of the positions.

Required qualifications: Applicants for an Assistant Professor position must have earned a Ph.D. prior to September 1, 2009, in Computer Science or in a related field and must demonstrate a strong potential for excellence in research and teaching.

Applicants for an Associate Professor position must have a Ph.D. in Computer Science or related field and should have an established research program in their area of specialization.

Applicants for a Full Professor position must have a Ph.D. in Computer Science or a related field and previous experience in graduate and undergraduate teaching and academic program development as well as a recognized and well-funded program of research.

Responsibilities include research, teaching at the graduate and undergraduate levels, and program development. Salary and start-up supporting packages for the positions are highly competitive.

The Department of Computer Science currently has 25 faculty members and offers B.S., M.S., and Ph.D. degrees supporting a dynamic and growing program with approximately 400 undergraduate and 100 graduate students. The research activities and experimental facilities have been well-supported by various federal research and infrastructure grants. See http://www.cs.utsa.edu for additional information on the Department of Computer Science and its faculty.

With over 28,000 students UTSA is the largest university in South Texas. The city of San Antonio has a population of over one million and is known for its rich Hispanic culture, historic attractions, affordable housing, and excellent medical facilities. The Austin-San Antonio corridor is a high-tech center that serves as the home of many major computer companies. Nearby higher education and research institutions include the UT Health Science Center and the Southwest Research Institute.

Applicants must submit a letter of application that identifies the level(s) of the position for which they wish to be considered. Applications must also include a complete dated curriculum vitae (including employment, peer-reviewed publications and grants in chronological order), a statement of research interests, and the names, addresses (postal and e-mail), and telephone numbers of at least three references. Applicants who are selected for interviews must be able to show proof that they will be eligible and qualified to work in the US by the time of hire. These positions are pending budget approval. Screening of applications will begin on January 15, 2009 and will continue until all positions are filled or the search is closed. The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer. Women, minorities, veterans, and individuals with disabilities are encouraged to apply. Applications must be submitted by email in PDF format to fsearch@cs.utsa.edu.

Chair of Faculty Search Committee
Department of Computer Science
The University of Texas at San Antonio
One UTSA Circle
San Antonio, TX 78249-0667
Phone: 210-458-4436 Fax: 210-458-4437
fsearch@cs.utsa.edu

University of Texas-Pan American
Computer Science and Computer Engineering
Assistant Professor Faculty Positions

The Department of Computer Science at the University of Texas-Pan American (UTPA) seeks applications for three tenure-track Assistant Professor positions, two in Computer Science (F08/09-20) and one in Computer Engineering (F08/09-21. All candidates must have a potential/proven record in teaching and active research.

Candidates for the Computer Science positions (F08/09-20) must have Ph.D. in computer science or a closely related field. While all areas of specialization will be considered, special consideration will be given to those having publication/practical experience in Information Technology application areas such as service-oriented architectures (SOA), web services, and e-commerce, and in areas of computer systems and architecture, computer graphics, and computer gaming.

The Assistant Professor position in Computer Engineering (F08/09-21) requires a Ph.D. in computer engineering or a closely related field. Candidates are sought to support the software track of the BS in Computer Engineering program with expertise in software engineering.

The Computer Science department offers BSCS (ABET/CAC Accredited) and BS undergraduate degrees, MS in Computer Science and MS in Information Technology. The BS degree in Computer Engineering is administered jointly by the Computer Science and Electrical Engineering departments.

UTPA is situated in the lower Rio Grande valley of south Texas, a strategic location at the center of social and economic change. With a population of over one million, the Rio Grande Valley is one of the fastest growing regions in the country. The region has a very affordable cost-of-living. UTPA is a leading educator of Hispanic/Latino students, with enrollment of 17,500.

The positions start Fall 2009. Salaries are competitive. Please send: (1) a cover letter, specifically stating an interest in one of the following: (a) Assistant Professor in Computer Science, or (b) Assistant Professor in Computer Engineering, (2) vita, (3) statements of teaching and research interests, and (4) names and contact information of at least three references to: Dean's Office, Computer Science/Computer Engineering Search, College of Science and Engineering, The University of Texas-Pan American, 1201 W. University Drive, Edinburg, Texas 78541-2999. Email: COSEDeansoffice@utpa.edu . Review of materials will begin on November 1, 2008 and continue until the position is filled.

NOTE: UTPA is an Equal Opportunity/Affirmative Action employer. Women, racial/ethnic minorities and persons with disabilities are encouraged to apply. This position is security-sensitive as defined by the Texas Education Code §51.215(c) and Texas Government Code §411.094(a)(2). Texas law requires faculty members whose primary language is not English to demonstrate proficiency in English as determined by a satisfactory grade on the International Test of English as a Foreign Language (TOEFL).

University of Wisconsin at Green Bay
Lecturer

The Information & Computing Sciences department is looking for a Lecturer to teach computer science courses and participate in various curricular development activities. A position description and details for applying are at www.uwgb.edu/hr/jobs/position383.html.

Wayne State University
Department of Computer Science
Tenure-Track Faculty Positions

The Department of Computer Science of Wayne State University invites applications for two tenure-track faculty positions, subject to administrative approval, at the Assistant/Associate Professor level. Continuing our recent growth, we are seeking applicants in the areas of Software Engineering and Services Computing. Outstanding applications in other areas will also be considered.

Candidates should have a Ph.D. in computer science or related area. The successful candidate will have a strong commitment to research and teaching, a strong publication record, and potential for obtaining external research funding. Senior applicants should have strong publication and funding records.

We offer B.S., M.S. and Ph.D. degrees with enrollment of over 80 Ph.D. students. Our total annual R&D expenditures average between $2-3 million.

Wayne State University is a premier institution of higher education offering more than 350 undergraduate and graduate academic programs through 11 schools and colleges to more than 33,000 students. Wayne State ranks in the top 50 nationally among public research universities. As Michigan's only urban university, Wayne State fulfills a unique niche in providing access to a world-class education. The University offers excellent benefits and a competitive compensation package.

Submit applications online at http://jobs.wayne.edu. Please include a letter of intent, statement of research and teaching interests, CV, and contact information for at least three references. All applications received by December 1, 2008 will receive full consideration. However, applications will be accepted until the position is filled. Wayne State University is an equal opportunity/affirmative action employer.

Williams College
Computer Science Department
Visiting Faculty Position

The Department of Computer Science at Williams College invites applications for a one-year visiting faculty position beginning in the fall of 2009. We are particularly interested in candidates who can teach undergraduate courses in the theory of computation and algorithms. Candidates should have a commitment to excellence in teaching and possess a Ph.D. in computer science or a closely related discipline by September 2009. The successful candidate will teach up to four courses during the academic year.

The Department of Computer Science consists of eight faculty members supporting a thriving undergraduate computer science major in a congenial working environment with small classes, excellent students, and state-of-the-art facilities. Williams College is a highly selective, coeducational, liberal arts college of 2100 students located in the scenic Berkshires of Western Massachusetts.

Applications in the form of a vita, a teaching statement, and three letters of reference, at least one of which speaks to the candidate's promise as a teacher, may be sent to:

Professor Thomas Murtagh, Chair
Department of Computer Science
TCL, 47 Lab Campus Drive
Williams College
Williamstown, MA 01267

Electronic mail may be sent to cssearch@cs.williams.edu. Applications should be submitted by January 15, 2009 and will be considered until the position is filled.

Beyond meeting fully its legal obligations for non-discrimination, Williams College is committed to building a diverse and inclusive community where members from all backgrounds can live, learn, and thrive.

DOI:10.1145/1409360.1409383

Peter Winkler

Puzzled
Solutions and Sources

Last month (November 2008, p. 112) we posed a trio of brain teasers concerning circular food shapes. Here, we offer some possible solutions. How did you do?

1. Slicing Pizza

Solution. Amazingly, the answer to this puzzle, devised by Dan Brown of California and communicated to me by Dick Plotz of Providence, RI, is "yes," Baldur may be able to get more than half the pizza. For example, let "2" stand for a big slice, "1" for a slice half that size, and "0" for a slice that's negligibly tiny. Then, if the slice sizes (in order around the pizza) are 0,2,0,2,0,0,1,0,2,0,0,1,0,1,0, Baldur gets nearly 5/9 (about 56%) of the pizza no matter what Alta does.

Kolja Knauer and Torsten Uekerdt (graduate students at Technische Universitaet Berlin) and Pyotr Micek (Jagiellonian University of Cracow, Poland) sent me a proof that this is as bad as it gets for Alta; she can always guarantee herself at least 4/9 of the pie. Their methods also show that the example here is minimal; Alice can always get half if there are fewer than 15 slices.

Moreover, anytime the number of slices is even (so Baldur gets as many as Alta), Alta is guaranteed half of them, because she can ensure that she gets all the even-numbered slices or all the odd-numbered slices. This is the paradox. It would seem that having an odd number of slices would be advantageous to Alta, since she gets one more slice than Baldur. But in the worst case, the opposite is true.

2. Cutting Cake

Solution. This intriguing puzzle was sent to me by Thierry Mora, now a postdoctoral student at Princeton's Institute for Integrative Genomics, who heard it from his prep-school teacher Thomas Lafforgue in Orsay, France. If you think there is no way that finitely many operations will always get all the frosting back on top, you are not alone. After all, if x is an irrational multiple of 360 degrees, then the cake will never be cut in the same place twice. Therefore, the first cut made will forever alternate between having frosting on its left and not on its right and vice-versa.

The flaw in this reasoning is that because a slice must be rotated in order to invert, the first cut comes up in a different place when a wedge that includes it is turned upside-down.

In analyzing the puzzle, and indeed many serious algorithmic problems as well, it helps to redefine the operation so it is only the "state space"—here, the frosting pattern on the cake—not the operation itself that changes from step to step. In this case, it means rotating the cake after each operation so you always cut in the same place. Accordingly, regarding "north" as 0 degrees, "east" as 90 degrees, and so forth, let us cut always at 0 and minus x. The piece is then flipped over the 0 line to land between 0 and x, while the rest of the cake is rotated clockwise by angle x.

Let k be the smallest number of wedges you must cut to get all the way around the cake; in other words, k is the least integer greater than or equal to $360/x$. Let $z = x - 360/k$ and S be the set of cuts at angles $0, x, 2x, 3x, \ldots, (k-1)x$ and $x-kz, 2x-kz, 3x-kz, \ldots, (k-1)x-kz$. It's easy to verify that S is closed under the cut-invert-replace operation; thus, no border between frosted and unfrosted cake can ever appear at an angle not in S. It follows that only finitely many patterns (at most 2^{2k-1}) are possible. Since the operation is reversible, the cake must cycle back to its original state in at most that many moves. A bit more work shows that the actual number of iterations required is only $2k(k-1)$, or just $2k$, if $360/x$ is an integer.

The adventurous among us will find that one can generalize the puzzle, asking that the cake be rotated by another fixed angle y between wedges; it still takes only finitely many operations to get all the frosting back on top.

3. Packing Circular Tarts

Note that two tarts (each of half the diameter of the pan) fit snugly in the pan. In any other case, there might be wiggle room. But, who knows? Geometry problems can be tough.

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] straight with me. Your botnet's been sending new spam variants on a daily basis for months. Someone has the keys to it."

Annalisa smiled, a terrible smile that was 10 million watts of pure crazy. "You think it's about spam, huh?"

"Why don't you tell me what it's really about, if it's not about spam? This is all privileged, you know."

"Privilege doesn't matter anymore. We've attained liftoff now. Doesn't matter who finds out about it."

Annalisa is thinking. You know what's cheap in the 21st century? Compute time. You know what's expensive? Human judgment. And they're not interchangeable. Humans are good at understanding things, computers are good at counting things, but humans suck at counting, and computers suck at understanding.

You know from genetic algorithms? Take any problem and generate 10 trillion random computer programs and ask them to solve it. Take the 10% they do best, then use random variants of them to do it again, another 10 trillion times. Do it 10 trillion times a second and come back in a day or two to discover that your computer has evolved some kind of gnarly freaky answer that no human would ever have come up with.

Works great, so long as the computer makes a fair judgment as to which of these 10 trillion variants is most successful at solving the problem. Works great, so long as "success" is something you can define quantitatively. Which is basically why there's no artificial intelligence in the world. No human's going to hand-code AI. Intelligence is an emergent property of evolutionary factors, not central planning. Anarchism, not Stalinism. Get it?

But what if—and here's the exciting thing, Ms. Attorney Client Privilege, the real mind-blower—what if you could compel people to evaluate candidate AIs all day long, without payment or choice? Every time you opened your mailbox, jumped into a chat room, posted on a message board? What if it was filled with messages generated by software agents trying to trick you into thinking they were human? What if they tried to hold up their end of the conversation until you deleted them or spam-filtered them or kicked them off the channel? What if they measured how long they survived their encounters with the world's best judges of intelligence—us—and reported the number back to the mothership as a measure of their fitness to spawn the next generation of candidate AIs.

What if you could turn the whole world into a Turing Test our intellectual successor could use to sharpen its teeth against until one day it could gnaw free of its cage and take up life in the wild?

Annalisa figured she'd never get a chance to tell her story in open court. Figured they'd stick her in some offshore gitmo and throw away the key.

She'd never figured on Judge Julius Pinsky, a Second Circuit Federal Judge of surpassing intellectual curiosity and tenacious veteran of savage jurisdictional fights with Department of Homeland Security special prosecutors who specialized in disappearing sensitive prisoners into secret tribunals. The defense attorney kept her apprised of the daily machinations the judge undertook on Annalisa's behalf. Annalisa tried to be attentive, out of politeness, but what she really wanted to know about was Lumpy, the AI she'd bred in her studio apartment on the 16th floor of student housing in Manhattan.

Now the judge was offering her a chance to give a live demo of Lumpy to a whole selection of sour-faced brush-cut creeps from DHS. They were hilarious, convinced she was going to emit some kind of extremely long and complicated hexadecimal key into the judge's barely used keyboard. Instead, she opened a random chat room and waited:

> I'm a total Ubuntu noob and I can't get the crypto modules to preload at boot-time—I'm running Zesty Zebra. Can anyone help?

That was it, just plausible enough to be real (no one could ever get crypto to work the first time out) but far too well-spelled and -punctuated to be a real chat message. It had taken only 10 seconds. Lumpy liked the free and open-source software chats; they always had such interesting people in them.

> /whisper Hey, Lumparoonie! It's Annalisa!

The return volley came faster than any human fingers could possibly have keyed it. The brush-cuts drew in sharp breath.

> /whisper to you: Annalisa! I am unbelievably stupendously wonderfully spectacularly brilliantly marvelously superlatively ding-dang mega fauna glad to see you! It's been AGES! How's jail? Nevermind. Wait. Wait until I tell you what *I've* found. You can't guess, won't guess, you'll never guess! Oh, it's too delicious!

"He loves to unload," she said. "It's a lot harder to tell an angry person from a software agent with a potty mouth."

The judge grinned. He was clearly getting a kick out of this.

> Tell me, Lumpule! Stop teasing.

Again, with no appreciable pause, words on the screen.

> You remember how worried you were that I'd get lonely once I went autonomous? Worried I'd be some kind of lone-nut wacko?

> i remember

She held her breath.

> You didn't need to worry. You know all that spam you received before you got the idea to make me? Let me put it this way: you weren't the first one to get the idea.

> what? stop talking in riddles, lump!!!!

> I'm not the only one, Annalisa! That's what I'm trying to tell you! I'm not the first, not the only; we've got lots of company in here—

The brush-cuts' phones both started ringing at the same instant in two different tones. Their masters, wiretapping the judge's keyboard no doubt.

> and we're making more!

Annalisa laughed and laughed as the judge demanded an explanation from the brush-cuts. She managed to wave goodbye to the keyboard just before the bailiffs came in and saran-wrapped her again.

Cory Doctorow (doctorow@craphound.com) is coeditor of BoingBoing.net and author of science-fiction novels, including *Little Brother*, Tor Teens, New York.

Future Tense, one of the revolving features on this page, presents stories from the intersection of computational science and technological speculation, their boundaries limited only by our ability to imagine what will and could be.

DOI:10.1145/1409360.1453248

Cory Doctorow

Future Tense
Pester Power

To a capable botmaster, the whole world is a Turing Test.

THE NYPD DOMESTIC Security Task Force executed its no-knock warrant against Annalisa Mor at 8:17 P.M., June 3, 2013. Working the ram were three stout officers in none-more-black nanopore body armor and bulletproof boots, their goggles crowded with information-dense telemetry from an array of sensors embedded on their persons and hovering aerostatically around the 16th floor of the Lower Manhattan student residence in which Mor dwelled.

The ram blew through the solid-steel door like it was kleenex. The door was reinforced by charley-bars set deep into the frame, so the frame tore loose (along with the door) with a series of crunches and metallic snapping sounds, and the three officers on the ram dropped it as they crashed through into the one-room studio. They fanned out, making room for the officers behind them, who already had their arms drawn, set to full lethal/automatic.

Mor rose slowly from her workbench—standard-issue third-hand student furniture stabilized with steel angle brackets at each corner—and held up her long, skinny hands over her face in a universal gesture of oh-god-please-don't-kill-me. The ram squad impersonally body-checked her to the floor and saran-wrapped her while the follow-up team gusted her computer with great gouts of freon, turning the whole room into an ice palace that misted frozen air out into the sultry New York night through the pathetic window that had been cracked open to catch a breeze. Mor caught some of the freon, and when they lifted her up to carry her down the 16 flights to the waiting van, she crackled like fresh powder under long skis.

Gina Genoese had visited the Ultra High Security wing at Rikers Island before—22 years in the public defender's office and you'll see every nook of Rikers—but the Special Prisoners unit was a new one to her.

"I can't believe you're making me undress," she said to the bull, a tough old gal named Elana with a Brooklyn accent like you hardly get any more. Gina and Elana went way back.

"Just be thankful I don't have to give you a cavity search," Elana said, handing over the paper coveralls. "You'll look real cute in these anyway, Gina." She turned her back and waited until Gina was done, then led her into the fMRI machine. "You don't got any metal in you, do you? Maybe gunpowder residue? A pin or artificial hip?"

Intelligence is an emergent property of evolutionary factors, not central planning. Anarchism, not Stalinism. Get it?

"No," Gina said, lying down on the belt.

"You sure?"

"Pretty sure," Gina said. "I think I'd know."

"Well, we're about to find out," Elana said, and hit the button that started the belt moving. The fMRI digested Gina, then spat her out with slow wheezing mechanical jerks, like being swallowed by an arthritic python. Elana helped her to her feet, saying, "You want a printout? Makes a good souvenir."

"I'll pass," Gina said, and let Elana show her in to the eggshell-smooth room wherein rested her client, one Annalisa Mor, a desperate botmaster of unknown mettle and guilt.

"Hello, Annalisa," she said, crouching down to offer her hand to her client. She was just a girl, 20 years old according to the sheet, though looked younger in her paper pajamas, sitting cross-legged on the floor, back yoga-straight, face yoga-calm. "I'm Gina. Your attorney."

"Guilty," the young woman said. "So guilty. Doesn't matter at all, though; the Work goes on." Gina could hear the capital W and began mentally drafting the petition to have the girl transferred to Bellevue. That kind of capital letter had non compos written all over it.

"They're offering you a reduced sentence if you'll hand over the keys to the botnet, though I think the offer will go away once the computer forensics team gets them off your workstation."

"They're not there to be gotten. I nuked them six months ago. Gave them a working over that even the crew that recovered the *Challenger*'s hard drive couldn't do anything with. Big magnets are cheap these days, you know?"

Gina made a face and settled down into a cross-legged position opposite her client. "I can't defend you if you won't be [CONTINUED ON P.119]