

Computer

Innovative Technology for Computer Professionals

JANUARY 2010

<http://www.computer.org>

OUTLOOK

IEEE Computer Society
President's Message, p. 6

Open Source Foundations, p. 86

Reducing IT Energy Use, p. 91



IEEE
computer
society

TIMELY, ENVIRONMENTALLY FRIENDLY DELIVERY

DIGITAL EDITIONS

Subscribe to the interactive digital versions of *Computer* and *IEEE Security & Privacy*, and access the latest news and information whenever and wherever you want it.

Computer

The IEEE Computer Society's flagship publication, *Computer* magazine publishes peer-reviewed technical articles on all aspects of computer science, computer engineering, technology, and applications.

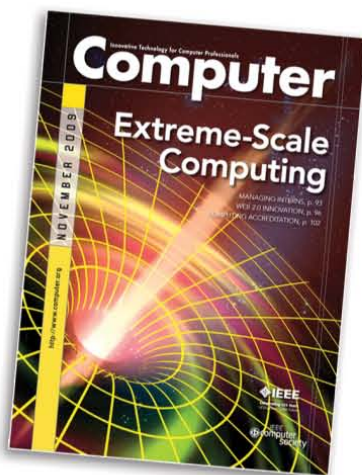
Industry professionals, researchers, and managers rely on *Computer* to keep current on research developments, trends, best practices, and changes in the profession.

Upcoming theme issues include:

- Extreme-scale computing,
- Multi- and many-core,
- Biometric identification, and
- Nano-architecture.

To see what you're missing, check out selected *Computer* articles for free in Computing Now, and then subscribe to the digital edition to get full access right away.

\$29.95
for 12 issues!



IEEE Security & Privacy

IEEE Security & Privacy brings together the practical and the leading edge advances in security, privacy, and dependability.

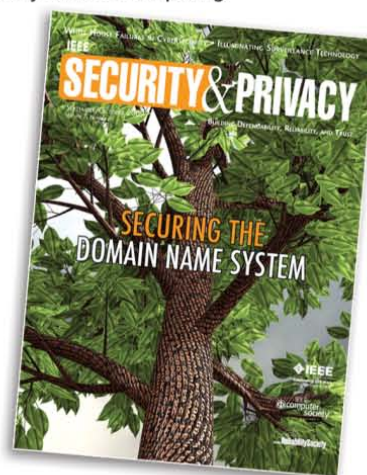
IEEE Security & Privacy covers and influences policy in the enterprise and the government—from basic training and attack trends to the US's cyberattack policy and telephone wiretapping, *S&P* brings guidance from some of the leading thinkers in the field. Bruce Schneier, Steve Bellovin, Gary McGraw, and Mike Howard have you in mind when writing their columns!

Upcoming theme issues include:

- The insider threat,
- Mobile device security, and
- The security and privacy of cloud computing.

Sample free *IEEE Security & Privacy* articles and the Silver Bullet podcast series from Computing Now, and subscribe to the digital edition today.

\$19.95
for 6 issues!



The latest content at your fingertips within minutes.

Email notification. Receive an alert as soon as each digital edition is available. Links will take you directly to the enhanced PDF edition OR the web browser-based edition.

Quick access. Download the full issue in less than two minutes with a broadband connection.

Convenience. Read your digital edition anytime -- from your home PC, at work, or on your laptop while traveling.

Digital archives. Subscribers can access the virtual archive of digital issues dating back to Jan./Feb. 2007.

To subscribe, go to: computer.org/digitaleditions

IEEE
computer
society

Innovative Technology for Computer Professionals

Computer

Editor in Chief

Carl K. Chang
Iowa State University
chang@cs.iastate.edu

Associate Editor in Chief

Sumi Helal
University of Florida
helal@cise.ufl.edu

Associate Editor in Chief, Research Features

Kathleen Swigger
University of North Texas
kathy@cs.unt.edu

Associate Editor in Chief, Special Issues

Bill N. Schilit
Google
schilit@computer.org

Computing Practices

Rohit Kapur
rohit.kapur@synopsys.com

Perspectives

Bob Colwell
bob.colwell@comcast.net

Web Editor

Ron Vetter
vettterr@uncw.edu

2010 IEEE Computer Society President

James D. Isaak
csresident2010@jimisaak.com

Area Editors

Computer Architectures

Steven K. Reinhardt
AMD

Databases and Information Retrieval

Erich Neuhold
University of Vienna

Distributed Systems

Jean Bacon
University of Cambridge

Graphics and Multimedia

Oliver Bimber
Johannes Kepler University Linz

High-Performance Computing

Vladimir Getov
University of Westminster

Information and Data Management

Naren Ramakrishnan
Virginia Tech

Multimedia

Savitha Srinivasan
IBM Almaden Research Center

Networking

Ahmed Helmy
University of Florida

Software

Robert B. France
Colorado State University

David M. Weiss
Iowa State University

Column Editors

Education

Ann E.K. Sobel
Miami University

Embedded Computing

Tom Conte
Georgia Tech

Green IT

Kirk W. Cameron
Virginia Tech

IT Systems Perspectives

Richard G. Mathieu
James Madison University

Invisible Computing

Albrecht Schmidt
University of Duisburg-Essen

The Known World

David A. Grier
George Washington University

The Profession

Neville Holmes
University of Tasmania

Security

Jeffrey M. Voas
NIST

Software Technologies

Mike Hinchey
Lero—the Irish Software Engineering Research Centre

Web Technologies

Simon S.Y. Shim
San Jose State University

Advisory Panel

Thomas Cain
University of Pittsburgh

Doris L. Carver
Louisiana State University

Ralph Cavin
Semiconductor Research Corp.

Dan Cooke
Texas Tech University

Ron Hoelzeman
University of Pittsburgh

Naren Ramakrishnan
Virginia Tech

Ron Vetter
University of North Carolina at Wilmington

Alf Weaver
University of Virginia

CS Publications Board

David A. Grier (chair), David Bader, Angela R. Burgess, Jean-Luc Gaudiot, Phillip Laplante, Dejan Milošević, Linda I. Shafer, Dorée Duncan Seligmann, Don Shafer, Steve Tanimoto, and Roy Want

CS Magazine

Operations Committee

Dorée Duncan Seligmann (chair), David Albonesi, Isabel Beichl, Carl Chang, Krish Chakrabarty, Nigel Davies, Fred Douglass, Hakan Erdogmus, Carl E. Landwehr, Simon Liu, Dejan Milošević, John Smith, Gabriel Taubin, Fei-Yue Wang, and Jeffrey R. Yost

Editorial Staff

Scott Hamilton
Senior Acquisitions Editor
shamilton@computer.org
Judith Prow
Managing Editor
jprow@computer.org
Chris Nelson
Senior Editor
James Sanders
Senior Editor

Contributing Editors

Lee Garber
Bob Ward

Design and Production

Larry Bauer
Design
Olga D'Astoli
Cover Design
Kate Wojogbe

Administrative Staff

Products and Services Director
Evan Butterfield
Magazine Editorial Manager
Jennifer Stout

Senior Business Development Manager

Sandy Brown
Senior Advertising Coordinator
Marian Anderson

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society. **IEEE Headquarters**, Three Park Avenue, 17th Floor, New York, NY 10016-5997; **IEEE Computer Society Publications Office**, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; voice +1 714 821 8380; fax +1 714 821 4010; **IEEE Computer Society Headquarters**, 2001 L Street NW, Suite 700, Washington, DC 20036. IEEE Computer Society membership includes \$19 for a subscription to *Computer* magazine. Nonmember subscription rate available upon request. Single-copy prices: members \$20.00; nonmembers \$99.00.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in USA.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

Innovative Technology for Computer Professionals

Computer

<http://computer.org/computer>

CONTENTS

ABOUT THIS ISSUE

In our January Outlook issue, we go back in time to look forward. Larry Smarr chronicles the growing threat of global climatic disruption and the key role the ICT community can play in this looming crisis. David Lorge Parnas invokes Robert Floyd’s foundational 1967 article to show how we might end a perceived impasse in formal methods. Simon Dobson and colleagues look back at IBM’s 2001 call to arms and assess progress toward autonomic systems engineering over the past decade. Finally, Marco Conti and Mohan Kumar explore a computing paradigm that exploits opportunistic communication between pairs of devices and the applications executing on them.



COVER FEATURES

22 Project GreenLight: Optimizing Cyber-infrastructure for a Carbon-Constrained World

Larry Smarr

Even with a variety of aggressive energy efficiency measures, the ICT sector’s carbon emissions will nearly triple from 2002 to 2020. We must develop ways to make our ICT systems more energy efficient so that we can use more of them in smart infrastructure that has great potential for reducing global greenhouse gas emissions.

28 Really Rethinking ‘Formal Methods’

David Lorge Parnas

We must question the assumptions underlying the well-known current formal software development methods to see why they have not been widely adopted and what should be changed.

35 Fulfilling the Vision of Autonomic Computing

Simon Dobson, Roy Sterritt, Paddy Nixon, and Mike Hinchey

Efforts since 2001 to design self-managing systems have yielded many impressive achievements, yet the original vision of autonomic computing remains unfulfilled. Researchers must develop a comprehensive systems engineering approach to create effective solutions for next-generation enterprise and sensor systems.

42 Opportunities in Opportunistic Computing

Marco Conti and Mohan Kumar

When two devices come into contact, albeit opportunistically, it provides a great opportunity to match services to resources, exchange information, cyberforage, execute tasks remotely, and forward messages.

RESEARCH FEATURES

51 A Discrete Stock Price Prediction Engine Based on Financial News

Robert P. Schumaker and Hsinchun Chen

The Arizona Financial Text system leverages statistical learning to make trading decisions based on numeric price predictions. Research demonstrates that AZFinText outperforms the market average and performs well against existing quant funds.

58 Online Security Threats and Computer User Intentions

Thomas F. Stafford and Robin Poston

Although computer users are aware of spyware, they typically do not take protective steps against it. A recent study looks into the reasons for this apathy and suggests boosting users’ confidence in installing and operating antispyware solutions as an effective remedy.

For more information on computing topics, visit the Computer Society Digital Library at www.computer.org/csdl.

IEEE Computer Society: <http://computer.org>
Computer: <http://computer.org/computer>
computer@computer.org
IEEE Computer Society Publications Office: +1 714 821 8380
Cover image © Lolaferari | Dreamstime.com

Flagship Publication of the IEEE
Computer Society

January 2010, Volume 43, Number 1

9 The Known World

Designing the Future
David Alan Grier

12 32 & 16 Years Ago

Computer, January 1978 and 1994
Neville Holmes

NEWS

14 Technology News

Is 3D Finally Ready for the Web?
Sixto Ortiz Jr.

17 News Briefs

Linda Dailey Paulson

MEMBERSHIP NEWS

6 President's Message

65 IEEE Computer Society
Connection

70 Call and Calendar

COLUMNS

77 Web Technologies

Web 3.0: The Dawn of Semantic Search
James Hendler

81 Embedded Computing

Mobile Supercomputers for the Next-
Generation Cell Phone
Mark Woh, Scott Mahlke, Trevor Mudge,
and Chaitali Chakrabarti



86 Industry Perspective

The Economic Case for Open Source Foundations
Dirk Riehle

91 Green IT

Proxying: The Next Step in Reducing
IT Energy Use
Bruce Nordman and Ken Christensen

96 The Profession

The Varieties of Reality
Neville Holmes

DEPARTMENTS

4 Elsewhere in the CS

21 Computer Society Information

72 Career Opportunities

76 Advertiser/Product Index



Reuse Rights and Reprint Permissions: Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post their IEEE-copyrighted material on their own Web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy.

Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2010 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

ELSEWHERE IN THE CS



Computer Highlights Society Magazines

The IEEE Computer Society offers a lineup of 13 peer-reviewed technical magazines that cover cutting-edge topics in computing including scientific applications, design and test, security, Internet computing, machine intelligence, digital graphics, and computer history. Select articles from recent issues of Computer Society magazines are highlighted below.

Software

Time and again, software projects fail. Some of the reasons for failure relate to software architecture. In the November/December 2009 installment of *The Pragmatic Architect*, Siemens' Frank Buschmann discusses two mistakes that aren't the prime responsibility of architects, but that directly affect architects if they occur: missing, wrong, or creeping system scope; and vague, unnecessary, or extreme nonfunctional requirements. Not addressing these mistakes can lead software projects into trouble before concrete architecture elaboration even begins.

IT Professional

TECHNOLOGY SOLUTIONS FOR THE ENTERPRISE

Trust in the workplace is both an ethical and a management issue. And although virtual teams have become common in IT—and promise to become even more popular as companies tighten travel budgets—the importance of building trust in such teams is often underappreciated. In "Virtual Teams and the Importance of Building Trust" in the November/December 2009 issue of *IT Pro*, authors Georgina Harell and Tugrul U. Daim of Portland State University examine the application of various definitions and theories of trust to virtual IT teams.

Computer Graphics and Applications

The November/December 2009 issue of *CG&A* features an article on 3D interaction by Alexander Kulik of Bau-

haus-Universität Weimar. "Building on Realism and Magic for Designing 3D Interaction Techniques" looks at how imagination-based interaction can complement reality-based interaction in the design of 3D user interfaces. This hybrid approach could lead to interface design guidelines that promote higher-level consistency, and thus usability, for a large range of diverse interfaces.

Computing

SCIENCE & ENGINEERING

Software testing can improve software quality. To test effectively, scientists and engineers should know how to write and run tests, define appropriate test cases, determine expected outputs, and correctly handle floating-point arithmetic. Using the Matlab xUnit automated testing framework, scientists and engineers can make software testing an integrated part of their software development routine.

CiSE is offering a preprint version of "Automated Software Testing for Matlab," by Steven Eddins of MathWorks, which describes the basic mechanics of automated unit testing.

SECURITY & PRIVACY

IEEE

An article featured in the November/December 2009 issue of *S&P* asserts that trusted insiders who misuse their privileges to gather and steal sensitive information represent a potent threat to businesses. In "Detecting Insider Theft of Trade Secrets," Deanna D. Caputo, Marcus A. Maloof, and Gregory D. Stephens tell how a prototype system developed by researchers at MITRE for identifying insider threats prompted a team of engineers and social scientists to study how malicious insiders use information differently from a benign baseline group.

Intelligent Systems

Web science is an emerging field that studies the origins, state, and future of the World Wide Web as both a

critical global infrastructure and a socially transforming phenomenon. In the first of two special issues looking at society online, guest editors James Hendler of Rensselaer Polytechnic Institute and Wendy Hall of the University of Southampton present the first six of 11 articles selected from presentations at the first international Web Science Conference, held in 2009 in Athens. The selected articles in the November/December issue of *IS* represent interesting implications for intelligent systems and the interdisciplinary nature of Web science.

IEEE Internet Computing

In “Phone + Internet Café = Secure Banking? You Betcha,” in the November/December 2009 issue of *IC*, Fred Douglass, editor in chief of *IC*, casts a wary glance at advances in mobile security

Some banks now offer an added level of security, requiring a temporary passcode obtained via SMS on a mobile phone or a SecurID dongle to log in. There’s even the possibility of using that bank as a springboard to access other accounts without providing the password. In theory, this might offer enough security to let a traveler do remote banking even at an insecure Internet café, but the author will stick to his laptop for now.

IEEE pervasive COMPUTING

Resource poverty is a fundamental constraint that severely limits the class of applications that can be run on mobile devices. The authors of “The Case for VM-Based Cloudlets in Mobile Computing” present a vision of mobile computing that breaks free of this fundamental constraint. In this vision, mobile users seamlessly utilize nearby computers to obtain the resource benefits of cloud computing without incurring WAN delays, jitter, congestion, and failures.

By Mahadev Satyanarayanan of Carnegie Mellon University, Paramvir Bahl of Microsoft Research, Ramón Cáceres of AT&T Labs, and Nigel Davies of Lancaster University, the article appears in the October-December 2009 issue of *PvC*.

IEEE micro

Low-power, high-speed chips, or “cool chips,” aim to reduce power consumption and enhance performance for applications ranging from multimedia to robotics. The Cool Chips conference series focuses on all aspects of cool technologies. The November/December 2009 special issue of *Micro* captures not only highlights from Cool Chips 2009 presentations, but also from ordinal submissions. Major topics at Cool Chips XII included multicore, video codec, and recognition processors.

IEEE MultiMedia

An October-December 2009 special issue of *MultiMedia* addresses multimedia metadata and semantic management. Authors present new research that focuses on interoperable, intelligent access to and management of multimedia materials. Guest editors Richard Chbeir of Bourgogne University, Harald Kosch of the University of Passau, Frederic Andres of the National Institute of Informatics, Tokyo, and Hiroshi Ishikawa of Shizuoka University present six articles that explore the application of Semantic Web technologies to multimedia content—assessing current technologies and exploring the major challenges and solution approaches.

IEEE Design & Test of Computers

The November/December 2009 special issue of *D&T* addresses the reliability challenges of VLSI chip design at 32 nanometers and beyond. Guest editors Yu Cao of Arizona State University, Jim Tschanz of Intel, and Pradip Bose of IBM introduce six articles that highlight R&D efforts to cope with progressively more unreliable components at the device, circuit, and system levels in the late CMOS era.

IEEE Annals of the History of Computing

An October-December 2009 special issue of *Annals* on the history of database management systems leads with DBMS prehistory: “How Data Got Its Base: Information Storage Software in the 1950s and 1960s.” Computing historian Thomas Haigh of the University of Wisconsin describes the foundations of DBMSs in the experiences and practices of administrative computing specialists working on report generators and file maintenance during the 1950s. He also explores the influences of the managerial and organizational contexts that drove work on “total information management systems” during the 1960s.

Editor: Bob Ward, *Computer*; bnward@computer.org

computing now

<http://computer.org/cn/ELSEWHERE>

January Theme:

INTERNET PREDICTIONS

PRESIDENT'S MESSAGE

Computer Society 2010: Innovation, Leadership, Community, and Careers



➔ **James D. Isaak**
*IEEE Computer Society
2010 President*



The Computer Society's business is to help members access the right resources and technical information, support them in their careers, and facilitate relationships with like-minded professionals that offer opportunities for mentoring and collaboration.

Colleagues, it is an honor and challenge to take on the role of Computer Society (CS) president for 2010. Our Society's business is to help you access the right resources and technical information when you need them, to support you in your career, and to facilitate relationships with like-minded professionals that can be used for mentoring, collaboration, and simple enjoyment. As a CS member, you have the opportunity to increase your innovation, build leadership skills, take advantage of vital technical communities, and develop your career. However, these opportunities do not result just from having a membership card and receiving *Computer*. You need to become engaged to get these real benefits.

**INVOLVEMENT AT THE
LOCAL LEVEL**

Your most immediate chance to become involved is with your local section or chapter. Your IEEE section meetings provide a place where

you can interact with a wide range of technical professionals. The perspectives of these individuals can provide insight into challenges you have in your own work and be the catalyst for innovation that leads to new features, products, and companies.

Our relationship with IEEE provides you with connections to folks in power systems telecommunications, consumer electronics, biomedical systems, and many more areas. In today's world of complex integrated systems, collaborating across these boundaries can be the key to the next big thing, from the smart grid to robotics. To pursue "out of the box" thinking, you need to get out of your office, and the activities of your local section and chapter provide a way to do this.

If there are no activities close by, consider arranging some—a first step on the leadership ladder that our Society provides. Chapters annually seek new officers and typically are looking for speakers and volunteers to coordinate events or help out with programs already in place. The same is true of your section. When you

pursue your interests in this context, you start making contacts with like-minded individuals and can increase both your leadership skills and your competencies.

The Society has a wide range of technical and professional activities in which you can become active and connected. Opportunities include participating in technical committees or conferences, serving as a publication peer reviewer or author, or volunteering for standards working groups, accreditation visits, or certification activities. There are also groups promoting student contests, in-service training for precollege teachers, computing history, and awards.

The collaborative and technical skills you acquire via CS activities will increase your value to your employer. Few companies provide leadership training internally, and taking on roles in the CS can demonstrate your leadership experience when management considers your capabilities. Cross-boundary interactions, such as your contacts with other professionals in the IEEE/CS, provide insight into prospective customers and suppliers

and offer a potential source of new hires.

When you can associate individuals with the users of your products and services, the quality and ease of use of those products and services improves. And of course this interaction operates both ways. Nine out of ten of my employment advancements were the result of networking via professional activities and workplace colleagues (my initial job out of college was the only transition initiated via a résumé and the human resources department). Please go to the President's Corner at www.computer.org/presidentscorner for links to Society resources that can help you, as well as additional insight on how *being active* can benefit you.

INSTANT COMMUNITIES

A new path for engagement is being developed. We will be providing instant communities in the first quarter of 2010 (<http://communities.computer.org>). As a CS member, you will be able to create a new community, and any individual with an IEEE Web account (membership not required) can join the interaction. For those who are already using similar capabilities such as Yahoo and Google, these communities provide the benefits of a focus on technical discussions with other professionals. Here is a medium to initiate a discussion of Web tools, multicore device design, data mining, security challenges, and so on as you see fit.

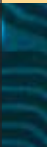
We encourage the development of communities associated with technical tracks at conferences, standards use, papers in publications, emerging technology, precollege contests, policy issues, grant opportunities, and the business of coordinating chapter, student, conference, and other activities. Communities can point to the foundational materials relevant to a specific topic: recent papers, tutorials, webinars, workshops, and so forth—helping interested professionals find

their way to the best resources while avoiding the relevance and quality issues that emerge from Googling keywords.

CAREER CHALLENGES

Communities are one step into the 21st century, and we are taking others. Helping you in your career in challenging economic times coupled with rapid technological change is essential. It is likely that the job you will have in 10 years does not exist

recommender systems like NetFlix to the facial recognition technologies portrayed in the movie *Minority Report*. With tens of thousands of peer-reviewed papers published each year, and many other sources of information as well, it is impossible for an individual to keep track of the rapidly growing body of knowledge. Search helps a bit, and the instant communities will be an additional asset. However, we need to find ways to automatically identify what may



As a Computer Society member, you have the foundation for your future at your fingertips, including opportunities to network at the chapter, conference, and committee meeting levels.

now—consider the role of search optimization 12 years ago, before Google, or 15 years ago, before Altavista.

The way to manage your career to achieve goals that are literally beyond our event horizon is to stay on top of your interests along with emerging technologies. CS publications, conferences, and events are all part of your toolkit. Our Build Your Career Web portal is also focused on this topic (<http://careers.computer.org>). You will find career-oriented articles and educational webinars that help you explore all phases of your career life cycle.

Combine these resources with our free online courses and free access to technical books, and you have the foundation for your future at your fingertips. But don't ignore networking at the chapter, conference, and even committee meeting level. Once you want, or need, a career change, you will waste critical time if you have not already started making these connections.

PERSONALIZATION OPTIONS

Our online future will be dominated by personalization—from

be of value to our members based on both their expressed and implicit choices.

As a not-for-profit organization, the CS has obligations to respect your privacy, and as a professional society, we have ethical standards that apply as well. In short, the Computer Society will be a trustworthy and trusted partner in your professional evolution.

Currently, a company is using AI technology to evaluate a digital library and user selection, and then, based on this information, identify potential job matches. Similar technology is needed to help you find new articles, workshops, webinars, whitepapers, and so on that are of real value to you. A key CS strategic goal in this area is that “the IEEE CS will develop personalized profiles of participating professionals, presenting them with the most relevant information, communities, networking opportunities, information exchanges, and materials”—a journey we have just begun.

Notice that some of the connections we provide may relate to high-quality paid content such as

PRESIDENT’S MESSAGE


webinars or whitepapers. This is a revenue source we can use to reduce our dependence on publication subscriptions, conference fees, and dues. Ultimately, we need to provide a timely flow of information “just for you” that contains only information and pointers highly relevant to your individual needs.

I invite you to join me to help move the Computer Society forward. In addition to the endeavors mentioned here, we have many rewarding projects in our

future. For example, we must find ways to use geographical information systems to remove the invisible borders between colleagues that may be “right next door.” We also need to embrace online publishing, collaboration, and communities to break down information boundaries that interfere with solving the problems that face humanity locally and globally.

Please get involved—with our chapters locally, in your fields of interest, and in our online communities. I encourage you to visit our online

forum, the President’s Blog (www.computer.org/portal/web/cspresident), where you can get a sense of some of the issues we face and also take the opportunity to suggest how we, together, can improve the Computer Society to serve your needs.

I look forward to our year together. 

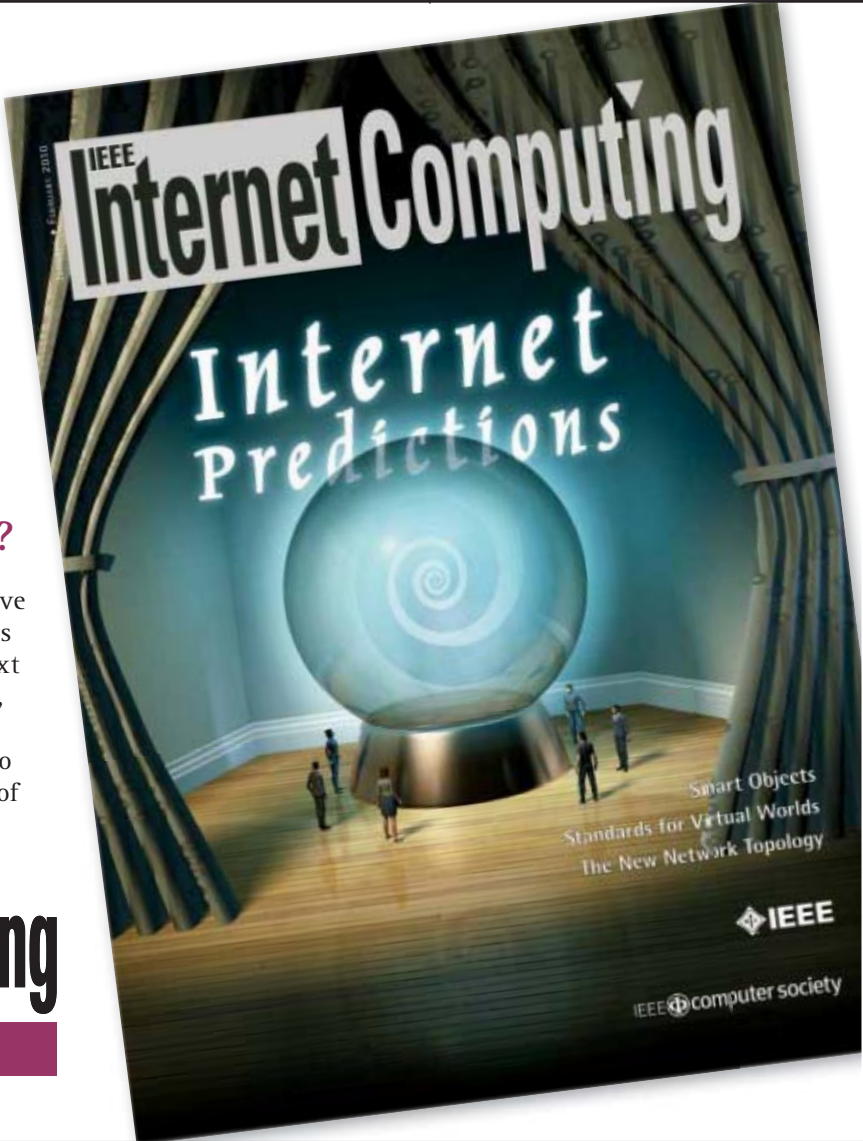
James D. Isaak retired after a 30-year career in industry operating systems and standards and six years in academia. Contact him at cspresident2010@jimisaak.com.

Where Are We Headed?

More than a dozen leading experts give their opinions on where the Internet is headed and where it will be in the next decade in terms of technology, policy, and applications. They cover topics ranging from the Internet of Things to climate change to the digital storage of the future.



www.computer.org/internet/



Designing the Future

➔ David Alan Grier, *George Washington University*



A good technological design requires substantial effort that shapes both the social and technical sides of an artifact.

Dividing the light from the darkness. Pushing back the wilderness to make a place for civilization. Unpacking your worldly goods after a household move. All of these activities require equal amounts of energy. This is a fundamental principle of physics: Newton's fourth law of motion—the conservation of unrealistic goals. Yet, Tamara's voice gave no clue that she was surrounded by chaos.

"Boxes?" I asked. "You are standing in a room filled with boxes?"

"Of course," was her reply.

Tamara and her husband were about to embark on the great organizing task of human experience, that of bringing a baby into this world and nurturing it into adulthood. She is coming to this task later in life and hence has a fairly accurate idea of the effort that it will require. We talked for a bit about how she planned to fit her dissertation research into the two-and-one-half-hour time slots that babies offer their parents.

"I believe I can do it," she said. "Although I wish that I could point to someone who had done this already. It is, after all, a problem of design."

THE INTERACTION BETWEEN TECHNOLOGY AND CULTURE

When I first met Tamara, she was not much interested in design problems or any other engineering tasks. She had studied human communication and was moving to California. "It has the fewest cloudy days in the country," she explained. "That was my criterion for success."

Yet when Tamara began working, she found that many of her clients were involved with digital technology and that their communications problems were concerned with fitting an engineered product into a social and cultural context. Shortly after arriving in her new home, she attended a seminar on innovation that included the entrepreneur Phillippe Kahn as one of the speakers.

"Kahn spent much of his time dwelling on social issues," Tamara recalled. "He kept returning to the question 'What is your consideration of culture?'"

Within the community of Silicon Valley entrepreneurs, Kahn was reputed to have a special insight into the interaction between human nature and engineered artifacts. That community tested such reputations with a scale defined by the values of business. A profitable business suggested a basic apprehension of

social issues. A successful product fell higher on the scale, as it demonstrated a grasp of human habits. An innovative idea that defined a new market, even an idea that was capitalized by others, held the highest place on the scale. Someone who could define a new market, so the reasoning went, must truly understand how human culture interacts with technology.

Kahn's reputation rested on contributions to all three categories of the entrepreneurial scale. He had run a profitable business and had marketed several highly successful products. He was also credited with creating the cell phone camera, thereby defining a new market for communication services. The tale of how he connected a digital camera to a cell phone has become a classic addition to the innovation literature, a story almost as famous as Alexander Graham Bell's urgent message, "Mr. Watson. Come here. I need you."

According to the story, Kahn constructed his cell phone camera in 1997 so that he could send pictures of his newborn daughter to friends and family. Connecting a camera to a digital cell phone was not necessarily innovative. Well before 1997, we knew that any digital device could be connected to any other digital

THE KNOWN WORLD

device, even though the effort might require a substantial amount of programming and the result might not be particularly pleasing. However, the combination of the cell phone and the camera evoked a new set of emotions from people. New fathers, yet untested by late night feedings, looked upon these tiny devices and realized that they could share their moment of sudden transformation with distant family and friends. Adolescents, former newborns themselves, saw a slightly different idea in these new devices, one that recorded their growing independence and connections to nearby friends. The purveyors of photographic services, if they were paying attention, saw a warning in these little devices that looked nothing like a traditional camera.

The cell phone camera has not only changed the way that we record the images of our age, it also has marked a change in visual style. Before the cell phone camera and its many smart siblings, new technology came in many shapes. Most of these shapes were three-dimensional, and a good number were beige in color. As the years have progressed, we have seen the shape of high technology converge to that of the unadorned rectangle. Phones, with or without cameras, are rectangles, as are laptops and desktop computers. Servers are more utilitarian rectangles that slide into frames, which are themselves large rectangles.

THE DESIGN OF THE NEW

It is too easy to dismiss the current shape of high technology as the natural outgrowth of the underlying developments and forget that the design of devices is the product of both the inner nature of the machine and the outer influence of culture. “Form follows function” argued the architect Louis Sullivan, and the culture that uses an artifact ultimately determines its function.

Over the history of the computer, we can point to several machines

that were consciously designed to be distinctive, to look different from common machines. Perhaps the most familiar of these examples is the original Macintosh (1984), but the most dramatic are the Cray-1 (1976) and the Connection Machine 1 (1986). These were both high-performance machines and looked substantially different from the standard boxes of data processing machines. The Cray was a partial cylinder, a little less than six feet tall. Its power supply

It is too easy to forget that the design of devices is the product of both the inner nature of the machine and the outer influence of culture.

occupied a low ring that surrounded the machine and often drew comparisons to second empire furniture. The trade press occasionally referred to it as the “world’s most expensive loveseat,” but the comparison was undermined by a set of aluminum cooling fins.

The Connection Machine was housed in a set of interlocked boxes that attempted to mimic the architecture of the machines. The “company’s president put a high priority on a package that would not only convince viewers of the machines’ uniqueness,” recalled the designer, “but would explain the nature of its architecture.” In fact, technical issues did not dictate the shape of either machine. The architecture of the Connection Machine did not map neatly into its boxes, and the Cray hardware did not take advantage of the circular shape.

LOOKING TO THE FUTURAMA

If we look to the design of earlier computing equipment, we see a steady and conscious effort to

make these devices look modern and to distinguish them from factory machines. IBM began thinking about the physical designs of its products in the 1930s. Company engineers modeled the shape of their accounting machines on Queen Anne furniture. They decided that the machines would have curved legs even though straight legs would have been cheaper to produce.

As IBM started to build more complicated computing machines, it hired professional designers to determine the outward appearance of its products. The most influential of these designers was Norman Bel Geddes. Bel Geddes had started as a set designer for Broadway plays and had established his reputation as an industrial designer by creating the General Motors exhibit at the 1939 World’s Fair. Known as the Futurama, the exhibit showed fairgoers the world of 1960, including massive skyscrapers and high-speed freeways. “Each day of the fair,” wrote one observer, “thousands of visitors waited for hours in lines up to a mile in length for the opportunity to experience the Futurama.”

To convey the idea that an object was new or modern, Bel Geddes liked to employ the curved shapes that were found on airplanes, a concept known as *streamlining*. The curved shapes suggested not only newness, but also speed, power, and the conquest of nature. He applied streamlining to buildings, automobiles, and even to household appliances.

In 1943, Bel Geddes designed a streamlined shell for Howard Aiken’s Mark I calculating machine, which had been built by IBM. He created a shape that had curved corners, brushed aluminum panels, and brightly lit windows. “It gave poor Howard Aiken an awful pain, because it was fifty or a hundred thousand bucks for the case,” recalled one worker. Aiken would rather have invested that money in the machine “and that irked him.”

THE MODERN STYLE

During the 1950s, the streamline style of Bel Geddes merged with the minimalist ideas of Elliot Noyes. Noyes was a junior designer in Bel Geddes' office and an army buddy of Thomas Watson Jr. His friendship with Watson helped Noyes win contracts to design the shape of an IBM electric typewriter and the décor in Watson's office in the IBM building. He "stripped away the walnut panels and heavy curtains" of the office, reported one magazine, "replacing them with large sheer planes of color, and installing works of modern art throughout."

Like Bel Geddes, Noyes liked to create both objects and the spaces that were used to display those objects. He conceived the idea that computer machine rooms were actually display areas. He designed machine rooms for IBM that were clean, white, and marked by a rectangular grid. He wanted nothing in the room to interfere with the opportunity to view the machines. "If you get at the heart of the matter," Noyes wrote, "what IBM really does it to help man extend his control over the environment."

For the computers themselves, Noyes stipulated that they would be housed in simple white boxes with minimal decoration. As well as any IBM engineer, he knew that the computers did not naturally fit into rectangular boxes, but he did not want his design to give any hint that the IBM products were in any way mechanical. They would have no visible fans, no moving parts beyond the tape drives, not even a smell of lubricating oil if that could be hidden.

From his position at IBM, Noyes influenced the entire industry. Some vendors, such as Westinghouse, hired him to design their machines. Others, such as Burroughs and NCR, copied his designs. He injected his ideas into popular culture through the 1968 movie *2001: A Space Odyssey*. Noyes served as artistic designer for that movie and created spaceship interi-

ors that looked like IBM computer rooms. These interiors had nothing in common with the spacecraft of the age, with their switches, dials, and utilitarian colors. They had bright white walls, undecorated surfaces, grid floors, and even Herman Miller chairs like the ones Noyes liked to purchase for IBM offices.

The computers of the 1960s did not look futuristic because digital technology made them look that way. They were presented in clean, simple designs because Eliot Noyes believed that such a design suggested the future of computing technology. Noyes died in 1977, but he would appreciate the current design of computers. Black and silver rectangles. No wires, no buttons. That is the way the future should look.

Tamara's future will begin in three weeks, when her firstborn son is scheduled to arrive. We hope that we will be able to continue our conversations, but such a goal will likely vanish in the presence of her new responsibilities. I don't know how she is preparing her new home in anticipation of the baby's arrival,

but I suspect that she will be fully aware of the messages that her choice of decor will communicate. We cover the walls of nurseries with primary colors and images of characters owned by the Disney Corporation to signify that a baby lives in that room, a baby that will spend most of his or her life in the future.

Of course, a baby does not see the nursery decor as representing the future. If anything, children eventually come to associate the design of their nursery with the past, with the time when they were young, immature, and helpless. Eventually, they will demand that their room be redecorated with new colors and pictures of pirates, princesses, software engineers, or something that points to their future, their hope to make a name for themselves, to push back the wilderness, to overcome the law of unrealistic goals. **■**

David Alan Grier is the former editor in chief of the IEEE Annals of the History of Computing. The current issue has nothing about the future but much about the past of databases and database software. Contact him at grier@gwu.edu.



Silver Bullet Security Podcast

In-depth interviews with security gurus. Hosted by Gary McGraw.

www.computer.org/security/podcasts

Sponsored by 

32 & 16 YEARS AGO

JANUARY 1978

PRESIDENT’S MESSAGE (p. 7) “... You have ratified the new IEEE Computer Society Constitution. The vote was 6398 for and 166 against. The principal changes relate to the membership electing the society’s officers and Governing Board. You have a new challenge to help select your leadership, and we have a new and difficult task to present you with candidates, qualifications, issues, and positions to help make the selection process meaningful to you. The new constitution also increases the participation on the Governing Board, assuring 20 elected positions plus four officers.”

DISTRIBUTED PROCESSING (p. 13) “At least four physical components of a system might be distributed: hardware or processing logic, data, the processing itself, and the control (such as the operating system). Some speak of a system that has any *one* of these components distributed as being a ‘distributed data processing system.’

“However, a definition that is based solely on the physical distribution of some components of the system is doomed to failure. A proper definition must also cover the concepts under which the distributed components interact. ...”

DISTRIBUTION RESEARCH (p. 28) “The Honeywell Experimental Distributed Processor (HXDP) is a vehicle for research in the science and engineering of processor interconnection, executive control, and user software for a certain class of multiple-processor computers which we call ‘distributed computer’ systems. Such systems are very unconventional in that they accomplish total system-wide executive control in the absence of any centralized procedure, data, or hardware. The primary benefits sought by this research are improvements over more conventional architectures (such as multiprocessors and computer networks) in extensibility, integrity, and performance. A fundamental thesis of the HXDP project is that the benefits and cost-effectiveness of distributed computer systems depend on the judicious use of hardware to control software costs.”

NETWORK DISTRIBUTION (p. 48) “The last decade has seen the rapid evolution of computer communication networks from research curiosities to operational utilities. The Arpanet, for example, currently supports communication among more than 100 computer systems and is used daily by hundreds of users. Commercial networks, such as Telenet in the United States and Datapac in Canada, have very bright futures. One attraction of these networks is their ability to provide access to a wide variety of resources distributed among the connected computers.”

INTERACTIVE COMPUTER GRAPHICS (p. 60) “After 15 years of development, much promise, but limited use, is interac-

tive computer graphics finally due for a sharp growth spurt? If the mood projected at the Fourth Annual Conference on Computer Graphics and Interactive Techniques held in San Jose

recently is any indication, the answer would seem to be yes. On the hardware side, costs are coming down and capability is going up, reflecting trends that have characterized the computer industry in recent years. On the software side, although programming for computer-aided design tends to be specialized and hence not broadly available, many standardized display packages are on the market. In applications, there is activity in more than 25 areas, ranging from stereotaxic surgery to landfills after strip mining.”

MAGNETIC BUBBLES (p. 82) “Data Systems Design has introduced the first mass storage system using magnetic bubble technology.

“The new DSD 640 is a DEC-compatible, nonvolatile memory system designed to replace floppy disks in harsh environments. It has an average access time of 4 msec compared to an average access time of 313 msec for a floppy disk (including latency). Maximum access time of the DSD 640 is 7.2 msec.”

MAGNETORESISTANCE (p. 92) “At Philips Research Laboratories extensive investigations have been carried out in recent years into the use of thin layers of magnetoresistive material for the construction of a thin-film or integrated read head. A miniature read head of this type is deposited by vacuum evaporation on a silicon slice, in a manner based on the thin-film processes used for integrated circuits. IC technology makes it possible to deposit a large number of thin-film heads of very small dimensions on a single silicon substrate. A multitrack read head produced in this way can be used for readout from tapes or disks with an extremely high information density. Another feature of the silicon substrate is that it can accommodate the electronic circuitry required for parallel readout, thus fully justifying the use of the term ‘integrated read head’.”

PDFs of the articles and departments from the January 1978 and 1994 issues of Computer are available through the IEEE Computer Society’s website: www.computer.org/computer.



JANUARY 1994

PRESIDENT’S MESSAGE (p. 6) “... Software still seems to be the most challenging area, the one with the greatest unmet potential. But significant hardware challenges remain as well—for example, maintaining external interfaces (as much as it’s reasonable to do so) and improving function and performance within those parameters. The National Research Council publication *Computing the Future* (National Academy Press, Washington, DC, 1992) discusses major unmet challenges in the computing environment today. One is providing unimaginable amounts of data upon user request, wherever that user is located and wherever the data is located. Another is providing a magnitude improvement in computing power—at the same cost as or less than today’s capability.”

PREDICTABILITY (p. 24) “It is tempting to think that speed (for example, processor speeds or higher communication bandwidths) is the sole ingredient in meeting system timing requirements, but speed alone is not enough. Proper resource-management techniques also must be used to prevent, for example, situations in which long, low-priority tasks block higher priority tasks with short deadlines. One guiding principle in real-time system resource management is *predictability*, the ability to determine for a given set of tasks whether the system will be able to meet all of the timing requirements of those tasks. Predictability calls for the development of scheduling models and analytic techniques to determine whether or not a real-time system can meet its timing requirements.”

HARDWARE/SOFTWARE CODESIGN (p. 42) “In traditional embedded system design, system architects decide which parts of the system will be in hardware or software. For certain operations the decision is clear-cut: high-speed packet manipulation, for example, will be implemented in hardware, and a recursive search will be software-based. However, some operations can be implemented in hardware or software or both: these operations are called hardware/software codesign operations.

“Effective partitioning of codesign operations into hardware and software depends on many factors, including performance, cost, maintainability, flexibility, and size. Ideally, an automated codesign system would provide a variety of partitions so that system architects could choose the best solution for their requirements.”

REPROGRAMMABLE COMPONENTS (p. 48) “Recent advances in the design and synthesis of integrated circuits have prompted system architects to investigate computer-aided design methods for systems that contain both application-specific and *predesigned* reprogrammable components. Although a reprogrammable microprocessor

like the Mips R3000 can implement most system functionality as a program, dedicated application-specific integrated circuits (ASICs) are needed for performance reasons. In this context, recent advances in ASIC synthesis and the proliferation of advanced and inexpensive processors have stimulated interest in hardware/software codesign.”

VIRTUAL ROBOTS (p. 80) “A British firm is offering a software package for personal computers that enables designers to ‘test drive’ a robot before it is built. Robot Simulations, Newcastle-upon-Tyne, Tyne & Wear, England, says that its Workspace 3 software facilitates the design and simulation of robot work cells. Users can choose from more than 100 robot models and can even simulate their control languages and interfaces. The simulations are displayed in full-color Super VGA 3D graphics. Real-time animation of a work sequence allows time cycles to be evaluated and problems to be resolved before equipment (or personnel) is placed at risk.”

INFORMATION NETWORKING (p. 81) “The Internet’s original users were scientists and engineers in academic and corporate settings who required remote computing for their government-funded research and development. Over the last five years the types of users and Internet applications have changed radically. Researchers, educators, and students from a host of educational institutions and research organizations now use the Internet. Moreover, the contemporary Internet population spends much more time using the Internet for communication and publication than for computation per se. In fact, as well as in vision, *information networking* has supplanted computer networking as the primary reason for expanding the Internet and for building other high-performance, fully digital networks.”

SUPER HARVARD ARCHITECTURE (p. 94) “Analog Devices announced a new class of single-chip digital signal processors (DSPs), the first to use Super Harvard Architecture, according to the company.

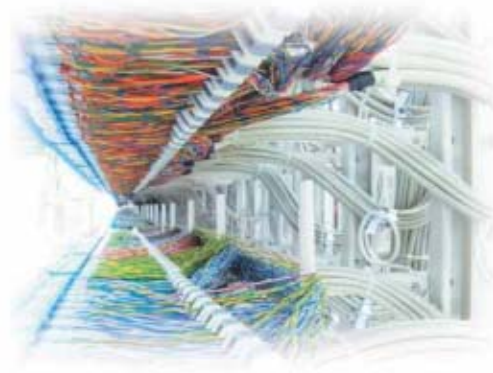
“The ADSP-21060 is an IEEE single-precision floating-point and 32-bit fixed-point DSP. The chip integrates an ADSP-21020 floating-point core processor with three arithmetic computation units (ALU, multiplier, and barrel shifter), 10 channels of multiported DMA, a 4-Mbit dual-ported SRAM memory, and an external parallel port. Dual-data-address generators with indirect, immediate, modulo, bit-reverse addressing, and program sequencing with zero-overhead looping contribute to the DSP’s throughput level. It achieves 40-MIPS performance with a 25-ns instruction rate.”

Editor: Neville Holmes; neville.holmes@utas.edu.au

TECHNOLOGY NEWS

Is 3D Finally Ready for the Web?

➔ Sixto Ortiz Jr.



3D content still is not widely found on the Web. Now, though, several new technologies may widen 3D's presence on the Web by transforming browsers into computing platforms powerful enough to play the content.

From its humble beginnings as a point-and-click environment, today's Web is a dazzling collection of pages filled with all types of applications for both entertainment and productivity.

Users can accomplish many tasks on today's Web, from purchasing products to interacting in real-time with users throughout the world.

However, one key element has yet to make its mark on the Web: 3D.

Today, 3D is primarily used online in applications such as games and virtual worlds, which are rendered using powerful computers and specialized software.

However, businesses, engineering firms, and other users also want the realism and additional detail that 3D adds, noted D.J. Edgerton, CEO and cofounder of Zemoga, a graphics design and marketing firm.

Users want their browser-based experiences to be more like those they have on a PC.

Consumers are becoming more accustomed to 3D content because of the use of the technology in movies, videogames, and other types of entertainment, said David Laubner, director of product marketing at Dassault Systèmes' 3DVIA, a vendor of 3D development tools.

There is thus demand for more and easier-to-access 3D content on the Web, said Antonio Collier, founder and CEO of Vzillion, which designs virtual environments.

And the better the browser experience, the more potential revenue that online content could generate for providers and others.

However, 3D on the Web remains primitive today because the complex technology has been difficult to use with typical PCs and browsers, said David Gardner, founder and CEO of the Venue Network, which developed the VenueGen 3D Web-based conferencing application.

In fact, browsers generally cannot natively run complex 3D content or offer either high frame rates or full-screen graphics, noted Joshua Smith, chief technology officer and cofounder of Kaon Interactive, a company that creates and develops interactive 3D product models.

Including 3D in real-time collaboration programs and other applications complicates already complex development processes, he added.

Now, though, several organizations are working on technologies that may finally widen 3D's presence on the Web by transforming browsers into more powerful computing platforms

that can deliver a PC-like experience, including the playing of 3D content.

This would enable applications such as product modeling, presentation, and configuration; 3D online meetings and worker collaboration; the simulation of processes such as surgery or mechanical procedures; virtual tours; and augmented reality.

Nonetheless, 3D on the Web will have to clear some obstacles before the technology can become reliable and mainstream.

3D on the web

The early Web ran without graphics, but that changed when the US National Center for Supercomputing Applications released Mosaic—the first browser able to display images along with text—in 1993.

There have been several technologies for 3D on the Web that basically work the same but use different file formats.

VRML and X3D

3D on the Web began when the VRML Consortium released the Virtual Reality Markup Language in 1994.

However, VRML never really caught on because it let developers write only 3D content, said Kaon's Smith.

To create full, compelling applications, he explained, developers must be able to write 3D, 2D, video, and audio content together.

Also, VRML appeared well before processors and software could support the graphics that the technology enabled, noted Eric Brown, president of [Saugus.net](#), a website design firm, and a Boston University lecturer.

And, 3DVIA's Laubner added, VRML was "too slow and incapable of rendering complex, high-fidelity models and scenes."

In 1997, the Web3D Consortium released X3D—an XML-based file format for representing 3D graphics that includes VRML extensions. Like VRML, Smith said, X3D has not really caught on.

According to Laubner, the gaming and interactive-3D developer communities have largely ignored X3D, which is supported by few commercial tools.

Other approaches

The 3D Industry Forum's Universal 3D technology, released in 2003, is a compressed file format for 3D graphics. However, proponents have promoted Universal 3D as a file format that will be used primarily in applications such as manufacturing and construction.

The open source 3D Markup Language for Web is an XML-based file-format for creating 3D and 2D content on the Web. 3DMLW, which 3D Technologies R&D released last year, works with most popular Web browsers via plug-ins.

TECHNICAL DEVELOPMENTS

Today's hardware is better able to produce 3D content than in the past. For example, faster CPUs, graphics processors, and video cards, as well as more pervasive 3D graphics accelerators, are contributing to the emergence of 3D on the Web.

JavaScript and HTML 5.0

Performance improvements in browser engines that process Java-

Script, the language that developers use to write many Web-based applications, have helped bring 3D to the Web.

For example, the engine in Internet Explorer 9 will be able to use the host system's graphics processor to perform graphics-related tasks quickly.

Also, Mozilla's enhancements to Firefox's JavaScript engine used a technique called tracing—which optimizes the way code is run—to improve performance.

And JavaScript's ability to access HTML 5.0's many capabilities lets developers combine video, audio, 3D, and 2D into one seamless application.

scenes from simple elements, called *primitives*, such as lines and polygons. OpenGL ES (embedded systems) works in small devices such as smart phones.

However, these capabilities cannot be implemented in a browser without plug-ins.

Many users prefer not to use plug-ins, finding them inconvenient to install, troubleshoot, and manage, said Vladimir Vukicevic, Mozilla's principal engineer for the Firefox browser.

WebGL lets browsers render 3D content without a plug-in. The technology extends OpenGL by providing

New technologies may promote 3D on the Web by improving browser capabilities.

The introduction of the *canvas* element to HTML 5.0 will also enable 3D on the Web, predicted Tim Johansson, Opera Software's core developer for the company's browser.

This element lets browsers, via their JavaScript engines, natively and dynamically render bitmap images, which makes it easier to display 3D content without plug-ins.

WebGL

The Mozilla Foundation—a group that creates and supports open source applications—and the Khronos Group—an industry consortium that designs standards for parallel computing, graphics, and dynamic media—are developing WebGL.

The technology brings hardware-accelerated 3D graphics to the Web without plug-ins. WebGL will work in any browser that supports Khronos' OpenGL (originally the Open Graphics Library) or OpenGL ES specification.

The cross-language, cross-platform OpenGL defines an API for writing applications that produce 2D and 3D computer graphics.

The specification provides programming tools for drawing 3D

an API that lets software programmatically access a PC's 3D-rendering hardware.

In essence, WebGL allows communication between JavaScript applications and the OpenGL software libraries, which access the host system's graphics processor. This enables use of the hardware's full capabilities to render 3D content.

Khronos has established a WebGL Working Group, which is slated to deliver the technology's first public release in the first half of this year.

But programmers are already building WebGL into developer versions of Firefox and the WebKit open source browser engine, used with both Apple's Safari and Google's Chrome.

O3D

Google, which participates in the WebGL Working Group, has developed a 3D graphics technology for browsers called O3D, which Figure 1 shows.

O3D is a plug-in for the Internet Explorer, Firefox, Safari, and Chrome browsers. Google is now building the technology into Chrome and hopes it

TECHNOLOGY NEWS

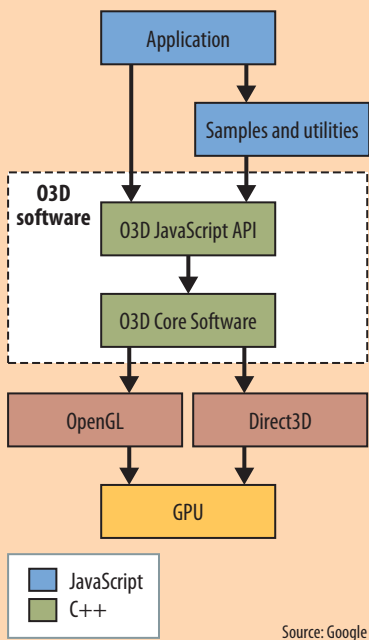


Figure 1. A JavaScript application can run via O3D plug-in software, which includes a JavaScript API and the Core Software. The Core Software taps directly into a host computer's graphics hardware—for better performance—via the OpenGL or Microsoft's Direct3D graphics API. JavaScript utilities provide sample code to help with common tasks.

eventually will be built directly into other browsers, too.

O3D—which works with Windows, the Mac OS, and Linux—is an open source JavaScript API for developing interactive 3D graphics applications—such as games, advertisements, and virtual product tours—that run within a browser.

The API provides an interface for JavaScript-based programs. It lets a JavaScript application talk to the O3D Core Software, contained in the O3D plug-in, to tap directly into a computer's graphics hardware.

Google spokesperson Eitan Bencuya said O3D is a retained-mode technology, which sets up a scene once, then draws only the changes necessary for each frame.

Unlike immediate-mode technologies such as WebGL, O3D doesn't redraw the entire scene every time.

This provides better performance but less developer control, Bencuya noted.

Adobe Flash

Adobe is adding better 3D capabilities to its proprietary Flash browser plug-in.

Adobe introduced 3D capabilities with the release of Flash Player 10 in 2008.

This technology incorporated support for 3D effects via the addition of new classes and methods—particularly the ability to specify an object's position in three dimensions—to the ActionScript programming language for Flash, said Tom Barclay, Adobe's senior manager for the Flash Player.

With this approach, developers without much 3D experience can author 3D content by designing objects in 2D and modifying them via the new classes and methods, he explained.

Flash Player 10.1, now in beta, will bring 3D effects to smart phones and other mobile devices, he added.

STANDING IN THE WAY

Before the 3D Web can truly flower, it must overcome numerous hurdles.

For example, plug-ins, used in some approaches, occasionally cause browser crashes and other problems.

Vzillion's Collier said the need for browsers to natively render 3D content and the current inability of 3D technology in general to work with all browsers, operating systems, and application types are challenges.

The lack of standardization is also an issue, he added.

If standardization doesn't occur, said independent 3D designer Lane Force, the Web will wind up with numerous incompatible formats and technologies, forcing developers to create multiple versions of content to run on different browsers.

Also, 3D on the Web entails long authoring times, and relatively few developers are familiar with the approach.

The upsurge in the use of netbooks and smart phones, which have slower processors, means more people are using devices that can't run 3D content, noted Kaon's Smith.

Added Zemoga's Edgerton, even though hardware capabilities and network bandwidth have increased significantly, they're still not enough for presenting highly complex 3D models.

And proponents aren't designing online 3D technology for the average user, which is the same mistake that occurred with VRML, said William Hurley, founder of whurleyvision LLC, an augmented-reality consultancy.

WebGL is the most interesting development for 3D on the Web because it doesn't require plug-ins, according to Opera's Johansson.

Once WebGL is released, he said, "We will see a large increase in the amount of 3D content on the Web."

3D won't really take off for about 10 years, though, until the increasingly popular netbooks and smart phones gain the processing power to play the data-intensive content, said Kaon's Smith.

Eventually, though, he predicted, 3D on the Web will do as well as video on the Web has done. **C**

Sixto Ortiz Jr. is a freelance technology writer based in Spring, Texas. Contact him at sortiz1965@gmail.com.

Editor: Lee Garber, Computer,
l.garber@computer.org

cn Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.

NEWS BRIEFS

Company Releases Programmable, Versatile, Open Source Data Glove



A company has released a programmable data glove that records users' hand and finger motions to let them interact in novel ways with computer systems.

Jack Vice, president, chief technical officer, and cofounder of AnthroTronix, said the open source AcceleGlove can be used for three types of functions: command and control of games, robots, other devices, and computer applications; communications such as hand signals and sign language; and motion analysis like that used in sports training and physical therapy.

The product includes a software developer's kit that lets programmers use Java to write applications for the glove.

The AcceleGlove uses six sensitive, 3-axis accelerometers—one on each finger and one on the back of the hand—to capture the hand's rate of motion and even slight 3D changes in position and orientation, explained Vice.

They feed position information through lightweight copper wires to a printed circuit board on the back of the glove. The system then transmits the information to a PC, laptop, or handheld device via a USB cord plugged into the glove. The system also receives power through the cord, avoiding the need for a cumbersome battery pack. The glove's software then processes the information and makes the necessary commands to initiate the desired actions.

The API used to develop applications for AcceleGlove includes a

predefined library of recognized gestures and resulting functions and commands. The system can also learn new gestures and assign new commands to them through training.

AcceleGlove can store about 1,000 gestures, although most users will not need more than 100, according to Vice. The system can work with data from a pair of gloves used in tandem by a single user.

AnthroTronix initially developed the glove with the US Department of Defense to let users control tactical military robots and to capture arm and hand signals.

AnthroTronix says its AcceleGlove costs \$649 and thus is considerably less expensive than similar data

gloves, which typically cost between \$1,000 and \$5,000. AcceleGlove is less costly, Vice explained, because it works with accelerometers, rather than the more expensive, less accurate sensors that its competitors use.

AnthroTronix plans to add an arm-tracking component to the glove within five years. This would provide more types of motions to track, thereby enabling more functionality. AnthroTronix has already developed a prototype that wraps around the user's biceps and triceps. **C**

News Briefs written by Linda Dailey Paulson, a freelance technology writer based in Portland, Oregon. Contact her at ldpaulson@yahoo.com.



AnthroTronix has released a programmable data glove that records users' hand and finger motions to let them interact in novel ways with computer systems. In this case, the user is using the AcceleGlove to play a game that requires communication via sign language.

NEWS BRIEFS

Vendors Accelerate Computer Bootups

Various PC-industry vendors are working on ways to reduce the long bootups that irritate many computer users.

Many users want their computers to start up instantly, like their telephones or consumer-electronics products, said Brian Richardson, senior technical marketing engineer with American Megatrends, which makes basic input/output systems.

Enabling this has become a goal for BIOS and OS developers, as PC bootup entails both BIOS and OS startup.

The BIOS—firmware stored on a small piece of nonvolatile RAM attached to the main chipset—is the first piece of code a computer runs when powering up. It tests and starts up system devices such as the RAM, hard drive, video display card, and keyboard.

The BIOS manages the preboot data flow between the OS and the system hardware, determines whether all peripherals are in place and operational, and then loads the OS into the computer's memory from the hard drive.

BIOS bootup can last up to 20 seconds for a PC and up to several minutes for a server, depending on the host's configuration, Richardson said. For example, the process takes longer for systems with more memory and

add-on cards because the BIOS must allocate memory and I/O resources.

The United EFI (Extensible Firmware Interface) Forum has developed the UEFI standard to replace today's BIOS. Unlike BIOS, which works only with Intel's x86 processors, UEFI would not be tied to a single chip architecture.

UEFI would improve the BIOS's intelligence so that it needn't perform all checks every time a PC powers up. Most users don't upgrade their hardware often, if at all, Richardson explained. UEFI could recognize if the hardware configuration hasn't changed and then use the system's previous configuration, without rerunning hardware tests.

Vendors would have to adapt hardware to support UEFI, said Nik Simpson, senior analyst with the Burton Group, a market research firm, but aren't rushing to do so because BIOS technology is still functional.

OS bootup typically takes up to several minutes, depending on various factors such as the nature and number of operating system features and the number of drivers that load on start-up.

The OS doesn't make every check the BIOS makes but instead performs operating-system-specific system checks. These include loading the OS, initializing hardware, loading the network stack and Desktop Window

Manager, and providing the logon prompt.

Microsoft says it is working to reduce Windows' startup times but didn't answer specific questions about this. However, in the company's E7 (Engineering Windows 7) blog, engineer Michael Fortin said that an entire team is working on start-up performance and that its goal is for the OS to boot in less than 15 seconds.

Microsoft made Windows 7 boot faster by optimizing the order in which it initializes OS functions, said Simpson. In addition, he explained, Windows 7 starts only those services needed to present a usable desktop to the user, such as the Desktop Windows Manager. Afterward, it starts other functions in the background.

And, he added, Windows includes the ReadyBoost system introduced in Vista. For faster startup, the system utilizes a USB flash drive to store copies of important OS components and frequently used applications. Thus, for Windows 7 machines, the combined BIOS and OS start time is about 45 seconds.

With its new Latitude Z laptop, Dell offers instant bootup for users who just want to check the Web or their e-mail, calendar, or contacts. The Latitude On mode comes with its own power button and lets the computer boot from a special, Linux-based chipset, bypassing the main OS.

Simpson said BIOS and OS improvements will likely eliminate some time from bootup. This may provide some competitive advantage, he predicted, but won't be "a game changer." □

Join the IEEE
Computer Society

www.computer.org

Editor: Lee Garber, *Computer*, l.garber@computer.org

➔ TWO FIRMS DEVELOP WORLD'S SMALLEST MEDICAL CAMERA

Medical diagnoses frequently necessitate invasive, costly, and potentially risky surgical procedures, x-rays, or scans. However, two Israeli companies have developed the world's smallest medical camera, which could eliminate the need for these approaches in some cases.

Medigus, which develops endoscopic devices and procedures, and Tower Semiconductor have made the IntroSpicio 120, a tiny, inexpensive, disposable, color video camera.

Avi Strum, vice president and general manager of Tower's Specialty Business Unit, said each camera sensor is 0.7 millimeters square. The camera, with its housing, measures 1.2 × 5 mm. The device's complementary metal-oxide semiconductor (CMOS) image sensor—which receives light and translates it to electrical current—has nearly 50,000 2.2-micron pixels and four wires for external connections to devices such as a computer monitor.

Both Medigus and Tower declined to discuss the techniques they used to build such a small camera.

The cameras fit in disposable endoscopes so that doctors can view inside some of the body's narrowest channels, such as small blood vessels. The camera would yield actual pictures of the body, instead of just x-ray or scan images.

According to Medigus business development manager Lior Lurie, the devices will be particularly useful for operations that could benefit from surgical cameras—such as removing stones from ducts leading to the gallbladder—but don't use them because they have been too large until now.

Lurie said the cameras could also be used for nonmedical applications such as visual inspections of very small areas.

Users could integrate the cameras into existing devices or create custom implementations for them.

Once in mass production, Strum said, each camera should cost about \$10, which would make disposing of the product less expensive than sterilizing nondisposable cameras for reuse.

Medigus also plans to make small quantities of reusable IntroSpicio 120s available.

Strum said that mass distribution of the camera is unlikely before the end of 2010 because of the government approvals needed. However, he noted, sample devices have been sent to customers for evaluation and procedure development.

Both companies developed the CMOS sensor, which Tower manufactured. Medigus developed the remainder of the system and assembled all of its elements.



Two Israeli companies have developed the world's smallest medical camera, the IntroSpicio 120, shown here next to a US penny.

New Lithography Approach Promises Powerful Chips

Purdue University researchers are working on a new nanolithography approach that promises to enable the creation of fast chips with extremely small feature sizes, an ongoing industrywide challenge.

Photolithography—used for making modern computer chips—projects light with ever smaller wavelengths through one or more masks to draw finer and finer circuit patterns on light-sensitive photoresists on chip substrates. The circuit paths are then etched into the substrate. As the resulting circuits become smaller, more of them can fit on a chip, enabling the processor to run faster.

The Purdue approach works with a form of extreme ultraviolet lithography, which uses light with very short wavelengths to print tiny circuits on chips.

EUV will become important because current lithographic techniques will reach their theoretical limits for making circuitry smaller in the not-too-distant future, explained Purdue professor Ahmed Hassanein, director of the school's Center for Materials under Extreme Environments.

Currently, most advanced chips are using 45-nm feature sizes, and vendors are working on chips with 32-nm circuit patterns, noted Nathan Brookwood, analyst with

Insight 64, a market research firm.

The industry is now experimenting with deep ultraviolet (DUV) lasers, with wavelengths of 193 to 248 nm, theoretically capable of drawing circuit patterns as small as 20 nanometers on a silicon wafer, according to Brookwood. EUV light has a wavelength of 13.5 nm and thus could yield even smaller features sizes.

Researchers have been working on EUV technology for about 15 years, Brookwood said, but have faced numerous challenges, including finding an effective way to generate the high-energy photons required for short-wavelength light.

The Purdue researchers heated xenon, tin, or lithium to create a

NEWS BRIEFS

plasma, which conducts electricity. The resulting magnetic field in the plasma helps shape photons into small-wavelength EUV light.

A key problem, Hassanein noted, is that most matter absorbs EUV radiation, which would hurt the Purdue technique's ability to focus light on the substrate.

The EUV lithography process thus must take place in a vacuum with

optical molybdenum-silicon elements that won't absorb the needed photons. This would make the process more expensive, Brookwood said.

DUV uses refractive lenses to focus and narrow the light to draw small feature sizes. However, refractive lenses absorb much of the EUV light, so the Purdue system uses reflective mirrors, Brookwood noted. But even the mirrors absorb between 20

and 30 percent of the light, added Hassanein.

The Purdue approach looks promising, said Brookwood.

However, Hassanein explained, his team must show the industry that its technique is reliable even in high-volume manufacturing, that the equipment is durable and cost-effective, and that all aspects of the chip-making process are in place. **C**

stay on the

Cutting Edge

of Artificial Intelligence



IEEE Intelligent Systems provides peer-reviewed, cutting-edge articles on the theory and applications of systems that perceive, reason, learn, and act intelligently.

The #1 AI Magazine

www.computer.org/intelligent

IEEE Intelligent Systems

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE: www.computer.org

OMBUDSMAN: To check membership status or report a change of address, call the IEEE Member Services toll-free number, +1 800 678 4333 (US) or +1 732 981 0060 (international). Direct all other Computer Society-related questions—magazine delivery or unresolved complaints—to help@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION: To obtain more information on any of the following, contact Customer Service at +1 714 821 8380 or +1 800 272 6657:

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

Computer: The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals: The society publishes 13 magazines, 18 transactions, and one letters. Refer to membership application or request information as noted above.

Conference Proceedings & Books: Conference Publishing Services publishes more than 175 titles every year. CS Press publishes books in partnership with John Wiley & Sons.

Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Committees: TCs provide professional interaction in more than 45 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers two software developer credentials.

For more information, visit www.computer.org/certification.

EXECUTIVE COMMITTEE

President: James D. Isaak*

President-Elect: Sorel Reisman*

Past President: Susan K. (Kathy) Land, CSDP*

VP, Standards Activities: Roger U. Fujii (1st VP)*

Secretary: Jeffrey M. Voas (2nd VP)*

VP, Educational Activities: Elizabeth L. Burd*

VP, Member & Geographic Activities: Sattupathu V. Sankaran†

VP, Publications: David Alan Grier*

VP, Professional Activities: James W. Moore*

VP, Technical & Conference Activities: John W. Walz*

Treasurer: Frank E. Ferrante*

2010–2011 IEEE Division V Director: Michael R. Williams†

2009–2010 IEEE Division VIII Director: Stephen L. Diamond†

2010 IEEE Division VIII Director-Elect: Susan K. (Kathy) Land, CSDP*

Computer Editor in Chief: Carl K. Chang†

* voting member of the Board of Governors † nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 2010: Pierre Bourque; André Ivanov; Phillip A. Laplante; Itaru Mimura; Jon G. Rokne; Christina M. Schober; Ann E.K. Sobel

Term Expiring 2011: Elisa Bertino, George V. Cybenko, Ann DeMarle, David S. Ebert, David A. Grier, Hironori Kasahara, Steven L. Tanimoto

Term Expiring 2012: Elizabeth L. Burd, Thomas M. Conte, Frank E. Ferrante, Jean-Luc Gaudiot, Luis Kun, James W. Moore, John W. Walz

EXECUTIVE STAFF

Executive Director: Angela R. Burgess

Associate Executive Director; Director, Governance: Anne Marie Kelly

Director, Finance & Accounting: John Miller

Director, Information Technology & Services: Carl Scott

Director, Membership Development: Violet S. Doan

Director, Products & Services: Evan Butterfield

Director, Sales & Marketing: Dick Price

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036

Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614

Email: hq.ofc@computer.org

Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314

Phone: +1 714 821 8380

Email: help@computer.org

Membership & Publication Orders:

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641

Email: help@computer.org

Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan

Phone: +81 3 3408 3118 • **Fax:** +81 3 3408 3553

Email: tokyo.ofc@computer.org

IEEE OFFICERS

President: Pedro A. Ray

President-Elect: Moshe Kam

Past President: John R. Vig

Secretary: David G. Green

Treasurer: Peter W. Staecker

President, Standards Association Board of Governors: W. Charlston Adams

VP, Educational Activities: Tariq S. Durrani

VP, Membership & Geographic Activities: Barry L. Shoop

VP, Publication Services & Products: Jon G. Rokne

VP, Technical Activities: Roger D. Pollard

IEEE Division V Director: Michael R. Williams

IEEE Division VIII Director: Stephen L. Diamond

President, IEEE-USA: Evelyn H. Hirt

Next Board Meeting:

5 Feb. 2010, Anaheim, CA, USA



revised 30 Nov. 2009

COVER FEATURE



PROJECT GREENLIGHT: OPTIMIZING CYBER- INFRASTRUCTURE FOR A CARBON- CONSTRAINED WORLD

Larry Smarr, *University of California, San Diego*

Even with a variety of aggressive energy efficiency measures, the ICT sector’s carbon emissions will nearly triple from 2002 to 2020. We must accelerate ICT energy efficiency so that we can increase the use of ICT in smart infrastructure capable of reducing global greenhouse gas emissions.

Everyone is now aware of the growing threat of global climatic disruption, but it’s less well known that our information and communication technology (ICT) community can play a key role in this looming crisis. The Climate Group, on behalf of the Global eSustainability Initiative (GeSI)—a consortium of major IT and telecommunications companies—recently issued an informative new study, *Smart 2020: Enabling the Low Carbon Economy in the Information Age*, on this topic (www.theclimategroup.org). This report argues that in addition to making ICT systems more energy efficient, application of those systems to electricity grids, logistic chains, intelligent transportation, and building infrastructure could reduce global greenhouse gas (GHG) emissions by as much as 15 percent by 2020, compared with business as usual. This could be a critical element for enabling countries to meet their emission reduction goals.

First, I will review some key scientific results that illustrate just how far we have already come in changing the Earth’s atmosphere and show that the climate is beginning to react. Then I will review ICT’s role in GHG emissions and describe several advances that allow us to reduce future emissions.

CURRENT EARTH SCIENCE RESEARCH

Recent research has developed a probability distribution for the warming that we can expect from the carbon dioxide and other greenhouse gases already emitted since the beginning of the Industrial Age 250 years ago.¹ The most probable outcome, shown in Figure 1 from that study, shows that over time, about 2.5°C warming will occur as a direct result of our past emissions. However, to this point, we’ve seen only about a 0.8°C increase in warming, or only a third of what is going to happen. This delay has two major reasons: First, it takes about 50 years for the thermal equilibrium of the oceans to adjust; second, the aerosols that are being emitted, particularly in Asia, are cooling the Earth. Somewhat ironically, the rest of that warming will appear as we clean up current air pollution over the next few decades.

There is an emerging scientific consensus² on a variety of climate tipping points which begin to occur as the global temperature rises—for example, melting of the summer Arctic ice, the Himalayan glaciers, and the Greenland ice sheet. As Figure 1 shows, the current level of greenhouse gases has already committed us to serious environmental changes. Unfortunately, as we continue to add more GHGs

to the atmosphere, the peak of this curve moves further to the right, fostering greater disruption.

Arctic ice sheet

If this analysis is correct, we would conclude from Figure 1 that we should already be seeing indications that we are past the first climatic tipping point—melting of the Arctic Ocean summer ice. Figure 2 presents a summary of NASA satellite data on the Arctic ice sheet over the past three decades.³ As the figure shows, from the 1980s to 2000, much of the ice was several years old, whereas in 2009 very little of this older, thicker, ice remains. The graph of the two-year and older ice shows why climate scientists predict that the Arctic may lose its several-year ice in the next five years, leaving only the annual ice, which is thin enough for ships to move through the Arctic Ocean. While this may be good for ocean transportation, this elimination of Arctic summer ice will cause dramatic climate changes all over the northern hemisphere.

The water towers of Asia are melting

In May 2009, UC San Diego and the University of Cambridge held a three-day conference on the next tipping point in Figure 1: the melting of the glaciers in the Himalayan and Hindu Kush Uplift in Asia, which contains the largest amount of snow and ice outside the north and south polar regions.⁴ This frozen water forms the “water towers of Asia,”⁵ being the source of the great rivers from India to Southeast Asia to China—the Indus, Ganges, Brahmaputra, Mekong, Yellow, and Yangtze, among others—which carry the melting snow and ice to the ocean. Over the past decade, the glaciers have begun to melt very rapidly, impacting the water supplies of over a billion people. Also, temporary natural dams often form, backing up large lakes from the melting ice and snow and then giving way, causing flash floods that can destroy villages downstream.

So we see that there is significant evidence that current levels of GHGs are beginning to shift the climate as predicted in the tipping-point study. Unfortunately, the global GHG emissions are continuing to increase, implying even more profound climate shifts. Therefore it is worth taking a look at just how unusual the current level of carbon dioxide is compared to historical values.

Historical climate oscillations

It is true that the temperature and CO₂ levels of the atmosphere have oscillated over time. Some skeptics say that we are just experiencing another natural oscillation. To examine that hypothesis let’s consider the last series of these oscillations. Figure 3 shows the past 800,000 years of oscillations in both CO₂ and temperature as derived from Antarctic ice cores.⁶ The lowest point of these oscillations coincides with ice ages and the peaks with

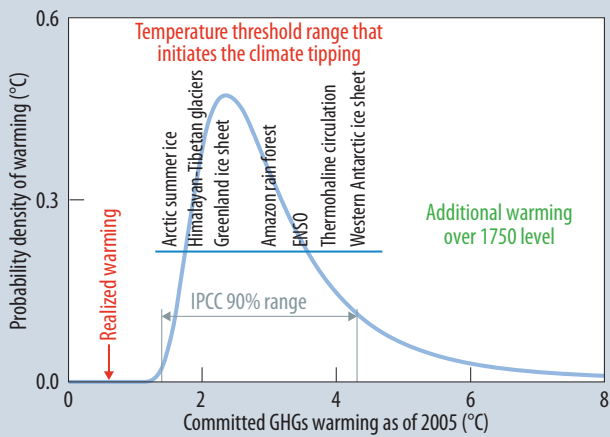


Figure 1. Temperature threshold range that initiates climate tipping. Earth has realized only one-third of the committed warming from previous greenhouse gas emissions; further emissions will move the curve to the right. Adapted with permission from V. Ramanathan and Y. Feng, “On Avoiding Dangerous Anthropogenic Interference with the Climate System: Formidable Challenges Ahead,” *PNAS*, vol. 105, pp. 14245-14250, Copyright 2008 National Academy of Sciences, U.S.A.

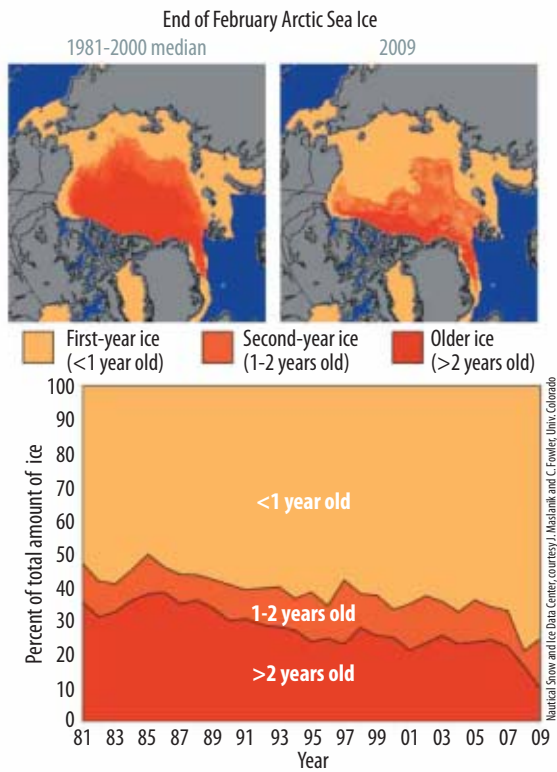


Figure 2. The multiyear Arctic ice sheet has diminished precipitously over the past decade, causing climate scientists to predict that the Arctic may become ice-free in the summer as soon as five years hence. Reproduced with permission, National Snow and Ice Data Center, courtesy of J. Maslanik and C. Fowler.

COVER FEATURE

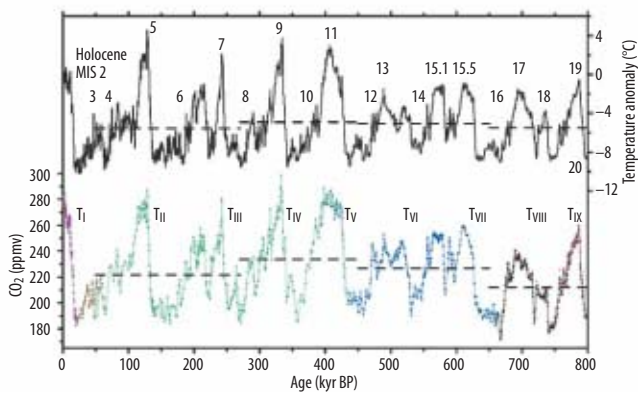
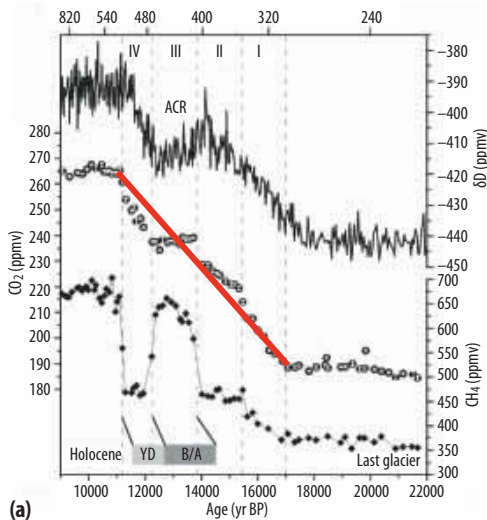
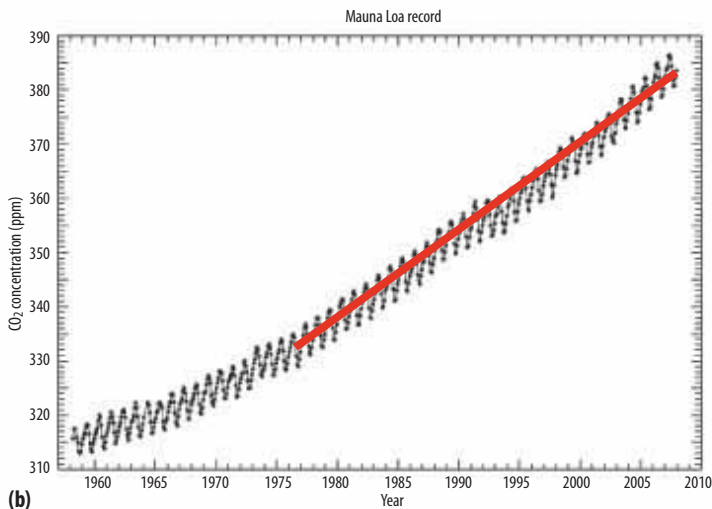


Figure 3. The measured changes in atmospheric CO₂ and temperature from ice cores over the past 800,000 years. Reprinted with permission from *Nature*, Nature Publishing Group, 15 May 2008, vol. 453, pp. 379-382.



(a)



(b)

Figure 4. Time rate of change in the atmosphere's CO₂. (a) The middle curve shows the increase in carbon dioxide emissions since the last ice age. Reprinted with permission from *Science*, vol. 291, 2001, pp. 112-114; (b) increase in emissions over the past 50 years; http://scrippsco2.ucsd.edu/program_history/keeling_curve_lessons.html.

interglacial periods, such as we have been in for the past 10,000 years. As Figure 3 shows, the amount of carbon dioxide in the atmosphere has oscillated between about 170 and 300 parts per million (ppm), while the temperature has oscillated as much as 12 degrees centigrade for many ice age/interglacial cycles.

In contrast, in 2010, the CO₂ will reach 390 ppm, and an MIT study indicates that global economic growth in a “business as usual” scenario would raise this to approximately 900 ppm by the end of this century.⁷ Clearly, we are entering levels of CO₂ that the Earth’s atmosphere has not seen for a very long time.

Next, let’s consider the time rate of change of the atmosphere’s CO₂. Figure 4a shows the warming transition at the end of the last ice age, starting about 17,000 years ago, during which the CO₂ level rose from about 190 ppm to 270 ppm, or about 80 ppm, in 6,000 years.⁸ This yields an average rate of change in the “natural” rise of carbon dioxide of 1.33 ppm per century. Turning to modern times, the Keeling curve (http://scrippsco2.ucsd.edu/program_history/keeling_curve_lessons.html) measured at the Mauna Loa Observatory, where researchers have been measuring carbon dioxide in the atmosphere since 1958, demonstrates that the atmospheric CO₂ level has increased 50 ppm during the past three decades. The slope of the curve indicates the current rate of change is approximately 1.6 ppm per year, over 100 times faster than the “natural rate of warming” experienced during the rise from the last ice age to our current interglacial period.

So we see that both the absolute value of CO₂ and its time rate of change are radically different today from historical oscillations, meaning that the Earth’s climate is dramatically out of equilibrium in an unnatural way. More unsettling still, a recent study by Shell Oil⁹ suggests that with very aggressive global efforts, the level of CO₂ might be held to “only” 550 ppm by 2100, 83 percent higher than the Earth’s atmosphere has seen in 800,000 years. Susan Solomon, one of the world’s leading atmospheric scientists, has carried out calculations showing that global warming will only slowly decline over the following 1,000 years.^{10,11} Clearly, we are facing an unprecedented challenge in this new century.

HOW CAN THE ICT COMMUNITY HELP?

The Smart 2020 report estimates that our ICT industry contributed about 2 to 3 percent of the

total global GHG emissions in 2007, growing at a compounded rate of approximately 6 percent, even assuming efforts to lower the industry’s carbon intensity over the next decade. This means that the total emission will roughly triple between 2002 and 2020. The graphs in Figure 5 include methane, nitrous oxide, and other greenhouse gases, combined into a “CO₂ equivalent” figure. The dark green in the figure shows the life cycle emissions that are associated with making our equipment and then disposing of it. The light green represents the carbon dioxide equivalent associated with generating the electricity needed to operate and cool all our ICT equipment. In forming its 2020 projections, the Smart 2020 study takes into account both the likely technological improvements in energy efficiency over the next decade, as well as detailed projections of adoption rates over various forms of ICT around the world.

The Smart 2020 study shows that all but 14 percent of the ICT emissions in 2020 will occur outside the US and Canada, with China alone emitting twice this level. Clearly, the efforts to reduce the ICT emission intensity will require a global effort.

Which ICT sectors?

The report also divides the emissions into the component parts: emissions resulting from the fixed and mobile telecommunications/Internet infrastructure, data centers, and the edge of the network. Much of the attention in green IT discussions focuses on data centers, whether located in academia and industry or forming the “back end” of the Internet such as those deployed by Google, Amazon, Yahoo, and Microsoft. This makes sense, since these “superclusters” are measured in hundreds of thousands of PCs. Yet the Smart 2020 report shows that this only adds up to less than 20 percent of the total emissions in 2020. The majority (57 percent) will come from the Internet’s edge: PCs, peripherals, and printers. This is because of the enormous scale as China and India rapidly adopt PCs. By 2020, the report estimates there will be 4 billion PCs in the world. So the vast number of PCs is going to dominate this problem.

Cleaning up the edge

Addressing the problem of power management in edge devices needs to start at the system level, focusing on the integration of the hardware and software architectures (<http://scipm.cs.vt.edu>). It requires coordination across processing, communications, and networking in a modern

environment that includes PCs, servers, laptops, and smart phones. All of these devices have complex architectures, including multiple radios, ASICs, microprocessors, DSPs, memories, batteries, AC/DC converters, disks, and displays. This makes monitoring and management of energy a hard problem.

There is wide variance in a component’s energy consumption depending on whether it is asleep or active, as much as 6 to 10 times, and radios have an even wider variance. So how can we exploit this for improved energy efficiency? One strategy is to deploy the devices that use the least energy to shut down the bigger ones when they’re not needed. For instance, it’s possible to coordinate between radios—Wi-Fi, cellular Internet, Bluetooth, Zigbee, and so forth—and use them to page each other to keep the system energy efficient.

The real help here is continued miniaturization, so that there is room to add sensors, control systems, and actuators throughout the system layout to provide data that can feed energy algorithms which attempt to attain the thermal limit of what the devices can achieve.

As UCSD’s Rajesh Gupta has shown, from an algorithmic point of view, a power-aware architecture either shuts down a component using dynamic power management or slows down using dynamic frequency scaling, or both. As a demonstration of this, working with Microsoft Research,

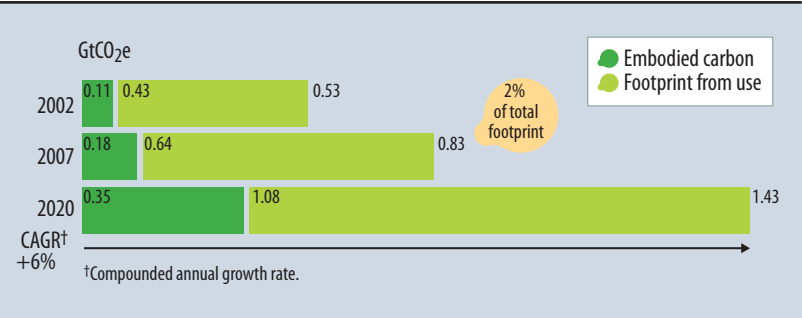


Figure 5. ICT carbon footprint worldwide. ICT emissions are increasing at 6 percent annually, with most of the increase in developing countries. Adapted with permission from the Smart 2020 report.

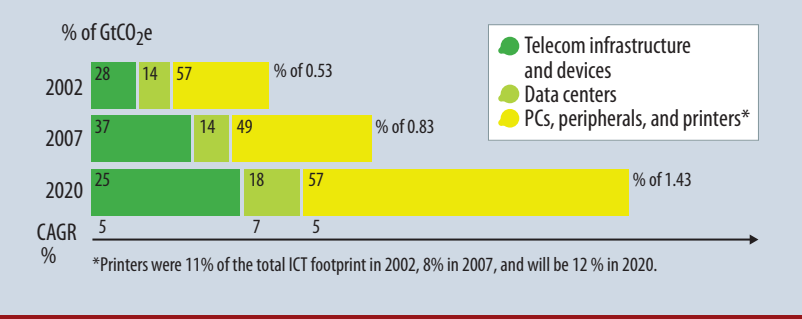


Figure 6. ICT carbon footprint by industry sector. The number of PCs (desktops and laptops) globally is expected to increase from 592 million in 2002 to more than 4 billion in 2020. Adapted with permission from the Smart 2020 report.

COVER FEATURE

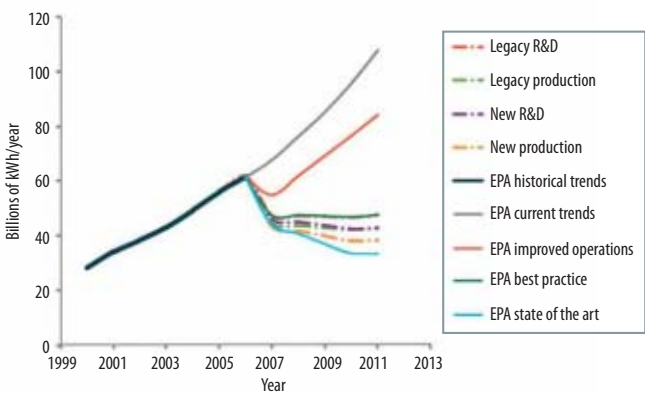


Figure 7. Annual electricity use in US servers and data centers.
Source: EPA Report to Congress on Server and Data Center Energy Efficiency, 2007.

Gupta and his team developed the Somniloquy architecture (<http://mesl.ucsd.edu/yuvraj/research/documents/somniloquy-NSDI09-yuvraj-agarwal.pdf>). An implementation of this approach, housed in a standard USB device and inserted into a ThinkPad laptop, manages radios and hardware components so that it's possible to power down to only 1W doing normal work and achieve 63 hours of battery life—as opposed to the four to six hours available with normal (16W) or low (11W) power strategies. Either widespread use of such devices or engineering this capability into edge devices themselves could have a major impact on reducing the estimated 2020 emissions for ICT.

Data centers

The professionals who run data centers have made major improvements in energy intensity in the past four years. As an example, consider the findings of the Data Center Demonstration Project, launched by the Silicon Valley Leadership Group and Lawrence Berkeley National Laboratory, with 17 case studies.¹² As Figure 7 shows, they find that with best-practice methods, the inexorable rise in the kWhr/year that had been taken for granted can be reversed and indeed rapidly decreased. Fundamentally, this involves a series of techniques that move traditional data centers from a strategy of cooling the entire room to one of cooling just the heat-generating processors.

One innovation for small data centers is to enclose the racks in a box not much bigger than the equipment. For instance, Sun Microsystems has created a Modular Data Center (www.sun.com/products/sunmd/s20) out of an international cargo container that can hold seven racks with both air and chilled-water cooling, adding sensors that monitor temperature and energy and allow for active management of disks, CPUs, routers, and so on. The NSF-funded GreenLight project at UCSD (<http://greenlight.calit2.net>) has purchased two of these data centers to explore methods that go beyond simple physical cooling to the role

of software and the applications themselves for increasing energy efficiency.

To provide a realistic load on the system, Calit2 has pulled together a wide range of computational science applications. For example, an end user doing metagenomics can use this service-oriented architecture to run an application remotely, choosing a variety of algorithms and running them on different computer architectures—multicore, GPU, FPGA, and so on—each combination of which has a different energy profile and turnaround time that can be measured and published in an open fashion on the Web. Project researchers are also developing middleware that automates the optimal configuration of hardware and software.

As Tajana Simunić Rosing and her colleagues at UCSD have shown,¹³⁻¹⁵ dynamic power management coupled with machine learning based on the outcomes of the sensors and performance counters can control voltage and frequency, achieving up to a 70 percent energy savings for a certain class of workloads. For dynamic thermal management, machine learning can predict that a certain algorithm, say a graphics algorithm, generates significant heat when it reaches the GPU, so the system can precool the GPU to immediately transfer the heat—with up to a 60 percent reduction with no performance hit. These are pioneering examples of how thinking about the interaction of software and hardware can achieve even higher levels of energy efficiency than just thinking about the hardware.

Another example is virtualization. On a typical campus, departments have compute clusters located in poorly air-conditioned rooms and often are only computing a small fraction of the time, even though the electricity to run and cool them is running 24/7. Virtualizing the workload to run on a larger system that is enclosed in an energy-efficient environment and that runs 80 percent of the time allows many more calculations for the same amount of energy. If the end user's laboratory is connected to the centralized cluster with a 10-Gbps clear channel optical fiber as in the GreenLight project, there will be no more latency than if the cluster was in the user's lab, yet the campus will be spending less on energy and lowering its carbon footprint.

Finally, if the energy can be generated in a manner that does not produce carbon emissions, then we end up in the best of all possible worlds—computing with zero carbon emission. Since campuses are beginning to install zero-carbon energy sources such as solar panels or fuel cells, why not use them to power the data centers? Even better, since these sources produce DC, we can use the power source to directly power the computer (which natively runs on DC) and save the wasted energy that goes into AC/DC conversion.

At UCSD, we are installing 2 megawatts of solar power cells, and next year we're going to launch a 2.8-megawatt fuel cell that liquefies methane produced at the Point Loma waste treatment plant and uses it as fuel. This process produces no carbon dioxide—in fact, it recycles the methane that would normally be released into the air—and we could run 10 or 20 Sun Modular Data Centers from this one fuel cell. As part of our GreenLight project we are exploring this option with Lawrence Berkeley Laboratory.

Whether in a laptop or a data center, we're wasting a large fraction of the energy we are using to power and cool ICT devices because we haven't focused on how we can more efficiently perform our calculations. However, the many experiments around the US and the world are a positive sign that this will change soon.

All of this effort is intended to develop ICT components that use less energy, so that we can use more ICT to build out smart infrastructure in electric grids, transportation systems, logistic systems, and buildings. The Smart 2020 report shows that such applications of ICT could reduce global carbon emissions by five times the amount of the entire ICT sector. We have a great opportunity in academia to explore these possibilities and transfer our innovations to society at large, because our campuses are essentially small cities, which can be thought of as testbeds¹⁶ for exploring changes that lead to a greener future. This process is already under way at many campuses (www.presidentsclimatecommitment.org).

With academia becoming first movers, they can drive innovations that will be transferred to the market and applied at scale, helping to speed society's transition from a high-carbon to a low-carbon economy. **E**

References

1. V. Ramanathan and Y. Feng, "On Avoiding Dangerous Anthropogenic Interference with the Climate System: Formidable Challenges Ahead," *Proc. Nat'l Academy of Sciences* (PNAS 08), vol. 105, no. 38, pp. 14245-14250, 2008; www.pnas.org/content/105/38/14245.full.pdf+html.
2. T.M. Lenton et al., "Tipping Elements in the Earth's Climate System," *Proc. Nat'l Academy of Sciences* (PNAS 08), vol. 105, 2008, pp. 1786-1793.
3. C. Lombardi, "NASA Images Show Thinning Arctic Sea Ice," CNET News, 7 Apr. 2009; http://news.cnet.com/8301-11128_3-10213891-54.html.
4. "Ice, Snow, and Water: Impacts of Climate Change on California and Himalayan Asia;" http://esi.ucsd.edu/esiportal/images/cambridge/ucsd_cambridge_gwi-conference_05-2009_press_declaration_final.pdf.
5. <http://maps.grida.no/go/graphic/water-towers-of-asia-glaciers-water-and-population-in-the-greater-himalayas-hindu-kush-tien-shan-tib>.
6. D. Luthi et al., "High-Resolution Carbon Dioxide Concentration Record 650,000-800,000 Years before Present," *Nature*, vol. 453, 2008, pp. 379-382.
7. A.P. Sokolov et al., "Probabilistic Forecast for 21st Century Climate Based on Uncertainties in Emissions (without Policy) and Climate Parameters," *J. Climate*, vol. 22, no. 19, pp. 5175-5204, 2009; http://globalchange.mit.edu/pubs/abstract.php?publication_id=99.
8. E. Monnin et al., "Atmospheric CO₂ Concentrations over the Last Glacial Termination," *Science*, vol. 291, 2001, pp. 112-114.
9. Shell Oil, "Shell Energy Scenarios to 2050"; www-static.shell.com/static/public/downloads/brochures/corporate_pkg/scenarios/shell_energy_scenarios_2050.pdf.
10. S. Solomon et al., "Irreversible Climate Change Due to Carbon Dioxide Emissions," *Proc. Nat'l Academy of Sciences* (PNAS 09), vol. 106, no. 6, pp. 1704-1709; www.pnas.org/content/106/6/1704.full.
11. D. Archer, *The Long Thaw: How Humans Are Changing the Next 100,000 Years of Earth's Climate*, Princeton Univ. Press, 2009.
12. Silicon Valley Leadership Group, "Data Center Energy Forecast," 29 July 2008; http://svlg.net/campaigns/data-center/docs/DCEFR_report.pdf.
13. G. Dhiman and T. Šimunic-Rosing, "System-Level Power Management Using Online Learning," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, May 2009, pp. 676-689.
14. A. Coskun et al., "Static and Dynamic Temperature-Aware Scheduling for Multiprocessor SoCs," *IEEE Trans. Very-Large-Scale Integration Systems*, vol. 16, no. 9, pp. 1127-1140.
15. A. Coskun et al., "Utilizing Predictors for Efficient Thermal Management in Multiprocessor SoCs," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1503-1516.
16. B. St. Arnaud et al., "Campuses as Living Laboratories for the Greener Future," *Educause Rev.*, vol. 44, no. 6, 2009, pp. 14-33.

Larry Smarr is the Harry E. Gruber Professor in the Department Computer Science and Engineering of the Jacobs School of Engineering at the University of California, San Diego, and is the founding director of the California Institute for Telecommunications and Information Technology, a UC San Diego/UC Irvine partnership. His research interests are in high-performance cyberinfrastructure, climate change, and green IT. He received a PhD in physics from the University of Texas at Austin. He is a member of the National Academy of Engineering and a Fellow of the American Physical Society and of the American Academy of Arts and Science. Contact him at lsmarr@ucsd.edu and follow him on twitter as lsmarr.

Vint Cerf and Munindar Singh, guest editors for *IEEE Internet Computing's* January/February special issue, have invited a dozen computing luminaries to discuss their predictions for the Internet's future over the course of this new decade. For additional information, see computingnow.computer.org.

COVER FEATURE



REALLY RETHINKING ‘FORMAL METHODS’

David Lorge Parnas, *Middle Road Software*

We must question the assumptions underlying the well-known current formal software development methods to see why they have not been widely adopted and what should be changed.

The theme of the feature articles in *Computer's* September 2009 issue was “Rethinking Formal Methods.” Rethinking this subject is certainly long overdue.

It has been more than 40 years since the late Robert Floyd showed us how to “assign meaning to programs” and demonstrated how we could verify that programs will do what they are intended to do.¹ It has been at least 35 years since I first heard Jean-Raymond Abrial present the ideas that were the basis of Z and its many dialects. The Vienna Development Method (VDM) community began its work about the same time. Even second- and third-generation formal methods show signs of age.

Since 1967, there have been numerous “revolutions” on the hardware side and amazing improvements in man-machine interfaces. The computer systems on my desk today were unimaginable when Floyd wrote that article. Unfortunately, there hasn't been comparable progress in formal methods. There have been new languages and new

logics, but the program design errors we saw in 1967 can still be found in today's software. Applications of formal methods to industrial practice remain such exceptions that they confirm that the use of formal methods is not common practice.

We must question the assumptions underlying current formal methods to see what needs to be changed. The articles published in *Computer's* September issue did not do that; they presented minor variations of ideas that their authors, and other researchers, have advocated for years.

CLAIMS OF PROGRESS AND INDUSTRIAL ADOPTION

The computer science research literature reveals that “formal methods for software development” are a popular research area. Variants of these approaches are frequently discussed and debated at conferences and in journals.

Funding agencies often require that larger research-funded projects include some cooperation with industrial organizations and demonstrate the practicality of an approach on “real” examples. When authors report such efforts, they state that they are successful. Paradoxically, such success stories reveal the failure of industry to adopt formal methods as standard procedures; if using these methods was routine, papers describing successful use would not be published.

Industry is so plagued by errors and high maintenance costs that it would use any method it thought would help; it chooses not to use methods such as Z or VDM.

Reports of successful industrial adoption do not always stand up to scrutiny. Sometimes, the authors are just playing with words. For example, the technique of placing debugging statements in code, taught to me in 1959, has recently been trumpeted as “industrial use of assertions.”

In other cases, close scrutiny reveals truly heroic efforts with very complex formal models but little evidence that the actual code is correct. Often, these efforts do not lead to repeat use or broader adoption of the method. Development organizations that routinely use these methods for actual products are rare.

Some of the reported success may be attributable to having two people looking hard at the problem and the code. Thirty years ago, in a paper that is still worth reading today, H.S. Elovitz described an experiment in which a program in a second programming language was used in the way that formal method advocates suggest that their notations be used.² One programming language was called the *specification language*; the other was the *implementation language*. One programmer wrote the “specification” and gave it to another, who translated it to the implementation language. The specifier reviewed the translation. The error rate was reduced, and this technique (an earlier version of pair programming) was considered successful. This effect alone could explain the few successes in formal methods application. Reports that formal methods are ready for industrial use must be taken with a grain of salt; if they were ready, their use would be widespread.

THREE ALARMING GAPS

The past 40 years have seen some negative developments in the software field.

Gap between research and practice

The gap between formal methods research and practical software development is much larger today than it was when Floyd wrote his paper. Floyd had been a successful, innovative, and productive programmer. His article clearly reveals the connection between mathematical expressions and the programmer’s product.

Today, many research papers are written as if the mathematics were all that matters. They do not show how to relate the formal models and results to the actual code on real machines. Further, they offer no way to deal with the complexity of software systems.

On the other side, most software developers perceive formal methods as useless theory that has no connection with what they do. There is no quicker way to lose the attention of a room full of programmers than to show them a mathematical formula. Developers see the need for im-

provement and will try almost any new method—provided that it does not look like mathematics.

Gap between software development and older engineering disciplines

There is also a disturbing gap between software development and traditional engineering disciplines. Software developers often identify themselves as engineers, but their education and way of working are not at all like those of traditional engineers. Engineering programs teach basic science, applicable mathematics, and how to apply mathematics to predict the behavior of products. Most computer science departments teach technology (which is often of fleeting value in our rapidly changing field) and abstract mathematics that the students do not learn to apply.

There is a disturbing gap between software development and traditional engineering disciplines.

Twenty years ago, I heard a wise and experienced top-level manager complain that his software developers and his other engineers spoke such different “languages” and thought in such different ways that it was difficult for them to work together. That gap is still with us.

Gap between computer science and classical mathematics

A more unexpected development is the separation between theoretical computer scientists and classical mathematics. Many computer science programs present their students with a very narrow slice of mathematics, usually stressing recent developments and “pure” mathematics over older work and what is called “applied mathematics.” Mathematics has evolved slowly and contains many mature and general concepts that can be used when describing and analyzing computer systems.

As a result of the “split” between computer science and mathematics departments, formal methods often use the notation of mathematics but do not take advantage of potentially useful mature concepts. It is easy to find situations in which a computer science researcher might invent an approach in which a classically trained mathematician would recognize that it was possible to use an older (and simpler) concept to solve the problem.

An insight-provoking illustration of this is the contrast between the approaches of two people at the same institution, mathematician N.G. de Bruijn and computer scientist E.W. Dijkstra, to programming semantics and verification. De Bruijn applied the classical concept of relations, while

COVER FEATURE

Dijkstra invented his “predicate transformers.” In my work, I have found the relational approach far more convenient and easier to use.

BEAUTY AND THE BEASTS

Most articles about the use of mathematics in software development contain two distinct messages.

- A general message that reminds us of the ubiquity of faults in software and argues that the use of mathematical notation and reasoning can ameliorate the situation.

We need to question, and be prepared to discard, most of the methods that we have been discussing and promoting for all these years.

- A specific message that describes a detailed syntax and semantics for a language that can be used to describe a model and rules that allow us to “reason about” that model and thereby check certain properties of it.

I always find the general message convincing. Educated as an electrical engineer, I know that mathematics is a valuable tool for all engineering disciplines and that there is no reason that software should be an exception. As a daily user of current software, I see faults that I am convinced would not be there if mathematics had been used for software in the same way that engineers use it for designing physical products. However, I find the specific messages unconvincing.

- Even on small examples, it often appears that the model is more complex than the code.
- The model is a program (a sequence of data transformations); it is not easier to write or understand the model than it would be to write or read the program that would actually run.
- The models often oversimplify the problem by ignoring many of the ugly details that are likely to lead to bugs.
- Often the example does not include the final code, and, if it does, it is difficult to “connect” the model with the code; it seems possible for the model to be proven correct in spite of subtle errors in the code.

The problems are exacerbated in larger examples. Often, it is necessary to understand a lot about the model

to answer even simple questions about the design. Sometimes the models are “write only” because no one but the author can answer questions.

We must learn to use mathematics in software development, but we need to question, and be prepared to discard, most of the methods that we have been discussing and promoting for all these years. We must examine the assumptions on which these methods are based and see which ones stand up to scrutiny.

WHAT TO RETHINK?

If we are really going to rethink formal methods, we need to objectively reconsider a set of issues.

Identifiers and variables

Robert Floyd, and most who followed him, represented the state of an executing program using the identifiers that appear in the program. Thus, if a program had a variable identified by “dogs,” that string would appear in the predicate expressions used to characterize the state.

Variables in a program are finite state machines; what we generally call the value of a variable is that machine’s state. The identifier is a string that we use to refer to a variable. A variable may have more than one identifier (aliasing), and one identifier may refer to different variables in different parts of a program.

In mathematics, variables are placeholders used to define functions/relations; they have neither state nor value. The difference is not always noted because the identifiers in programs look like the variables in mathematics.

Floyd implicitly assumed a one-to-one correspondence between the program variables and the identifiers. He also assumed that if an identifier does not appear in an expression or program line, the corresponding variable is not involved in the calculations when that line is executed. This is not always the case. “Workarounds” have been proposed, but they tend to complicate use of the methods. Many researchers have suggested that using programming techniques that destroy a simple relationship between identifiers and variables—for example, pointers—is bad programming practice; those practices are useful, and researchers are trying to cover up weaknesses in their methods by casting aspersions on things they cannot handle.

Arrays are a particularly vexing example of this problem. “A[j]” and “A[2]” may be identifying the same variable. If we treat them as different, we can prove a program correct when it is not. Edsger Dijkstra proposed treating the whole array as a single variable. This works nicely when all elements of an array are used in the same way, but is not always helpful.

A better way to deal with arrays is needed. In general, it is time to rethink how states should be represented.

Conventional expressions or something more structured?

The mathematical expressions used in most methods are relatively easy to read for simple expressions and when they are used to describe continuous functions. For complex programs, we are describing piecewise-continuous or discrete valued functions; in those cases, conventional expressions can be very hard to read and write correctly.

It is time to look for new forms of expressions that are designed for use with the functions implemented by digital computer programs.

Hidden state: Normal or extension?

Variables in the programming languages of the 1960s had the property that there was a simple relation between the visible value and the state. With the introduction of abstraction or information-hiding, and the subsequent introduction of user-defined "abstract types" into programming languages, it became possible to have variables such as stacks where the visible values are only part of the state. To deal with this, formal models often include ad hoc explicit state representations. These are arbitrary and often add complexity.

It is time to consider hidden state to be the normal case and develop methods that deal with it systematically.

Termination: Normal or exception?

The earliest formal methods assumed that a program would be started with initial values of a data structure, and execution would terminate with an answer in that data. They introduced the concept of partial correctness, which meant that if a program terminated, the answer would be correct, but the program might not terminate. Yet some programs fail by not terminating and others are intended to execute indefinitely. Extensions and notations that deal with nonterminating programs have been added to a basic model that assumes otherwise.

Rethinking requires asking if nontermination should be treated as the normal case in a way that lets us treat terminating programs as a special case.

Time: A special variable or another variable?

When the original formal methods were developed, execution time was not a major concern. If a program executed more quickly than expected, users were happy. If a program was too slow, the user might become annoyed, but the answer was still useful. Time was not a factor in the program correctness proofs.

With the advent of real-time systems, this all changed; programs that are too fast or too slow are incorrect. Special logics were developed for dealing with time issues. This is quite different from such areas as control theory and circuit theory, where time is represented by an additional variable that is not treated in any special way.

Rethinking would require serious consideration of this alternative. We gain simplicity if we do not have to treat time as anything special.

Axioms: Assignment or relational algebra?

In his early paper titled "An Axiomatic Basis for Computer Programming," C.A.R. Hoare introduced about a dozen axioms.³ In logic, axioms are usually simple, intuitive, and obviously universally true. Hoare's axioms (which I believe to be essentially the same as Floyd's) don't have those properties. The axioms describing the arithmetic are not true of any practical computer. The axiom of assignment is only true under very restrictive conditions. The axiom given for iteration is more a sketch of a method of proof than an axiom since it requires identifying the right invariant.

It is time to look for new forms of expressions that are designed for use with the functions implemented by digital computer programs.

There is an alternative. Some researchers have been studying the use of relational methods in computer science; they note that the effect of a terminating program could be described by a relation on states and that the well-known laws of relational algebra can serve as the axiomatic basis for programming. The axioms of relational algebra are simple and universal. They do not embody the characteristics of any particular type of program and can be used with any set of primitive programs. This approach seems to have been neglected by most "mainline" researchers in the area of formal methods.

Direction of analysis: Forward, backward, or inside out?

Floyd, Hoare, and most others analyzed a program in the direction of execution—that is, starting with the first statement and initial conditions and continuing to the end of the program. Loops required an "inductive assertion" or an "invariant," but otherwise the direction was forward. In a break from previous work, Dijkstra proposed going in the opposite direction. His "weakest precondition" approach starts with the desired postcondition and determines what had to be true before the program ran to get the stated result. Few have followed him.

We can work either way. More important, going "bottom up" or "inside out," summarizing inner programs until the whole program has been summarized is also a possibility. In this approach, the relational method has an advantage because the axioms are not based on particular

COVER FEATURE

programs (such as assignment) but are general and apply to programs of any size.

Side effects: Normal or bad?

One limitation on the axioms in the Hoare paper is that they are not true if the programs that evaluate an expression have side effects—that is, if they affect the values of other variables. Most mainline methods disparage side effects as a bad programming practice. Yet even in well-structured, reliable software, many components do have side effects; side effects are very useful in practice. It is time to investigate methods that deal with side effects as the normal case.

Perhaps a formal method should treat nondeterminism as the normal case and deterministic programs as a special case.

Nondeterminism: Normal or extension?

Early formal methods dealt with deterministic programs—programs in which the starting state determines the final state. When such methods are extended to deal with other programs, it is usually an afterthought and more complex. In practice, there are many reasons to deal with a component of a system as nondeterministic. Specifications are usually nondeterministic. Perhaps a formal method should treat nondeterminism as the normal case and deterministic programs as a special case.

Models, descriptions, and specifications

Many papers on formal methods use the words “model” and “specification” interchangeably, perhaps based on a standard dictionary definition of specification as specific information. This definition does not correspond to engineering use, and there are alternatives that should be considered.

- In engineering, the word “specification” is used in a narrower sense, denoting a detailed statement of requirements.
- A “model” of a product is something that resembles the original system but is simpler. Some properties of the model may not be properties of the original.
- A model or document is a “description” if everything that you can learn from it is true of the real thing. A specification is a description of a satisfactory product, but some descriptions are not specifications because they describe properties that are not required.

Using models that are not descriptions is dangerous because they can lead to incorrect conclusions about the real product. Treating descriptions that describe unneces-

sary properties as specifications may result in a product that is overdesigned or unnecessarily expensive. Methods that will be useful in practice must use models that are descriptions and clearly state whether or not they can be interpreted as specifications.

Specifications: Programs or predicates?

Before Floyd’s work became known, some of the researchers interested in verification argued that since we had no way to state what a program should do, we could only prove program equivalence. They would write a program that was “obviously right,” then prove that a more complex, usually more efficient, program would get the same answers.

This way of thinking seems to live on in some approaches to formal methods. The “specification” is a program that describes a sequence of state changes or data transformations.

Unless the sequence of transformations is a requirement, programs should not be used as specifications. An alternative view that has not received enough attention is to view a specification as a predicate. With a predicate you cannot directly compute an answer but you can easily check the correctness of a proposed answer.

It is time to look for methods that use predicates on observable behavior as specifications.

Specification language: Programming language or mathematical description?

The term “specification language” often causes confusion. Since a specification is also a description, specification languages are actually description languages. Further, most notations that are presented as specification languages are actually programming languages.

We should be considering methods that do not use the term “specification language.”

What can be ignored?

A formal analysis uses a simplified description of the real system—that is, a model. Simplification is achieved by ignoring certain facts such as the limits in the sizes of data elements and the errors in arithmetic operations. Unfortunately, these are exactly the type of details that can cause faults and lead to failures. No formal analysis of such a model that leaves out critical limits can reveal faults attributable to those limits.

We should be looking for methods that do not ignore the finite limits that are one of the most frequent causes of bugs.

How do we establish correspondence between model and code?

Because practical programming languages often do not have a complete formal semantics, one that takes into account such issues as the support for software behavior

and finite limits, many formal methods work with a model quite different from the actual code. Often the connection between the code and the model is complex and not clearly described; in such cases, there is a question about whether the model could be (proven) correct while serious bugs remain in the code. In electrical engineering, a mathematical model is usually a set of equations that can be derived from the circuit systematically by, for example, using Kirchoff's laws. Some "idealization" happens in practice, but often this is taken into account by stating tolerances and deriving the possible inaccuracy in the computed result.

We need similar techniques for deriving mathematical models from program text. Floyd was careful to do this, but many of today's methods do not.

Mathematics in documentation

To do their job properly, both programmers and users need precise information. They need to know what is expected of their products and what they can expect of the programs they use. Even "small" details are important because, in software, small errors can cause serious failures.

Experience has shown that natural-language documentation rarely provides what is needed. Those documents are usually incomplete, imprecise, and poorly organized. The information in them is often wrong either because the original writer made an error or because the document was not properly updated when a change was made. There is no way to test an informal document. If the documentation is not trustworthy, a programmer in search of information must either find a knowledgeable colleague or read and understand thousands of lines of code written over many years by many other people. Neither technique is likely to provide complete and accurate information.

One of the most important roles that mathematics could play in software development would be to provide precise, provably complete, easy-to-use, testable documents. The popular formal methods have not been designed with use in documentation as the main goal.

When advocates of formal methods do provide documentation that can help programmers understand a program, it is usually in the form of assertions within the code. This works well for small programs but is impractical for large ones. For large programs and components, there is a need for external documentation that summarizes the behavior of hundreds or thousands of lines of code, allowing a programmer who uses that code to avoid reading it. Some formal methods use abstract models for this purpose, but these models do not usually capture all the details that programmers need to know about the programs they use.

Mathematical documentation should be a major re-

search area in formal methods. It is not.

Pre- and Postconditions

The earliest formal methods were based on associating a program with a pair of predicates. One was the precondition, describing a class of states that must hold before the program is executed; the other was the postcondition, describing the states that must hold afterward. More than 30 years ago Susan Gerhart and Lawrence Yelowitz clearly showed that these are not really two separate conditions.⁴ They specified a sort program with a precondition requiring only that there be some values in an array and a postcondition that the array be sorted. A program that assigned the value *j* to the *j*th element of the array would



Natural-language documentation is usually incomplete, imprecise, and poorly organized.

satisfy this specification.

Instead of two separate conditions, we need a relation between starting state and stopping state. A few researchers have used this approach, but in most methods we still see pre- and post- as separate conditions. Extra variables are often added to allow the initial values to be mentioned in the postcondition.

It is time to consider abandoning the idea of pre- and postconditions.

Correctness proof or property calculation?

Computer scientists interested in mathematical software development methods have focused on "proof of correctness." Strangely, although engineers heavily use mathematics, they rarely use that phrase. Instead, they use mathematics to calculate properties of a product such as voltage on the output, harmonic distortion, heat loss, and so on. They use these calculations to evaluate and compare designs. "Correctness" is a useful term in mathematics, but not in engineering. In engineering, it is usually a serious challenge to define "correctness" for any application. Moreover, engineers are often interested in choosing the best design from a set of "correct" designs. Correctness is not the issue.

Researchers interested in developing practical formal methods should consider the engineering viewpoint; it replaces a vague general question with a set of specific ones.

OBSERVATIONS

This article is intended to ask questions, not answer them. There are, however, some observations that can be made.

COVER FEATURE

Software is broken, but broken formal methods won't fix it

There is widespread agreement that something must be done to improve the quality of software. We have nothing better than mathematics for that purpose. However, there are serious questions about the popular formal methods, and researchers must find answers that are more convincing.

We need research, not advocacy

When we find that people are not adopting our methods, it is tempting to try "technology transfer" and other forms of advocacy. When that fails, which it has, we must return to research and look seriously for ways to improve those methods. It is our job to improve these methods, not sell them. Good methods, properly explained, sell themselves. Our present methods don't sell beyond the first trial.

Reach Higher

Advancing in the IEEE Computer Society can elevate your standing in the profession.

- Application in Senior-grade membership recognizes ten years or more of professional expertise.
- Nomination to Fellow-grade membership recognizes exemplary accomplishments in computer engineering.

GIVE YOUR CAREER A BOOST

UPGRADE YOUR MEMBERSHIP

www.computer.org/join/grades.htm

Step by step from user to code

Software is complex, and the only way to deal with this complexity is to proceed in small, systematic steps so that the relation between the abstract view given the user and the concrete code that runs on the machines can be followed. Any mathematics-based method must be an integrated set of techniques that supports a systematic step-by-step process. The integration means that consistent notation must be used at every step.

Abstract views must be simpler but true

Nobody doubts the value of abstraction, but it is essential to remember that everything that we can derive from an abstraction must be true of the real thing. If we can derive something that is not actually true, what we have is not an abstraction but a lie.

Our role model should be engineers, not philosophers or logicians

Engineers use mathematics in very different ways from pure mathematicians and logicians. Mathematicians who prove theorems use axiom systems that allow them to search for a proof. Engineers usually evaluate expressions, a process that requires no search, just repeated substitution of values for variables and application of functions.

More money is not the answer. It is common for researchers who do not achieve what they set out to achieve to blame the funding. Research in formal methods for software development has been very well funded. More money won't help; more thinking will.

References

1. H.S. Elovitz, "An Experiment in Software Engineering: The Architecture Research Facility as a Case Study," *Proc. 4th Int'l Conf. Software Eng.*, ACM Press, 1979, pp. 145-152.
2. R.W. Floyd, "Assigning Meanings to Programs," *Proc. Symp. Applied Mathematics*, Am. Mathematical Soc., vol. 19, 1967, pp. 19-31.
3. C.A.R. Hoare, "An Axiomatic Basis for Computer Programming," *Comm. ACM*, Oct. 1969, pp. 576-580.
4. S.L. Gerhart and L. Yelowitz, "Observations of Fallibility in Applications of Modern Programming Methodologies," *IEEE Trans. Software Eng.*, vol. 2, no. 3, 1976, pp. 195-207.

David Lorge Parnas is professor emeritus at McMaster University, Canada, and the University of Limerick, Ireland, as well as president of Middle Road Software. His nearly 50 years of research have delved into many topics looking for ways to connect theory and practice in the field of software design. Parnas received a PhD in electrical engineering from Carnegie Mellon University. Contact him at parnas@mcmaster.ca.

COVER FEATURE



FULFILLING THE VISION OF AUTONOMIC COMPUTING

Simon Dobson, *University of St. Andrews, UK*
Roy Sterritt, *University of Ulster, Northern Ireland*
Paddy Nixon and Mike Hinchey, *Lero—the Irish Software Engineering Research Centre*

Efforts since 2001 to design self-managing systems have yielded many impressive achievements, yet the original vision of autonomic computing remains unfulfilled. Researchers must develop a comprehensive systems engineering approach to create effective solutions for next-generation enterprise and sensor systems.

In 2001, IBM researchers predicted that by the end of the decade the IT industry would need up to 200 million workers, equivalent to the entire US labor force, to manage a billion people and millions of businesses using a trillion devices connected via the Internet.^{1,2} Only if computer-based systems became more autonomic—that is, to a large extent self-managing—could we deal with this growing complexity, and they accordingly issued a formal challenge to researchers. We have reached 2010, and, much like the Y2K problem, the situation clearly is not as extreme as anticipated. So was it all hype, or has the IT industry had a very productive decade? Have we met IBM's challenge, or have we simply performed another heroic effort without solving the underlying problem?

BACK TO THE FUTURE

In its autonomic computing call to arms, IBM compared what the IT industry faced in 2001 to what occurred in the US telephony industry in the 1920s. At that time, the rapid expansion and infiltration into daily life of the phone aroused serious concern that there would not be enough trained operators to work the manual switchboards. Analysts predicted that by the 1980s, half the country's population would have to become telephone operators to meet demand. AT&T/Bell System's implementation of the automated switching protocol and other technological innovations averted this crisis.

In 2001, unfilled IT jobs in the US alone numbered in the hundreds of thousands, even in uncertain economic conditions, and global demand for IT workers was expected to increase by more than 100 percent in the next five years. Today's actual employment numbers are hard to determine, as government statistics do not explicitly capture system administration, IT maintenance, and other related functions. However, crude data from the Bureau of Labor Statistics suggests that there are approximately 260,000 IT workers in the US, with employment in the industry declining slightly but steadily over the past decade³ despite the enormous increase in computing power available. This trend suggests that consolidation of computing power, which will increase alongside the use of cloud computing

COVER FEATURE

and Web 2.0, reduces the amount of management per unit of service.

The story is not that simple, of course.

Seven years ago, Jeffrey Kephart and David Chess published “The Vision of Autonomic Computing” in *Computer*,⁴ setting forth IBM’s autonomic computing manifesto¹ in the specific context of enterprise systems management. This article has been wildly influential, with more than 1,100 direct citations according to Google Scholar. Moreover, the study of autonomic systems has become a significant component of systems research, with its own dedicated journals, conferences, and IEEE Computer Society technical committee (TCAAS), as well as a substantial presence in mainstream computing and networking venues.

The most widely recognized elements of autonomic systems are the so-called self-* properties.

The vision of autonomic computing represents a surprising combination of revolution and retrenchment. By focusing on total costs of ownership for enterprise systems, Kephart and Chess highlighted the central impact that IT systems can have on the core economics of modern businesses. Indeed, the deployment, maintenance, and evolution of enterprise systems often require enormous efforts by extremely valuable staff, whose successes add little visible business value but are nevertheless vital and whose failures can be catastrophic for the whole enterprise. Autonomic computing, in its broadest sense, seeks to reduce the need for such heroic efforts and their consequential risks.

To what extent is the vision set forth by Kephart and Chess being fulfilled? What is the status of autonomic computing systems research in its current realization, and how has it influenced research thinking?

THE BROADENING VISION

The increasing use of information systems to collect, analyze, locate, collate, summarize, and otherwise process information has had an immense impact on modern life. That so much of this change has occurred in back offices makes it easy to underestimate the extent to which the design, construction, and especially maintenance of these systems challenge our capabilities as engineers. Feature interaction is a major cause of system failures, and its avoidance is a major cost for system administrators deploying new features.

In some minds autonomic computing today remains closely associated with the original IBM initiative, but to

the IEEE and other organizations the term more broadly describes the application of advanced technology to the management of advanced technology. Similar proposed visions are clearly related: organic computing, bio-inspired computing, self-organizing systems, ultrastable computing, autonomous and adaptive systems, to name a few. We use the term *autonomic* to encompass all of these initiatives.

Enterprise systems are only one member of a class of complicated systems that must function consistently and reliably in the absence of detailed human involvement. Many management tasks can no longer be handled with sufficient efficiency by manual operators, however skilled: The system itself must take responsibility to adapt its own operation in the face of changing conditions. This need for self-adapting behavior characterizes the domains in which autonomic computing ideas are gaining traction.

To take two examples:

- The main cost for the operator of a data center is power, thus the provisioning of systems to match workloads and service-level obligations becomes a critical business success factor. Because workload demands change minute by minute, no human operator can provision services with sufficient efficiency.
- Applications like environmental sensing cause the network to meet the real world in ways that preclude direct human management. The viability of environmental sensing—essential for effective science and policymaking—therefore depends on sensor systems’ ability to self-manage in the face of a changing environment.

The most widely recognized elements of autonomic systems are the so-called self-* properties: For systems to be self-managing they should be *self-configuring*, *self-healing*, *self-optimizing*, and *self-protecting* and exhibit *self-awareness*, *self-situation*, *self-monitoring*, and *self-adjustment*. Despite their seeming simplicity, these goals mask a complex interaction between the behaviors of systems and their goals, users, and relationships with the external environment. We can only optimize a system against some external criteria, so self-optimization implies that these criteria are made available in some way to the management system. Moreover, composition and analysis of systems probably imply that the criteria be explicit, symbolic, and machine-readable rather than embedded implicitly into algorithms.

In thinking of systems rather than simply of machines, we must also consider communications a component of the problem space,⁵ the most notable omission from Kephart’s and Chess’s vision. Mikhail Smirnov⁶ propounded the notion of autonomic communications based on David Clark and colleagues’ call for a *knowledge plane* for the Internet,⁷ and it has become an active research topic.⁸

especially in Europe, where it has received considerable support from the EU's Framework programs. Considering communications as well as computing naturally leads to an exploration of the interplay of these different aspects.

THE EVOLVING STATE OF THE ART

As the "Autonomic Computing: Biological Inspiration" sidebar describes, the term *autonomic* suggests an analogy to the biological nervous system, with the self-* properties similar to those of homeostasis and responsiveness, as well as to the more conventional notions of closed-loop feedback and control. Critics have argued that the autonomic computing field simply synthesizes results from other areas, but this view ignores the breadth that comes from a whole-systems focus. Indeed, we believe that autonomic systems research has the potential to provide a broad systems theory for open adaptive systems.

It is perhaps best to start with the driving forces. Systems are exhibiting rapid increases in complexity of construction, evolution, and management. Putting enterprise systems together is difficult; changing them to meet changing business conditions is complicated by unexpected dependencies and limitations imposed by earlier design decisions; and managing a system to maintain adequate quality of service in the face of a dynamic environment tests the abilities of human managers.

Developers considering the evolution and management of systems in terms of self-* properties must take a different perspective—for example, by including programmatic monitoring and management interfaces. Such a perspective, while common in telecommunications in the form of managed components, is unusual in software architectures still based largely on configuration files read only at start-up time.

As Figure 1 shows,⁸ providing monitoring and control suggests the application of control theory—expressing a control action derived from a system's observed behavior against a model of intended or expected behavior. Researchers have successfully applied such techniques to, for example, power management,⁹ to achieve clear closed-form representations. However, it is less clear whether the techniques can be applied more broadly in areas where the control domain changes dynamically.

There are numerous alternative implementations of control-theoretic ideas. Clark and colleagues' vision of a knowledge plane maintaining a configuration view of a system⁷ allows for distributed access to nonlocal information, which in turn can inform control systems built from agents that act to optimize some local aspect. Such systems' richness comes from the interaction of agents within the agent ecosystem, which can make it difficult to predict overall system behavior in any given circumstances.

The analogy of autonomic computing to biology is proving extremely fruitful. Researchers are using

AUTONOMIC COMPUTING: BIOLOGICAL INSPIRATION

BM's autonomic computing initiative largely derives its inspiration from the biological nervous system.¹ The idea is that built-in regulatory mechanisms in the body that require no conscious thought can suggest the construction of mechanisms that will likewise enable a computer system to become self-managing.

The body's sympathetic nervous system deals with defense and protection ("fight or flight"), and the parasympathetic nervous system deals with long-term health ("rest and digest"), performing vegetative functions such as circulation of the blood, intestinal activity, and hormone secretion. An autonomic computing system similarly tries to ensure its continued health and well-being by sending and monitoring various signals.

An autonomic system has four objectives—to be self-configuring, self-healing, self-optimizing, and self-protecting. These represent broad system requirements. To achieve these objectives, the system must be aware of its internal state (self-awareness) and current external operating conditions (self-situation), detect changing circumstances (self-monitoring), and accordingly adapt (self-adjustment). These four attributes constitute implementation mechanisms. The system must therefore have knowledge of its available resources as well as its components, their desired performance characteristics, their current status, and the status of interconnections with other systems, along with rules and policies of how these may be adjusted.²

The self-managing mechanisms in an autonomic computing system are not independent entities. For instance, a successful attack on the system will necessitate self-healing actions and a mix of self-configuration and self-optimization, initially to ensure dependability and continued system operation and later to increase self-protection against similar future attacks. They should also minimize disruption to users by avoiding significant delays in processing.

References

1. M.G. Hinchey and R. Sterritt, "99% (Biological) Inspiration...", *Proc. 4th IEEE Int'l Workshop Eng. of Autonomic and Autonomous Systems (EASE 08)*, IEEE CS Press, 2007, pp. 187-195.
2. R. Sterritt and D.W. Bustard, "Autonomic Computing—A Means of Achieving Dependability?" *Proc. 10th IEEE Int'l Conf. and Workshop on the Eng. of Computer-Based Systems (ECBS 03)*, IEEE Press, pp. 247-251.

swarm and ant-colony models to coordinate robot behavior, sometimes in combination with more traditional formal methods.¹⁰ The "ANTS: A Milestone in Autonomic Computing" sidebar describes one notable project. Stigmergic approaches capture the notion of depositing time-limited information into an environment to influence later computations. Physics provides another source of inspiration—for example, using models derived from electromagnetic field theory to control load and communications balancing.¹¹

The Internet has become the de facto pervasive communications system across the world today, providing the means for cloud computing. Its success lies in its generality and heterogeneity, the combination of a simple transparent network (the data plane) with rich end-system functional-



COVER FEATURE

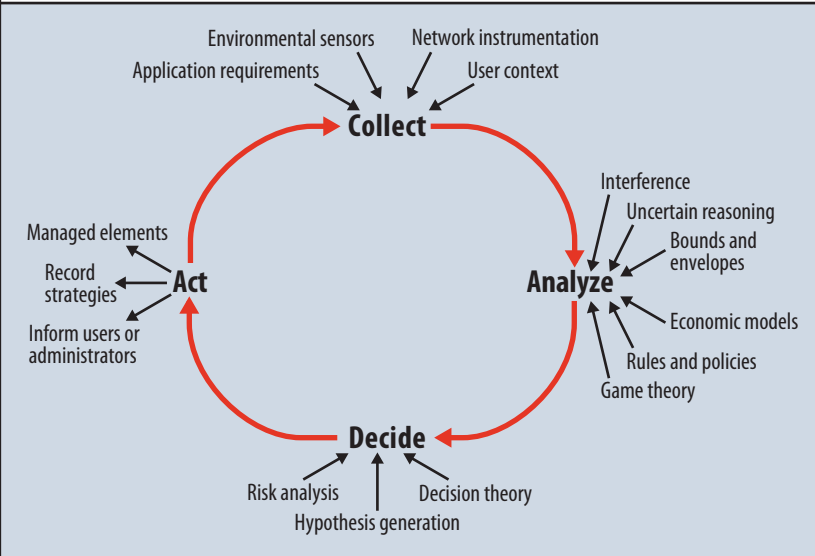


Figure 1. Technologies applied to the four stages of the autonomic control loop. Although inspired by control theory, this structure encompasses symbolic and other techniques within a common framework as well as aspects of both computing and communications.

ity. However, it has high manual configuration, diagnosis, and design costs, and problems become apparent when something fails. Because the Internet has a simple and transparent core and intelligence lies at the edge, the network carries data without knowing its nature or purpose. If a combination of events prevents data from reaching the edge, the core may recognize a problem but has no idea what should be happening.

Researchers recognize that a new construct is needed for next-generation communication networks, a pervasive system within the network that builds and maintains high-level models of what the network is supposed to do to provide services and advice to other network elements. The knowledge plane would function as a global, decentralized overlay to the transparent network that aggregates global information, observations, assertions, requirements, constraints, and goals.⁷ In terms of fault detection and isolation, it would facilitate cross-correlation assessment, with diagnoses traveling up to the knowledge plane and conclusions being passed down. Knowledge plane proponents argue that it can apply machine-learning algorithms to garner knowledge and increase self-awareness.

How to achieve the knowledge plane is an open question, although given the uncertainties and complexities involved it would likely rely on AI and cognitive system tools rather than traditional algorithmic approaches—possible building blocks include epidemic algorithms for distributing data and Bayesian networks for learning. Because the knowledge plane resides in a different space than the data and control planes, it does not move data

such as accounts.^{7,12-13} Researchers are exploring the use of mobile and static agents to provide network knowledge points.

A SYSTEMS THEORY FOR ADAPTIVE SYSTEMS

Autonomic computing techniques provide sophisticated and often extremely impressive solutions to problems that until recently would have been intractable without the intervention of a skilled human operator. However, researchers still lack an understanding of the broader software engineering aspects of autonomic system development. The International Conference on Software Engineering’s SEAMS (Software Engineering for Adaptive and Self-Managing Systems) workshop and the IEEE’s EASE (Engineering of Autonomic and Autonomous Systems) conference and workshops provide forums to explore fundamental

questions about the requirements, specification, and verification of such systems.

Some might argue that adaptive systems do not really differ from other systems: They map an input space to an output space of behavior and actuation. However, the input space may include elements of the environment in which the system operates. In adaptive provisioning, for example, the “environment” includes estimates of expected tasks extrapolated from the previously observed workload. We know from the development of traditional control systems that such feedback requires careful design if the system is not to diverge or exhibit other undesirable dynamics. We also know that this is difficult to accomplish in systems in which the behavior is underspecified and expected to vary over time.

Further, what does it mean for a system to be “correct” when its behavior is expected to change over time? Perhaps a better question is, How can we describe the range of acceptable possibilities for a system’s behavior as well as the preferred behavior at any given instant, and over any given sequence of events? Rather than accepting that systems must simply be “point-correct” in response to a situation at any given time, they must also be “process-correct” by responding correctly to changing situations.¹⁴

In addition, system management is not simply a combination of independent choices, but rather the balancing of a range of possibilities to obtain the best overall result. It is not enough to state, for example, that an autonomic power management subsystem can reduce power requirements: We must be able to state the bounds within which the power demand will vary, its impact on response times, and its interaction with subsystems that may affect load or

➔ **ANTS: A MILESTONE IN AUTONOMIC COMPUTING**

Autonomous Nanotechnology Swarm^{1,2} is a concept NASA mission that represents a significant achievement in autonomic computing.³ In one ANTS submission, the Prospecting Asteroid Mission (PAM),⁴ a transport ship launched from Earth will travel to a point in space where gravitational forces on small objects are all but negligible. From this point, termed a Lagrangian, the transport ship will launch 1,000 pico-class spacecraft assembled en route into the asteroid belt. It is expected that as much as 60 to 70 percent of the craft will be lost during the mission, primarily due to collisions with each other or with an asteroid during exploration operations, since, having only solar sails to provide thrust, their ability to maneuver will be severely limited. Because of their small size, each spacecraft will carry just one specialized instrument to collect data from asteroids in the belt.

Approximately 80 percent of the spacecraft will be *workers* that carry the specialized instruments—a magnetometer or an x-ray, gamma-ray, visible/infrared, or neutral mass spectrometer—to obtain specific types of data. Some will be *rulers* that decide the types of asteroids and data the mission is interested in and that will coordinate the workers' efforts. Finally, *messengers* will manage communication between the rulers and workers, and between the swarm and the Earth ground station.

As Figure A shows, the swarm will form subswarms under the control of a ruler, which contains models of the types of science that it wants to perform. Each worker uses its individual instrument to collect data on specific asteroids and feeds this information back to the ruler, which will determine which asteroids are worth examining further. If the data matches the profile of an asteroid of interest, the swarm will send an imaging spacecraft to ascertain its exact location and create a rough model for other spacecraft to use when maneuvering around the asteroid. Other teams of spacecraft will finish mapping the asteroid to form a complete model.

New approaches to space exploration missions such as ANTS augur great potential but also pose many challenges. The missions will be unmanned and necessarily highly autonomous; to assist in survivability, the swarms will be self-protecting, self-healing, self-

configuring, and self-optimizing.^{5,6} Many ANTS missions will be sent to parts of the solar system beyond the reach of manned spacecraft and to where the round-trip communications delay exceeds 40 minutes, meaning that responses to problems and undesirable situations must be made in situ rather than from ground control on Earth. Future swarm-based missions may employ additional techniques and self-* paradigms.⁷

References

1. W. Truszkowski et al., "NASA's Swarm Missions: The Challenge of Building Autonomous Software," *IT Professional*, vol. 6, no. 5, 2004, pp. 51-56.
2. W.F. Truszkowski et al., "Autonomous and Autonomic Systems: A Paradigm for Future Space Exploration Missions," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 36, no. 3, 2006, pp. 279-291.
3. M.C. Huebscher and J.A. McCann, "A Survey of Autonomic Computing—Degrees, Models, and Applications," *ACM Computing Surveys*, article no. 7, vol. 40, no. 3, 2008.
4. P.E. Clark et al., "PAM: Biologically Inspired Engineering and Exploration Mission Concept, Components, and Requirements for Asteroid Population Survey," *Proc. 55th Int'l Astronautical Congress (IAC 04)*, Int'l Astronautical Federation, 2004; <http://ants.gsfc.nasa.gov/documents/Clark308IAC2004.pdf>.
5. R. Sterritt and M. Hinchey, "Apoptosis and Self-Destruct: A Contribution to Autonomic Agents?" *Proc. 3rd Int'l Workshop Formal Approaches to Agent-Based Systems (FAABS 04)*, LNCS 3228, Springer, 2005, pp. 262-270.
6. R. Sterritt and M. Hinchey, "Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions," *Proc. 12th Int'l Conf. and Workshops Eng. Computer-Based Systems (ECBS 05)*, IEEE CS Press, 2005, pp. 506-511.
7. W. Truszkowski et al., *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, Springer, 2010.

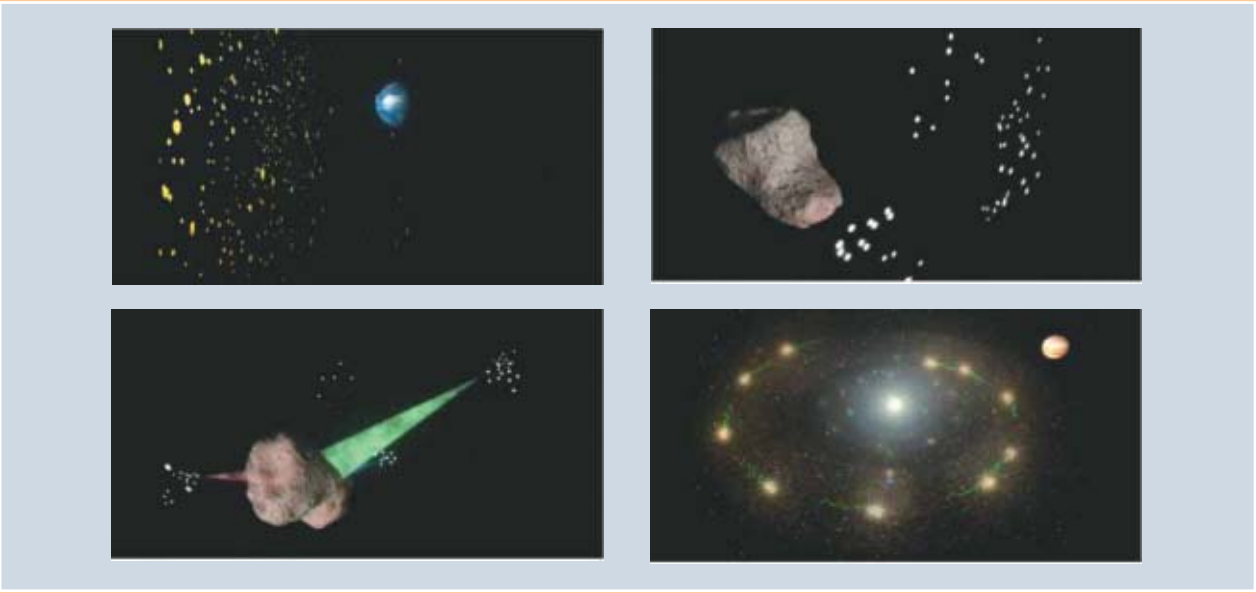


Figure A. Autonomous Nanotechnology Swarm visualization. ANTS is a concept NASA mission that, among other things, will launch highly autonomous swarms of pico-class spacecraft to explore asteroids.

COVER FEATURE

communications, as these factors influence other choices within the management system. In other words, autonomic management requires autonomic mechanisms that are broadly self-describing in terms of the impacts they have across the range of system concerns.

Moreover, we must be able to prove that self-* properties are maintained; analyze the effects of changes on constraints through “what if?” scenarios; and perform these operations on systems composed from individual, independently developed subsystems.

These issues, which are vitally important to the continued development and deployment of autonomic systems, require a comprehensive systems theory for adaptive systems. No such theory has yet emerged, but elements of one are surfacing within the literature, and we can certainly sketch desiderata for any candidate theory.

An autonomic system functions as part of a wider environment and exists to fulfill some externally defined purpose. In describing the system, we must not lose sight of this purpose, suggesting that we capture its requirements and constraints within the description. These in turn provide an envelope within which we can adapt the system.

Adaptation is a dynamic process, not simply a functional response to a change in some variables. There may be many acceptable choices, from which we choose a particular one to exhibit. The different choices may emphasize one system aspect over another. The choices, and the way they vary over time, provide different dynamics through the space of possibilities, a formulation similar to those physicists use with dynamic systems.

For example, suppose the purpose of a wireless sensor network is to sense a pollutant concentration over an extended time. To accomplish this, we must balance the system’s behavior between two extremes. At one pole, the system senses and communicates constantly and expends available power quickly; at the other, it performs no sensing or communication at all and expends virtually no power. Exactly where in the spectrum the system lies is a dynamic decision that might use a baseline of infrequent sensing overlaid with more rapid activity when the network observes “interesting times” and with intensified communications as the battery runs down, to extract the last value from a sensor node.

We may want to select different adaptations depending on observed values of the pollutant as well as first and second time derivatives—for example, start sampling much more rapidly if pollution is increasing rapidly, and then decay sampling more slowly. Any choice will impact the nodes’ longevity and the network overall, and there may be additional constraints on nodes acting as communications gateways or providing other specialized services. The point is that the system description, purpose, and dynamics are essentially one piece and must be stated, analyzed, and evolved as such.

Of course, the mathematics of dynamic systems is nowhere near rich enough to describe all the features of modern enterprise computing. In physics, researchers typically model an already-existing system; in computing, researchers seek to design a system with certain properties, to build complexity by combining subsystems, and to predictably evolve that system over time. Put another way, we need an approach for critical systems that combines adaptive systems analysis with evolving software engineering techniques.

Efforts since 2001 to create self-managing systems have yielded many improvements, yet IBM’s original vision of autonomic computing remains unfulfilled. Indeed, the need for progress is actually greater: IT departments are scrambling to put out fires as technologies such as VoIP and new applications converge, adding even more complexity to our systems.

How we as humans manage an event depends on multiple factors: our mood; the time, place, and circumstances; what happened to us earlier in the day; what we plan to do later; and so on. The sheer complexity of influences on a moment in time and an evolving situation are mind-boggling. Have we as engineers therefore set an impossible goal for ourselves? A computer-based system can be programmed to do tasks but has no emotion, no passion, no soul; how can we create true autonomy from a collection of programs?

Over the years, the AI field has fallen victim to unrealistic expectations, and we see similar warning signs in the autonomic computing field. Yet from the beginning there has been a successful focus on evolutionary research, tightly linked to applied industrial problems. Additional funding and industrial collaboration are crucial to future success, but something more is required: Researchers must develop a long-range, overarching strategy to realize the vision propounded by Kephart and Chess.

That vision has enriched the landscape of systems research, and has in turn been enriched by insights gained within the past decade. It has resulted in a substantial roster of achievements, especially in enterprise and cloud computing but also in communications, sensor networks, and other fields. The idea that computer-based systems can and should be self-managing is having an enormous impact on system design and evaluation.

Yet in some ways that success is deceptive. Researchers have devised innovative autonomic solutions to individual problems, but the larger, more difficult task of combining these point solutions into autonomic systems remains. More consideration must be given to integrating solutions, and to choosing solutions from the range of possibilities—to *autonomic systems engineering*, in other words. Without

such an approach, we will simply rediscover the risks of feature interaction at a higher level, and in a way that is so dynamic as to be resistant to debugging and testing. We are confident, however, that the foundation exists to construct a systems theory and practice from which we can engineer autonomic solutions for the next generation of enterprise and sensor systems. **□**

Acknowledgments

This work is partially supported by Science Foundation Ireland under grant number 03/CE2/I303_1, "Lero—the Irish Software Engineering Research Centre," and the University of Ulster's Computer Science Research Institute (CSRI). The authors also wish to acknowledge the ideas and support of Dave Bustard, Kevin Ryan, Brian Fitzgerald, Lorcan Coyle, M.A. Razzaque, Norah Power, Ron Black, Jim Rash, Chris Rouff, Walt Truszkowski, and Manila Rhythm.

References

1. P. Horn, "Autonomic Computing: IBM's Perspective on the State of Information Technology," 15 Oct. 2001, IBM Research; www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf.
2. E. Mainsah, "Autonomic Computing: The Next Era of Computing," *Electronics & Comm. Eng. J.*, vol. 14, no. 1, 2002, pp. 2-3.
3. US Dept. of Labor Bureau of Labor Statistics, "Current Employment Statistics, Table B-1: Employees on Nonfarm Payrolls by Major Industry Sector, 1959 to Date," [ftp://ftp.bls.gov/pub/suppl/empstat.ceseeb1.txt](http://ftp.bls.gov/pub/suppl/empstat.ceseeb1.txt).
4. J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," *Computer*, Jan. 2003, pp. 41-50.
5. R. Sterritt, "Towards Autonomic Computing: Effective Event Management," *Proc. 27th Ann. NASA Goddard Software Eng. Workshop (SEW 02)*, IEEE CS Press, 2002, pp. 40-47.
6. M. Smirnov, *Autonomic Communication: Research Agenda for a New Communications Paradigm*, tech. report, Fraunhofer FOKUS, 2004.
7. D.D. Clark et al., "A Knowledge Plane for the Internet," *Proc. 2003 Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (Sigcomm 03)*, ACM Press, 2003, pp. 3-10.
8. S. Dobson et al., "A Survey of Autonomic Communications," *ACM Trans. Autonomous and Adaptive Systems*, vol. 1, no. 2, 2006, pp. 223-259.
9. D. Kusic et al., "Power and Performance Management of Virtualized Computing Environments via Lookahead Control," *Proc. 2008 Int'l Conf. Autonomic Computing (ICAC 08)*, IEEE CS Press, 2008, pp. 3-12.
10. C.A. Rouff et al., "Towards a Hybrid Formal Method for Swarm-Based Exploration Missions," *Proc. 29th Ann. IEEE/NASA Software Eng. Workshop (SEW 05)*, IEEE CS Press, 2005, pp. 253-264.
11. M. Mamei and F. Zambonelli, "Programming Stigmergic Coordination with the TOTA Middleware," *Proc. 4th Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS 05)*, ACM Press, 2005, pp. 415-422.
12. J.M. Agosta and S. Crosby, "Network Integrity by Inference in Distributed Systems," *NIPS Workshop on Robust Communication Dynamics in Complex Networks*, NIPS Foundation, 2003; www.agosta.org/pubs/NIPS03/Agosta.pdf.
13. R. Sterritt, S. Dobson, and M. Smirnov, eds., *Proc. IJCAI Workshop on AI and Autonomic Comm.*, IJCAI, 2005.
14. J. Coutaz et al., "Context Is Key," *Comm. ACM*, vol. 48, no. 3, 2005, pp. 49-53.

Simon Dobson is a professor in the School of Computer Science at the University of St Andrews, UK. His research focuses on the design, programming, and analysis of autonomic and sensor systems. Dobson received a DPhil in computer science from the University of York, UK. He is a senior member of the IEEE and the ACM and a fellow of the British Computer Society (BCS). Contact him at sd@cs.st-andrews.ac.uk.

Roy Sterritt is a faculty member in the School of Computing and Mathematics, and a researcher in the Computer Science Research Institute, at the University of Ulster, Northern Ireland. His research focuses on the engineering of computer-based systems, in particular self-managing/autonomic systems. Sterritt received a BSc in computing and information systems from the University of Ulster. He is a member of the IEEE and the IEEE Computer Society. Contact him at r.sterritt@ulster.ac.uk.

Paddy Nixon is a research area leader at Lero—the Irish Software Engineering Research Centre, and Science Foundation Ireland Research Professor in Distributed Systems at University College Dublin. His research interests lie in the areas of pervasive and autonomic computing. Nixon received a PhD in computer science from the University of Sheffield, UK. He is a chartered engineer and a member of the IEEE, the ACM, and the BCS. Contact him at paddy.nixon@ucd.ie.

Mike Hinchey is scientific director of Lero—the Irish Software Engineering Research Centre and a professor of software engineering at the University of Limerick, Ireland. His research interests include self-managing software and formal methods for system development. Hinchey received a PhD in computer science from the University of Cambridge. He is a senior member of the IEEE and currently chairs the IFIP Technical Assembly. Contact him at mike.hinchey@lero.ie.

Join the IEEE
Computer Society

www.computer.org



OPPORTUNITIES IN OPPORTUNISTIC COMPUTING

Marco Conti, *Italian National Research Council*
Mohan Kumar, *University of Texas at Arlington*

When two devices come into contact, albeit opportunistically, it provides a great opportunity to match services to resources, exchange information, cyberforage, execute tasks remotely, and forward messages.

In recent years, opportunistic networks have gained popularity in research and industry as a natural evolution from mobile ad hoc networks (MANETs). In opportunistic networks, nodes come into contact with each other opportunistically and communicate wirelessly. Opportunistic networks are human-centric because they opportunistically follow the way humans come into contact. Therefore, opportunistic networks are tightly coupled with social networks and can exploit human relationships to build more efficient and trustworthy protocols.

OPPORTUNISM

Technological advances are leading to a world replete with mobile and static sensors, user cell phones, and vehicles equipped with a variety of sensing and computing devices, thus paving the way for a multitude of opportunities for pairwise device contacts. Opportunistic computing exploits the opportunistic communication between pairs

of devices (and applications executing on them) to share each other's content, resources, and services.

Opportunistic computing opens an exciting avenue for research and development, one hitherto not fully exploited, and at the same time expands the potential of opportunistic networks for real-life application problems.

Feasibility study

Yet we may still wonder about the feasibility of such a computing paradigm based on pairwise node contacts. Several motivating factors have led to the concept of opportunistic computing: key technological developments, wireless communications, and device architectures; burgeoning application areas; human-centered pervasive computing; and society-influenced social networks.

Mobile cell phones with integrated technology such as Wi-Fi, cameras, Bluetooth, and other, similar capabilities—along with embedded computing devices in moving vehicles and mobile and static sensory devices, including surveillance cameras and others—are available worldwide at reasonable costs. The widespread use of these devices creates a huge number of contact opportunities that are key to opportunistic communications. Significantly, the frequency and potential of opportunistic contacts are mind-boggling, thanks to the cell phone market's estimated 22 percent annual growth rate.

Worldwide market

Analysts estimate that 3.3 billion people worldwide use cell phones (www.reuters.com/article/technology-News/idUSL2917209520071129)—a little more than half the world's population. This amounts to an estimated potential of one billion parallel opportunistic contacts worldwide at any given time, assuming two billion cell phones are turned on at that time. A conservative look at each cell phone's processor reveals a performance figure of 100 MIPS and communication at 200 Kbps. Exploiting these opportunistic contacts gives the potential to perform approximately one quadrillion processing tasks, and exchange 1 petabyte of data per second.

If we consider the 10 billion ARM processors (<http://arm.com/products>) in use today in embedded systems such as vehicles, appliances, and other devices, the estimates would be much higher. Indeed, a typical downtown or university has hundreds of—or O(100)—street cameras; O(1,000) user cell phone cameras; O(1,000) user devices, including laptops, PDAs, and cell phones; and O(100) desktops and information servers.

Given the plethora of wired and wireless communication technologies, such as Wi-Fi, Bluetooth, cellular, and WiMax, along with device capabilities, opportunistic contacts among pairs of devices are the norm rather than a rarity. The necessary infrastructure for opportunistic computing is thus all-pervasive. Opportunistic networks provide the concrete communications support, while the application scenarios provide the motivation to address opportunistic computing challenges. It is only a question of how and when we overcome the challenges to enhance existing applications and develop new ones. In effect, large-scale opportunistic computing, which can simply be defined as *delay-tolerant distributed computing* (DTDC), has tremendous potential.

BACKDROP

Up through the mid-1990s the computer occupied the center of the computing universe, and users were expected to make all possible adaptations to use it and its associated resources. In recent years, there has been a perceptible shift, however, as researchers and designers began focusing on the user. In his seminal paper, Mark Weiser¹ prophesied that the advent of pervasive computing would enhance the user experience. M. Satyanarayanan² made the case that while technological advances are inevitable, user attention is constant.

Indeed, pervasive technologies from smart spaces to iPhones have begun to recognize the user as the computing universe's center. Opportunistic computing takes pervasive and mobile computing further, to the as yet to be explored but highly potential realm of users' social behavior. Specifically in opportunistic computing, this behavior is utilized

to send and forward messages, acquire and disseminate information, and acquire and distribute resources.

Hitherto, with few exceptions, there have been clear distinctions between servers and clients, producers and consumers of information, service providers and consumers, and resourceful and resource-poor entities. However, the Internet's encroachment into everyday life and the development of new technologies such as social networks and peer-to-peer (P2P) systems on the one hand, and anytime, anywhere wireless communications on the other, has made the gap between these sets narrower.

Further, the pervasive deployment of sensors and radio-frequency IDs (RFIDs) has enhanced the potential of user-generated information. Content is increasingly generated in a participatory fashion by the users themselves, following the user-generated content (UGC) model best exemplified by Web 2.0 services such as blogs, YouTube, and Flickr, along with grassroots journalism and similar movements. The users have become both content producers and consumers.

Given the plethora of wired and wireless communication technologies, opportunistic contacts among pairs of devices are the norm rather than a rarity.

The proliferation of powerful mobile devices and UGC models is strictly intertwined. Clearly, a user with a cell phone in hand that includes a camera, microphone, speaker, and perhaps an RFID reader has become a producer as well as a consumer of information and services.³ The more powerful the user devices are, the likelier it is that they will generate and share content, leading to omnipresent content in devices and the diminishing role of centralized servers.

Utilizing the full potential of opportunistic contacts requires new networking and computing paradigms. While in the past few years significant research efforts have focused on exploiting opportunistic contacts to develop mainly message or content-forwarding applications,⁴ opportunistic computing initiatives are still in their infancy.

Several mobile and pervasive computing projects have attempted to exploit all available resources in the environment in the presence of a significant degree of connectivity among the computing devices. Opportunistic computing's major challenge is to effectively utilize opportunistic contacts to make information available and accessible and to provide collaborative computing services to applications and users. The challenges include content distribution and management in an opportunistic P2P environment; management and sharing of scarce and seemingly

COVER FEATURE

disconnected resources; remote execution of tasks in a delay-tolerant environment; and cross-layer issues such as trust, authentication, and privacy.

THE ENABLERS

Several major challenges must be addressed to make opportunistic computing a reality, but significant research trends also push in this direction. Indeed, from the network perspective, there are already several examples of the opportunistic network paradigm's effectiveness, from special-purpose networks like the Sámi Network to general-purpose networking like the Haggle project (www.haggleproject.org).⁴ At the same time, several application scenarios, from crisis management to pervasive healthcare, are emerging that naturally benefit from the opportunistic paradigm.

Opportunistic computing exploits humans' mobility and their gregarious nature to enable a transmission only if two users are sufficiently close.

Opportunistic networks

While human centrality lies at the core of pervasive computing's vision, legacy wired and wireless network architectures force human communications to follow network engineering paradigms. For example, exchanging a message between two participants at a conference entails at least two mail servers spread across the world—just to carry a few hundred bytes across a local space. Clearly, this overly engineering-centric communication paradigm is derived from the wired Internet. Wireless communications and mobile computing freed computing from the leash of tethered networking, but in these networks, mobility and the related disconnections continue to be an engineering challenge instead of an opportunity to communicate.

Mobility management in MANETs exemplifies the engineering-centric approach in the design of self-organizing networks: Mobility is a challenge to cope with, and routing-protocol design focuses on building stable end-to-end paths, as do mobile nodes. Opportunistic networks represent the first attempt to close the gap between human and network behavior by taking a user-centric approach to networking and exploiting user nodes' mobility as an opportunity—rather than a challenge—to improve data forwarding.⁴

Basically, this approach exploits humans' mobility and their gregarious nature to enable a transmission only if two users are sufficiently close. It might seem that the probability of a source coming into contact with a destination is rare, but the use of opportunistic, delayed paths comprising one or more opportunistic contacts between

the source and the destination transforms this simple idea into a powerful one given the potential of opportunistic contacts. This communication model constitutes a theoretical basis for opportunistic networking.

In opportunistic networks such as MANETs, the communication is multihop, with intermediate nodes acting as routers that forward the messages addressed to other nodes. In this case, however, forwarding is not “on the fly” because intermediate nodes such as mobile relays store the messages when no forwarding opportunity exists—for example, there are no other useful nodes in the transmission range—and exploit any contact opportunity with other mobile devices to forward information.

For this reason, developers refer to the forwarding paradigm as “store, carry, and forward.” In opportunistic networks, the nodes' mobility creates opportunities for communication, unlike MANETs, in which mobility is viewed as a disruption.

In the literature, developers often refer to opportunistic networks as delay-tolerant networks. The DTN architecture and protocols are currently under study in the Internet Research Task Force's Delay Tolerant Networking Research Group (www.dtnrg.org), which is concerned with the interconnection of heterogeneous networks. The DTN approach is based on building an overlay atop disconnected networks to combat network disconnections through persistent storage. The overlay provides functionalities similar to the Internet layer (www.ietf.org/rfc/rfc4838.txt) even if end-to-end connectivity may never be available.

Opportunistic networking is a more general concept as it does not assume any compatibility with the Internet architecture, nor any a priori knowledge regarding the network topology, areas of disconnections, or future link availability. In opportunistic networks, route computations differ from those in traditional Internet- or MANET-routing algorithms. Specifically, forwarding and routing protocols are merged because routes are actually built while messages are forwarded. Indeed, routes must be computed on the fly and hop by hop as each message progresses toward its destination. Nodes carrying messages to be forwarded opportunistically evaluate if any other node they contact could provide a good next hop toward the destination, then hand over the message if so.⁴

The Haggle project has developed and implemented a novel layerless architecture for opportunistic networks. Specifically, this architecture has been implemented on mobile phones with the Windows Mobile OS. Preliminary experimental results have also shown the potential of this paradigm when using simple devices like mobile phones. Similarly, the Metrosense project³ uses Nokia mobile phones with the Symbian OS as sensing devices and opportunistic carriers of the sensed information inside a city.

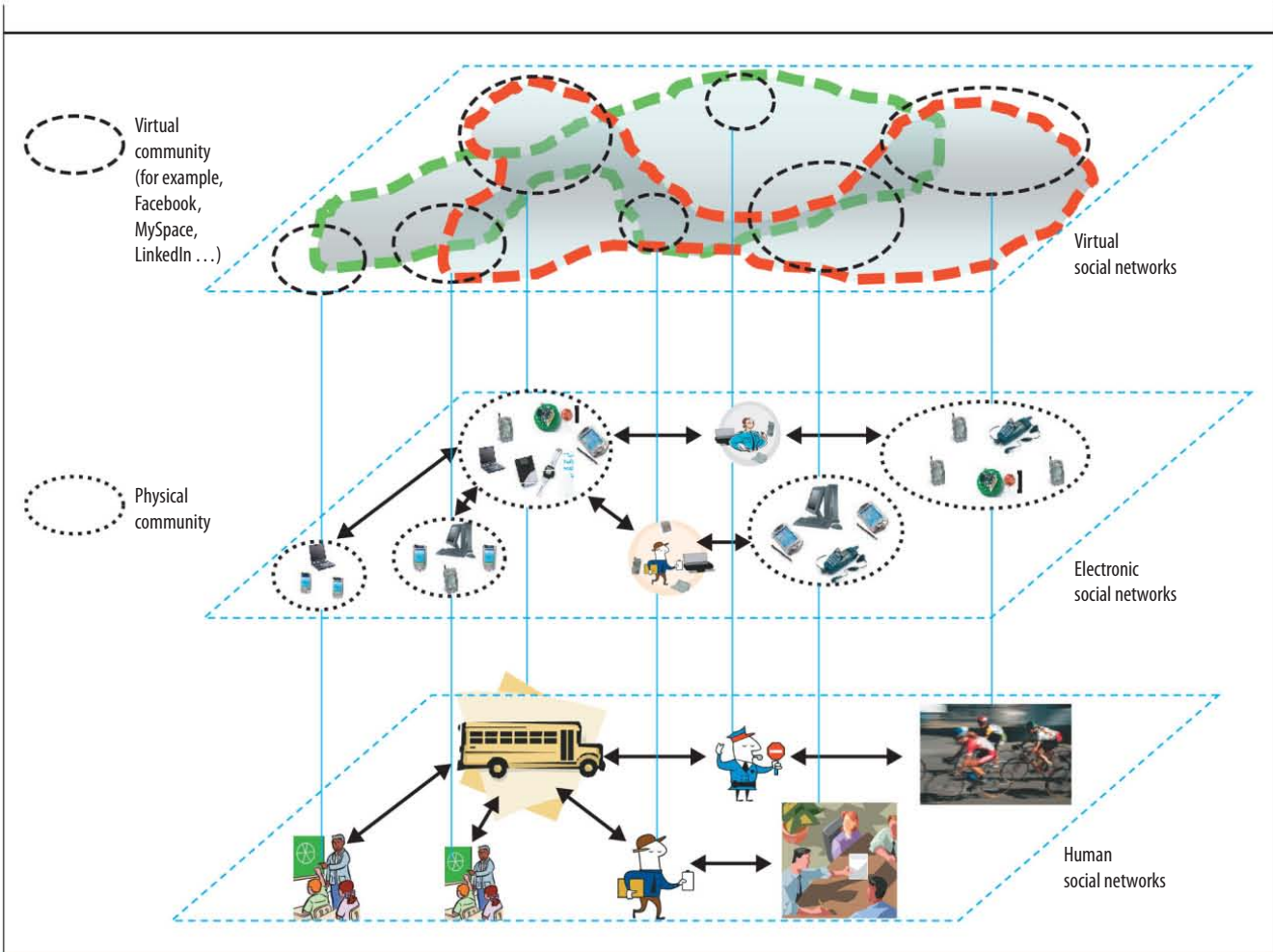


Figure 1. Human, electronic, and virtual social networks. Embedding the social relationships in the electronic world identifies at least two levels in an opportunistic environment: an electronic social network (where relationships depend on the physical properties) and a virtual social network that builds an overlay atop the electronic social network.

Social networks

Opportunistic networking tends to dissolve the traditional networking paradigms and integrate communication more closely with human behavior. Indeed, in the opportunistic networking field, a small but increasing number of attempts have been made to exploit social network features for driving the protocols' design. This looks to be a promising approach, as contacts between nodes are fundamentally tied with users' behavior and hence with social network structures.

However, as of today only some aspects of social networks have been exploited. Studying and modeling human mobility is a research area that has attracted increasing attention. Mobility models based on social behavior represent an important tool for testing the performance of opportunistic systems. Further, a clear understanding of the properties that characterize user movements (such as for any couple of nodes, their contact times, and their intercontact times) provides a cornerstone to design efficient protocols.⁵


A promising direction to fully exploit opportunistic networks involves building the networking solutions around the high-level communication patterns established by the users themselves, rather than applying a legacy engineering-centric approach to bring together devices in a common network plane (layer). An attempt to systematically exploit the underlying social network structure to develop effective social-inspired opportunistic network protocols is currently being carried out inside the Socialnets project (www.social-nets.eu). This project exploits social interactions and user habits to drive the design of protocols for a pervasive system. This is achieved through an interdisciplinary research effort aimed at merging a set of disciplines that are currently running in parallel and without integration—for example, statistical physics studies of complex networks, the social anthropology studies of human behavior, and the computer networking perspective.⁶

Figure 1 summarizes the basic ideas of the Socialnets project. By embedding the social relationships in the

COVER FEATURE

electronic world, we can identify at least two levels in an opportunistic environment: an electronic social network (in which relationships depend on the physical properties) and a virtual social network that builds an overlay atop the electronic social network.

Bubble Rap offers a promising forwarding protocol that tries to exploit the electronic social network idea to design effective opportunistic network protocols.⁷ Specifically, it focuses on two aspects of a social network: the community and the centrality. Human society is structured in communities, and inside a community some are more popular than others: They have a high centrality. The basic idea of the forwarding algorithm is to use nodes with high centrality to deliver a message to the community that makes clear who the destination node belongs to.



Opportunistic computing can benefit from the ongoing and past research outcomes in pervasive and sensor systems, distributed and fault-tolerant computing, and mobile ad hoc networking.

While electronic social network relationships provide key information for designing opportunistic network protocols, the virtual social network provides a basis for the development of opportunistic computing services. For example, information and services can be replicated and distributed inside the community's electronic social network, taking into consideration its members' interests and locations.^{8,9}

OPPORTUNISTIC COMPUTING

Essentially, opportunistic computing can be described as distributed computing with the caveats of intermittent connectivity and delay tolerance. Indeed, mobile and pervasive computing paradigms are also considered natural evolutions of traditional distributed computing. However, in mobile and pervasive computing systems, the disconnection or sleep device situations are treated as aberrations, while in opportunistic computing, opportunistic connectivity leads to accessing essential resources and information.

As Figure 2 shows, opportunistic computing exploits communication opportunities to provide computing services to meet the pervasive application requirements. Opportunistic networking research has benefited from past work in areas such as wireless mobile ad hoc networks and delay-tolerant networks, while pervasive, mobile, and social computing all motivate their respective applications.

Opportunistic computing exploits all available resources in an opportunistic environment to provide a platform for the execution of distributed computing tasks. The major

challenge in opportunistic computing is to effectively utilize opportunistic contacts to make information available and accessible and to provide collaborative computing services to applications and users. To make opportunistic computing a reality, middleware services must mask disconnections and delays and manage heterogeneous computing resources, services, and data to provide a uniform view of the system to the applications.

Opportunistic computing can benefit from the ongoing and past research outcomes in pervasive and sensor systems, distributed and fault-tolerant computing, and mobile ad hoc networking. In particular, work in the areas of heterogeneity and interoperability,¹⁰ proactivity and transparency, context-aware computing,¹¹ location-aware systems, sensor systems, failure handling techniques,¹² and others can be adapted to opportunistic computing systems. However, many challenges to opportunistic computing are unique from those in other systems.

Trusted collaboration

In a disconnected environment, mechanisms for establishing trust among peer nodes play a critical role. Trusted collaboration among the entities of social computing creates opportunities for distributed execution of computing tasks. However, the increasing trend toward decentralization has resulted in significant challenges because traditional security solutions often require centralized online trusted authorities or certificate repositories, which are not well suited for opportunistic networks in which connectivity as well as centralization requirements are both relaxed. Opportunistic networking requires a paradigm shift toward human-centric solutions to establish trust for interactions between peers.

As Figure 2 shows, social links between humans carrying the devices provide strong support for new concepts of trust and security to establish trustworthy relationships among devices and for incentivizing their collaboration at both the network and middleware levels.

Trust, security, and cooperation policies require strict interactions between the two layers, but we also envision several other cross-feed channels. For example, content-based forwarding strategies for routing and forwarding inside the electronic social network layer can exploit context information and content preferences provided by the upper layer.

The collective actions lead to the execution of high-level tasks, as they opportunistically exploit each other's resources. For example, a device that has collected a huge amount of data from the environment can utilize the data compression service offered by another user's device to optimize its memory resources.

More generally, when two nodes meet to perform a collaborative task, they are required to know each other's resources, data, and services. Therefore, when they come

in contact opportunistically, they swap information, process the information, and take actions. For example, when two devices—say *a* and *b*—are within the wireless communication range of each other, they have an opportunity to exchange a description of their services. If *b* has a service relevant to *a*, and *b* is a trustable entity, *a* can send the service input parameters to *b*, which then runs the requested service on behalf of *a*. When the service is completed, *b* returns the output parameters to *a*, immediately if they are still in contact, or otherwise the next time they meet.

Key challenges

Creating a distributed computing platform in a seemingly hostile and unstable networking environment requires overcoming the challenges the opportunistic environment poses.

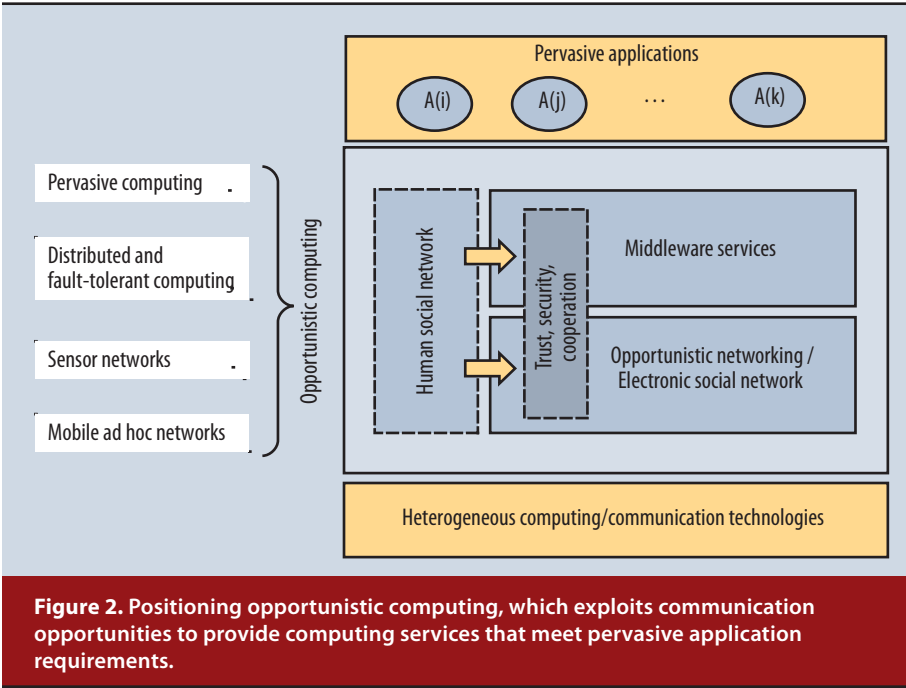
Intermittent connectivity. In opportunistic networks the contact between pairs of devices provides the critical resource for collaboration. The connectivity problem is exaggerated by the lack of prior knowledge about the location, time, and communication bandwidth of such contacts. Hybrid routing protocols that employ context, a profile, or a history of mobile users and devices should be investigated for adoption into opportunistic networking environments.

It will be necessary to develop middleware mechanisms that mask delays and hide the complexity of the opportunistic paths from applications. The information acquired must be evaluated for caching, purging, and dissemination because resources such as memory and bandwidth are limited.

Delay tolerance. Successful implementations of DTN applications have demonstrated the usefulness of opportunistic networks. Delay tolerance is the key to distributed opportunistic computing.

First, protocols for the creation of delay-tolerant opportunistic communication paths should be developed. Second, delay-tolerant information acquisition and dissemination requires new cache consistency mechanisms to mask delays and the underlying network. Third, execution of remote services in opportunistic environments requires novel mechanisms for service discovery, service execution, and management.

Heterogeneity. Potentially, many kinds of devices will likely come in contact opportunistically—cell phones, handheld and notebook computers, sensors, cameras, and



RFID-enabled objects. These devices can be supported by diverse communication capabilities and the radio frequencies at which they communicate. Contact interoperability among these pairs of heterogeneous devices is a major challenge.

RESEARCH ISSUES

Opportunistic networks, combined with social computing, herald the new paradigm of opportunistic computing for pervasive applications. Whereas pervasive computing seeks to enhance user quality of life through proactive application services, opportunistic computing also recognizes and exploits users' social behavior.

User devices, and indeed their BANs/PANs, possess complementary capabilities in terms of computing, communication, storage, energy, sensing, and related applications. This opens several lines of research for developing a set of middleware services that mask disconnections and heterogeneities and provide the applications with uniform access to data and services in a disconnected environment.

Middleware services

Middleware services provide mechanisms for managing information and access through a variety of applications, such as data and services placement, resource management (for example, storage, bandwidth, and energy), trust, security, and privacy for opportunistic computing, mobile agents, remote execution, and cyberforaging, among others. Trust and data privacy pose key issues. For example, reputation mechanisms should be in place to detect malicious users who might join a group to thwart collective actions or acquire sensitive information.

COVER FEATURE

Users' social behavior should be embedded in the middleware mechanism to increase efficiency and security. Novel mechanisms for service sharing in a disconnected environment must be devised. Developing modular tools for message passing, information dissemination and acquisition, resource management, service discovery, service management, and other tasks poses a huge challenge to heterogeneous opportunistic environments.

Fault tolerance is critical to many distributed computing applications. While most current work assumes the existence of local networks, not much research has been done in the prevention, detection, and recovery aspects of faults in challenged environments.

Collaboration in opportunistic environments calls for new, robust strategies that facilitate collaboration in the absence of continuous connectivity. Mechanisms for replication and redundancy must take into consideration the limited resources in such constrained systems. Application tasks executing on one device will be required to interact with resources and services on other devices under time and connectivity constraints.

In an opportunistic computing environment, user devices and sensors carry or supply different kinds of information that is useful to other users and applications.

Information management

In an opportunistic computing environment, special attention must be dedicated to information management and provisioning because a vast range of information is embedded in the environment of pervasive computing and communication systems. However, the use of opportunistic techniques to provide situated information has not yet received much attention, even though there are developments of significant relevance in the distribution of content within P2P systems. Many developers have considered the extension of Internet-based content-sharing systems to mobile ad hoc networks by overlaying the P2P structure. However, very little development has taken place for P2P information provision in opportunistic networking.

The lack of distinction between information producers and consumers on the one hand and the utilization of opportunistic contacts to disseminate and acquire them on the other makes this task challenging. Aggressive broadcasting mechanisms, such as those based on epidemic dissemination protocols, have a tendency to load the network, abusing contact capacity and the content cache. From an information-centric perspective, use of opportunistic networks for information provision results in three

fundamental issues: determining what to store, where to store it, and how to acquire relevant information.

Context awareness

Context awareness is a relevant key for searching the network. First, most of the content is relevant for people physically close to the source, who thus form a transient, local community with which to jointly interact. This requires establishing dynamic and temporary trust relationships between humans and machines. In addition, part of the generated content will be of interest to other users in virtual communities, which share common interests irrespective of their physical location. This means that a much wider range of objects can generate and store information while situated in an environment.

Context information and profiles of devices, individuals, and applications—together with cache optimization techniques—are needed for the effective management of the content cache. To share information within a social environment, researchers have proposed *social caches*. A social cache is a logical collective view of individual device caches that cache information objects useful to the members of its social group. Given that members are expected to meet more frequently, and information in the social cache can be effectively utilized by many members, social caching can significantly increase system performance.⁸

Services and data placement and replication

In an opportunistic computing environment, applications need different kinds of resources to execute services, and such resources may be available within the network. Similarly, user devices and sensors carry or supply different kinds of information that is useful to other users and applications. Users' cooperation is tightly coupled with their organization in social communities. Therefore, to increase system efficiency, it is critical to make services and data available in the environment closer to users who need them.

Replication of data or services increases their availability, while their migration may reduce the access delay. While placement of services is a well-investigated problem in traditional distributed systems, dynamic pervasive environments, such as those created by opportunistic contacts, pose new challenges. Andrea Passarella and colleagues¹³ have investigated efficient and effective schemes for service provision in opportunistic environments. These investigations will lead to deployment of application-level services.

Resource management

The "product of connection time and available bandwidth," termed *contact capacity*, should be used effectively. The capacity is limited and varies in accordance with wireless communication conditions and the mobility of devices and their users. The contact capacity should be utilized

effectively to ascertain reputations, establish trust, collaborate, and exchange information between the two meeting devices and their users.

The second most important resource is the memory or buffer space in the devices themselves. We call this the *content cache*. In opportunistic computing, devices carry one another's information in their content cache, which should be optimally maintained by purging unwanted data and keeping data useful to the applications on the device, such as peers it expects to meet in the near future. The content cache can be tuned to certain applications, contexts, or other criteria.

Energy is another key resource for an opportunistic environment, in which most devices are battery enabled. Energy management is a cross-layer issue with respect to the management of storage and bandwidth. Increased data transmission on the wireless interface results in more energy spent, while local data storage might incur significant energy costs for memory management.

Finally, the hardware and software resources on the devices must be exploited by providing seamless accessibility to applications executing on other devices. As most devices in opportunistic networks are mobile, they possess limited and varied hardware and software resources. Using resources distributed across the devices in a given space, such as a social network, is critical. Matching services to resources in opportunistic networks presents another challenge.

Trust, security, and privacy

Establishing trust and security for an interaction between a priori unknown peers in an opportunistic network is challenging. However, social network structures offer a basis to enhance trust and security provision by capitalizing on "communities" of devices that have commonality between them, either physically or logically.

The idea of using social network structures and properties for enhancing network security is not novel. Indeed, the literature contains several proposals based on using social networks to fight e-mail spam and defend against attacks. However, the use of social networks in completely decentralized networks is a completely new and challenging task because, in such an environment, legacy security solutions based on centralized server or online trusted authorities becomes infeasible. In this case, a natural direction to pursue exploits electronic social networks and the trust and security relationships naturally embedded in human interactions.

Economic model and social cooperation

We might argue that a solid economic model is fundamental to justifying implementation of an opportunistic computing paradigm. Why should one user make computing resources available to another? This is an even more critical question when the computing platforms are mobile

devices that have very limited and critical resources, such as energy.

The development of an economic model to stimulate cooperation among peers has been extensively discussed in the framework of both P2P platforms and mobile ad hoc networking, where solutions based on incentives or reputation have been devised. Similar strategies can probably be applied here. However, we believe that exploiting the natural cooperation that exists in human social relations is the catalyst for opportunistic computing. In principle, rational users gain the most from an uncooperative behavior but, despite this, human society often exhibits cooperative behaviors. Characterizing and enforcing human cooperation is highly relevant for electronic social networks.

Mobile agents, remote execution, and cyberforaging

In an opportunistic computing environment, services are often only available on remote nodes outside direct communication of the requesting device. This requires developing mechanisms to support the remote execution of tasks and return the results to the node(s) requesting a service.

Exploiting the natural cooperation that exists in human social relations is the catalyst for opportunistic computing.

Mobile agent technology can be an effective tool to address this issue. Mobile agents may migrate from one node to another during contacts, carry input data and code, and exploit services and resources in the visited nodes. When a task execution is completed, these agents return to the source node together with their results. Similarly, mobile agents can be employed for information acquisition and dissemination.

APPLICATIONS

Opportunistic computing can be the basis for addressing challenges in many application areas. Three critical application areas can benefit from opportunistic computing services and hence constitute driving forces for research in this direction.

Crisis management

Legacy communications networks are not designed to withstand unplanned and unexpected disruptive events and are unsuitable to reliably support communication services for first responders. Opportunistic networking techniques can be adopted for interconnecting surviving parts of the telecommunication infrastructures, and services can be deployed for specific applications.

COVER FEATURE

Infomobility services and intelligent transportation systems

Vehicular ad hoc networks (VANETs) exploit vehicle-to-vehicle communications, as well as the communication with roadside infrastructure, to implement cooperative systems and to increase traffic efficiency and safety. Other applications include tourist information and assistance such as parking availability notification and maps, and entertainment such as gaming and streaming video.

Pervasive healthcare

Opportunistic computing and network technologies can be used to create a pervasive system of intelligent devices comprising sensors and actuators that embrace patient surroundings at different levels. Transparently embedded body area networks and sensors can cooperatively gather, process, and transport information on our lifestyle and the social and environmental context around us without requiring any major change in the users' behavior.

Opportunistic networking techniques can be deployed as basic tools in distributed context-aware pervasive applications for performing real, noninvasive, continuous multiparametric monitoring of physical and physiological parameters.

Distributed computing on opportunistic networking platforms offers a new paradigm in computing, one with tremendous potential. Opportunistic computing will become a reality in the near future, given current growth in the proliferation of powerful and ubiquitous devices and the variety of applications.

A need exists, however, to develop effective solutions to the many new research challenges posed by this intriguing computing opportunity.

Going forward, we require architectures for reliable, secure, and delay-tolerant computing platforms built atop highly dynamic networks and characterized by transient and distributed interactions among devices. This will in turn require great flexibility, scalability, and a general ability to adapt what is typically embedded in human behavior.

Thus, exploiting the properties of human social links to build an electronic social device network offers a promising direction for developing efficient and effective pervasive computing systems that can adapt to highly dynamic and transient distributed systems. ■

Acknowledgments

Mohan Kumar's research on opportunistic computing is supported under the US National Science Foundation's Grant CSR 0834493. Marco Conti's research is supported by the European Commission under the Haggle (027918) and Socialnets (217141) FET Projects.

REFERENCES

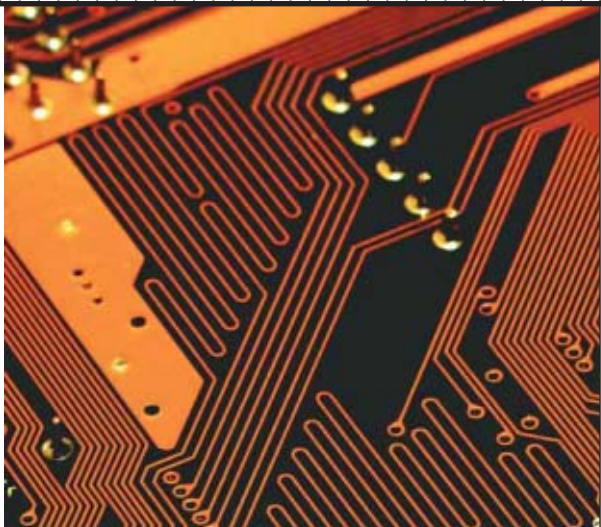
1. M. Weiser, "The Computer for the 21st Century," *Scientific Am.*, Sept. 1991, pp. 94-104.
2. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Computing*, Aug. 2001, pp. 10-17.
3. A. Campbell et al., "The Rise of People-Centric Sensing," *IEEE Internet Computing*, July/Aug. 2008, pp. 12-21.
4. L. Pelusi, A. Passarella, and M. Conti, "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," *IEEE Comm. Magazine*, Nov. 2006, pp. 134-141.
5. C. Boldrini, M. Conti, and A. Passarella, "User-Centric Mobility Models for Opportunistic Networking," *Bio-Inspired Computing and Communication*, LNCS 5151, Springer, 2008, pp. 255-267.
6. S.M. Allen et al., "Social Networking for Pervasive Adaptation," *Proc. 2nd Int'l Conf. Self-Adaptive and Self-Organizing Systems Workshops (SASOW 08)*, IEEE Press, 2008, pp. 49-54.
7. P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay Tolerant Networks," *Proc. 9th ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 08)*, ACM Press, 2008; www.cl.cam.ac.uk/~ph315/publications/hoc86309-hui.pdf.
8. C. Boldrini, M. Conti, and A. Passarella, "Design and Performance Evaluation of ContentPlace, a Social Aware Data Dissemination System for Opportunistic Networks," *Computer Networks*, 2009; <http://dx.doi.org/10.1016/j.connect.2009.09.001>.
9. P. Costa et al., "Socially-Aware Routing for Publish-Subscribe in Delay-Tolerant Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Communications*, May 2008, pp. 748-760.
10. M. Kumar et al., "Pervasive Information Communities Organization (PICO): A Middleware Framework for Pervasive Computing," *IEEE Pervasive Computing*, July-Sept. 2003, pp. 72-79.
11. K. Henriksen and J. Indulska, "Developing Context-Aware Pervasive Computing Applications: Models and Approach," *Pervasive and Mobile Computing*, Feb. 2006, pp. 37-64.
12. Y. Yu and V.K. Prasanna, "Energy-Balanced Task Allocation for Collaborative Processing in Wireless Sensor Networks," *Mobile Networks and Applications (MONET)*, special issue on algorithmic solutions for wireless, mobile, ad hoc and sensor networks, Feb. 2005, pp. 115-131.
13. A. Passarella et al., *Minimum-Delay Service Provisioning in Opportunistic Networks*, IIT-CNR tech. report 11-2009; http://bruno1.iit.cnr.it/~andrea/docs/TR_11-2009.pdf.

Marco Conti is a research director at IIT, an institute of the Italian National Research Council (CNR). His research focuses on the design, modeling, and performance evaluation of computer-network architectures and protocols. Contact him at marco.conti@iit.cnr.it.

Mohan Kumar is a professor in the Department of Computer Science and Engineering at the University of Texas at Arlington. His research interests include pervasive and mobile computing, opportunistic networking and computing, and sensor systems. Kumar received a PhD in computer science from the Indian Institute of Science. He is a senior member of the IEEE. Contact him at mkumar@uta.edu.

RESEARCH FEATURE

A Discrete Stock Price Prediction Engine Based on Financial News



➔ Robert P. Schumaker, *Iona College, New Rochelle, New York*
➔ Hsinchun Chen, *University of Arizona, Tucson*

The Arizona Financial Text system leverages statistical learning to make trading decisions based on numeric price predictions. Research demonstrates that AZFinText outperforms the market average and performs well against existing quant funds.

While researchers have made numerous scientific attempts, no single method has yet been discovered to accurately predict stock price movement. The difficulty in prediction comes from the complexities associated with market dynamics, where parameters are constantly shifting and are not fully defined.

The use of textual data offers one area of limited success in stock market prediction. Information from quarterly reports or breaking news stories can dramatically affect a security's share price. Applying computational methods to this textual data forms the basis of financial text mining. Most existing literature on financial text mining applies a representational technique to news articles where only certain terms are used and weights are assigned to the terms based on the direction the stock price moves. Prediction then applies these weighted terms to a new article to determine a likely direction of movement. To their credit, these simpler forms of analyses have shown a weak but definite ability to predict price direction but not the price itself.

However, using computational approaches to predict stock prices based on financial data is not unique. In recent years, interest has increased in quantitative funds, or *quants*, that automatically sift through numeric finan-

cial data and issue stock recommendations.¹ While these systems are based on proprietary technology, they differ in the amount of trading control they have, ranging from simple stock recommenders to trade executors. Using historical market data and complex mathematical models, these methods are constrained to make assessments within the scope of existing information. This weakness means that they cannot react to unexpected events falling outside historical norms. However, this disadvantage has not stopped fund managers at Federated, Janus, Schwab, and Vanguard from trusting billions of dollars in assets to the decisions of these computational systems.

The Arizona Financial Text system (AZFinText) is a different type of quant trader that focuses on making discrete numeric predictions based on the combination of financial news articles and stock price quotes. Our contribution rests on building the AZFinText system in which trends and patterns are machine learned from stock quotes and textual financial news. While prior textual financial research has relied on tracking price direction alone, AZFinText leverages statistical learning to generate numeric price predictions and then make trading decisions from them. Our research demonstrates that AZFinText outperforms the market average and performs well against existing quant funds.

RESEARCH FEATURE

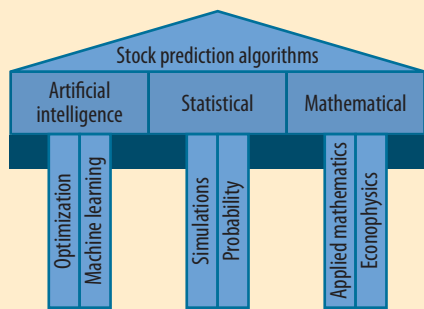


Figure 1. Technical stock prediction algorithms.

SECURITIES PREDICTION

There are several security forecasting theories. Within the confines of *efficient market hypothesis*, it is assumed that the price of a security is a direct reflection of all information available and that everyone has some degree of access to this information. According to the principles of EMH, the markets are efficient and react instantaneously to new information by immediately incorporating it into the share price. A different perspective on prediction comes from the *random walk theory*, where prices fluctuate randomly in the short term. This theory has similar theoretical underpinnings to EMH as both contend that all public information is available to everyone and that consistently outperforming the market is an impossibility.

From these theories, two distinct trading philosophies emerged: the fundamentalist and the technical. In a fundamentalist trading philosophy, the price of a security is determined through myriad financial numbers and ratios. Numbers such as inflation, return on equity, and price to earnings ratios can all play a part in determining a stock's price.

Time-series data is not considered in a fundamental strategy, but is a critical part of technical analysis. Technicians reason that price movements are not random and that patterns can be identified. However, technical analysis is considered to be an art form and as such is subject to interpretation. Researchers also believe that there is a window of opportunity where weak prediction exists before the market corrects itself to equilibrium. Using this small window of opportunity (in hours or minutes) and an automated textual news parsing system, the possibility exists to capitalize on stock price movements before human traders can act.

Algorithmic quants

Among trading professionals there has been significant interest in the computational analysis of financial data. Their systems follow various stock parameters and are essentially automated versions of existing market

strategies—for example, look for high growth, unvalued securities, and so on—except with the ability to follow all stocks in real time. This advantage has led quants to steadily outperform market averages by 2 to 3 percent for the past several years.¹

While the exact strategies used are a closely guarded secret, some quantitative funds do disclose the parameters they track. The number and weights assigned to these parameters fluctuate frequently to keep pace with market conditions and to tweak model performance. Quant programs are also becoming a part of the individual investor's toolbox. For example, investors can use Wealth-Lab Pro software to track upward of 600 parameters through 1,000 preset investment strategies.²

The number of quant funds increased from just a few in 2001 to more than 150 by the beginning of 2006.³ These funds have also branched out, covering worldwide financial markets or focusing exclusively on a select boutique of securities.

Quants generally operate in a two-stage manner. First, researchers use a technical analysis strategy to analyze securities; securities that do not meet basic criteria are removed from further analysis. Second, the quantitative algorithm rank orders the remaining stocks. Figure 1 illustrates a brief taxonomy of technical stock prediction algorithms used by quants as well as their discipline of origin. While this figure is not an exhaustive list, it highlights some of the more important algorithms.

Artificial intelligence. Artificial intelligence has mostly contributed algorithms that deal with optimization and machine learning. Examples such as genetic algorithms, support vector machines (SVMs), and neural networks all take input parameters from financial securities and return predictions based on the hidden patterns within the historical data. However, most of these techniques have been constrained to either identifying the most relevant parameters or evaluating stock data in terms of a direction of movement. Genetic algorithms use a global search and optimization approach to identify the parameters that have the greatest impact on stock price performance.⁴ SVM, a machine learning algorithm, can classify the potential stock price into likely movement directions such as rise, drop, or no recommendation. Neural networks function by weighting various stock parameters. All of these methods performed marginally better than chance in prior research.⁴⁻⁶

Statistical approaches. Statistical approaches use simulations and probability methods such as Monte Carlo simulations and game theory.⁷

In Monte Carlo simulations, the problem of price prediction is too difficult to approach directly, so input parameters are given a series of suitable random numbers and are observed for how close they arrive at the predicted value.⁸

In game theory, price prediction is modeled in terms of strategies and potential payoff. Theoretically, the players in the game will evaluate other players' strategies and adopt a stance that will earn them the best payoff. However, these types of systems do not function well in stock market prediction because of new entrants, constantly changing strategies from other players, and the inherent difficulty in predicting price changes.

Mathematical approaches. These techniques borrow heavily from the areas of applied mathematics and econophysics. This branch of predictive algorithms uses more complex mathematical formalisms, such as percolation methods, log-periodic oscillations, and wavelet transforms to model future prices.⁸

Percolation methods use dimensional membranes to constrict trading actions and price movements. In one such example, a lattice of traders is modeled, where a cluster of traders indicate a single company, and at each time interval traders are given the choice to buy, sell, or sleep. This method is then used to model the supply and demand of securities and the potential impact on security prices.

Log-periodic oscillations use long-term historical data to describe macro movements in the market, such as impending crashes and market bubbles. While it has been suggested that previous "crash" predictions from this model had more to do with luck, market psychology would make these oscillations more pronounced through rapid sell-offs in the face of an anticipated crash.⁸

In wavelet transforms, input parameters are consecutively sampled to provide a finer-grained resolution into the microscopic movements that comprise the input signal. These successive filters can then be analyzed to provide parameter relation insights.

Financial news representation

To address the weaknesses of current quant systems and obviate some of the risks associated with unexpected news, researchers have focused on learning patterns from financial news articles and making predictions from them.⁹

The *bag of words* approach has emerged as a standard representation in textual financial research because of its ease of use. This process involves removing stopwords such as conjunctions and declaratives from the text and using what remains as the textual representation. While this method has been popular, it encounters noise-related issues associated with seldom-used terms and problems of scalability where immense computational power is required for large datasets.

To improve on many of the representational and scalability problems, noun phrases retain only the nouns and *noun phrases* within a document and have been found to adequately represent the important article concepts. *Named entities* is a representational technique that extends noun phrases by selecting the proper nouns of an

article that fall within well-defined categories. This process uses a semantic lexical hierarchy as well as a syntactic/semantic tagging process to assign terms to categories. The Message Understanding Conference (MUC-7) *information retrieval task* describes selected categorical definitions, encompassing the date, location, money, organization, percentage, person, and time entities. This method allows for better generalization of previously unseen terms and removes many of the scalability problems associated with a semantics-only approach.

A fourth representational technique is *proper nouns*, which functions as an intermediary between noun phrases and named entities. This representation is a subset of noun phrases and a superset of named entities, without the constraint of predefined categories. This representation removes the ambiguity associated with those particular proper nouns that could be represented by more than one named entity category or fall outside one of the seven defined named entity categories.

Once financial news articles are represented, computers can then begin the task of identifying the patterns responsible for predictable behavior.

In a comparison of different textual representation schemes, bag of words was found to be least effective in predicting future prices, whereas the proper nouns of an article were most effective for representing an article because of its concise nature.¹⁰

Another problem that arises in textual representation is *infrequent term usage*, where a term may appear only one or two times in an entire corpus. Basing predictions on infrequent term appearances can lead to unpredictable results. To reduce the impact of infrequent term use, a cutoff is often introduced where only those terms that appear multiple times in any one article are used. This strategy effectively limits the number of text features in support of scalability.

Once financial news articles are represented, computers can then begin the task of identifying the patterns responsible for predictable behavior. One accepted method, *support vector regression* (SVR), is a regression equivalent of SVMs without the aspect of classification.¹¹ Like SVMs, SVR attempts to minimize its fitting error while maximizing its goal function by fitting a regression estimate through a multidimensional hyperplane. This method is also well suited to handling textual input as binary representations and has been used in similar financial news studies.^{10,12}

RESEARCH FEATURE

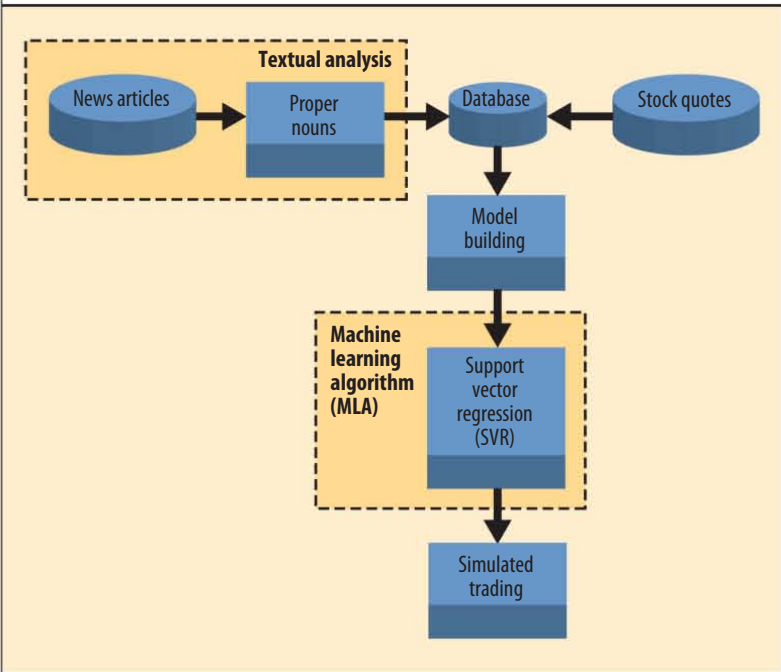


Figure 2. AZFinText system design. The design includes a proper noun representational scheme and an SVM regression derivative that outputs discrete numeric values.

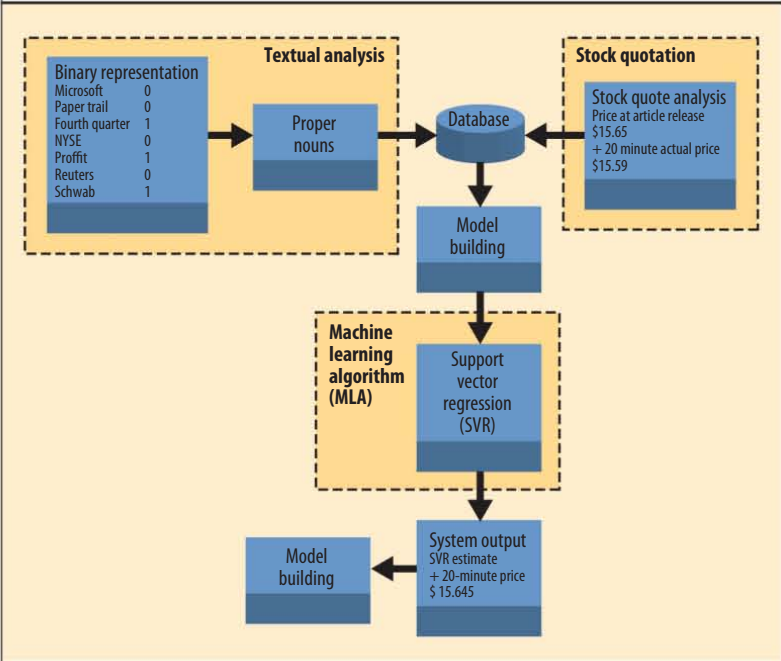


Figure 3. Sample AZFinText representation. The first step is to extract all article terms from every article in the corpora. Second, terms are identified by their parts of speech. Third, the entire set of proper nouns is represented in binary as either present or not in each individual article. Fourth, the price of the stock at the time the article was released is appended to each news article at the model building stage

ARIZONA FINANCIAL TEXT SYSTEM

To address the weaknesses of quants to unexpected information, we developed AZFinText, a machine learning

system that uses financial news articles and stock quotes as its input parameters. Figure 2 illustrates the AZFinText system design.

The differences in our design include a proper noun representational scheme and an SVM regression derivative that outputs discrete numeric values. Financial news articles are gathered from Yahoo Finance and are represented by their proper nouns, such as specific people or organizations. To limit the size of the feature space, we select only proper nouns that occur three or more times in a document.

Once the text has been represented and per-minute stock quotation data has been obtained, the next step is to build a machine-learning model. In prior work at the University of Arizona, various data models were tested with this goal in mind. Our results showed that using both the article terms and the stock price at the time of article release led to the best predictive results.¹³ After building the model, the data is used to train the machine-learning algorithm, and the results are evaluated.

Figure 3 shows an AZFinText system usage example. The first step is to extract all article terms from every article in the corpora. Second, terms are identified by their parts of speech. Third, the entire set of proper nouns is represented in binary as either present or not in each individual article. Fourth, the price of the stock at the time the article was released is appended to each news article at the model building stage.

For this particular article, the price of Schwab stock was \$15.65 at the time of article release and the +20 minute stock price was \$15.59. We selected the +20-minute interval to make our predictions because of its prior representation as a small window of opportunity in textual financial research.^{10,13}

Once the model is built, machine learning takes place via the SVR algorithm. The SVR is fed a matrix of proper nouns, coded in binary as present or not in the article, as well as the price of the stock at the time the article was released. This is done for each textual financial news article, and the SVR component makes a discrete prediction of what the +20 minute stock should be. In this instance, the output price was \$15.645.

After training, we analyze the data using a simulated trading engine that invests \$1,000 per trade and will buy or short the stock if the predicted +20-minute stock price

is greater than or equal to 1 percent movement from the stock price at the time the article was released.^{5,10,13} Any bought/shorted stocks are then sold after 20 minutes.

RESEARCH TESTBED

For our experiment, we selected a consecutive five-week research period from 26 Oct. 2005 to 28 Nov. 2005 and limited the scope of activity to companies listed in the S&P 500 as of 3 Oct. 2005. This five-week period of study yielded 9,211 financial news articles, which is comparable to prior studies.^{10,13} Articles gathered during this period were further restricted to occur between the hours of 10:30 a.m. and 3:40 p.m. While trading starts at 9:30 a.m., we felt it was important to reduce the impact of overnight news on stock prices and selected a period of one hour to allow prices to adjust. The 3:40 p.m. cutoff for news articles was selected to disallow any +20-minute stock predictions to occur after market hours. A further constraint was introduced to reduce the effects of confounding variables, where two articles on the same company cannot exist within 20 minutes of each other or both will be discarded.

This process reduced the 9,211 candidate news articles to 2,809, where the majority of discarded articles occurred outside market hours. Similarly, 10,259,042 per-minute stock quotations of the same trading period were gathered from a commercial system.

In the model-building stage, financial news articles were aggregated by their industry sectors prior to training. Financial trading analyses often use sector-based comparisons to evaluate individual company performance. For AZFinText, articles are partitioned using the Global Industry Classification Standard (GICS) classification system developed by Morgan Stanley. Companies from the S&P 500 are assigned an eight-digit GICS classifier that is used to identify sector, industry group, industry, and subindustry categories. Articles in each GICS sector (10 sectors in total) were then sent to AZFinText separately for tenfold cross-validation. Each sector averaged 281 articles, with a standard deviation per category of 160.8.

TESTING AZFINTEXT:
EXPERIMENTAL RESULTS

With more than 90 quant funds operating for a full year at the time of our study, we selected the top 10 funds according to their trailing one-year returns³ to compare against AZFinText during our trading period. We also compared our AZFinText system against the S&P 500 index, which is the industry benchmark of performance.

As Table 1 shows, AZFinText had an 8.50 percent return on trades versus the S&P 500 of 5.62 percent during the same period. Comparing AZFinText against the top 10 quants shows AZFinText performing well, outperforming six of the top 10 quant funds. The four better performing quants were trading in the Nikkei and gold markets

Table 1. Simulated trading results of the top 10 quants.

Fund	Return (%)
ProFunds Ultra Japan Inv (UJPIX)	24.73
ProFunds Ultra Japan Svc (UJPSX)	24.59
American Century Global Gold Adv (ACGGX)	12.96
American Century Global Gold Inv (BGEIX)	12.93
AZFinText	8.50
Quantitative Advisors Emerging Markets Insti (QEMAX)	8.16
Quantitative Advisors Emerging Markets Shs (QFFOX)	8.15
S&P 500 Index	5.62
Lord Abbett Small-Cap Value Y (LRSYX)	5.22
Lord Abbett Small-Cap Value A (LRSCX)	5.19
Quantitative Advisors Foreign Value Insti (QFVIX)	4.99
Quantitative Advisors Foreign Value Shs (QFVOX)	4.95

Table 2. Simulated trading results of S&P 500 quants.

Fund	Return (%)
AZFinText	8.50
Vanguard Growth & Income (VQNPX)	6.44
BlackRock Investment Trust Portfolio Inv A (CEIAX)	5.48
RiverSource Disciplined Equity Fund (ALEIX)	4.69

whereas AZFinText was constrained to the companies in the S&P 500. In making a more direct performance comparison, Table 2 shows the trade returns of AZFinText versus several quant funds that are also operating within the S&P 500.

As Table 2 shows, AZFinText performed better than its peer quant funds. AZFinText's success came mostly from making predictions from financial news articles and stock quotes, whereas quants used sophisticated mathematical models on a large set of financial variables. We believe that our research helps identify a promising research direction in financial text mining. However, more research is critically needed.

Given its indifference to the internal fiscal makeup of the companies traded, AZFinText's performance against existing quant funds is surprisingly robust. We believe that this approach may encourage quant traders to incorporate a financial news analysis engine into existing strategies. Future quant funds can be more flexible and potentially more robust by obviating risks that are captured by unexpected news events. Future directions for this research include relaxing certain assumptions and carefully testing their impact on prediction as well as testing newer machine-learning techniques such as probabilistic modeling—for example, Gaussian process—and adaptive boosting classifiers such as adaboost. ■

RESEARCH FEATURE

References

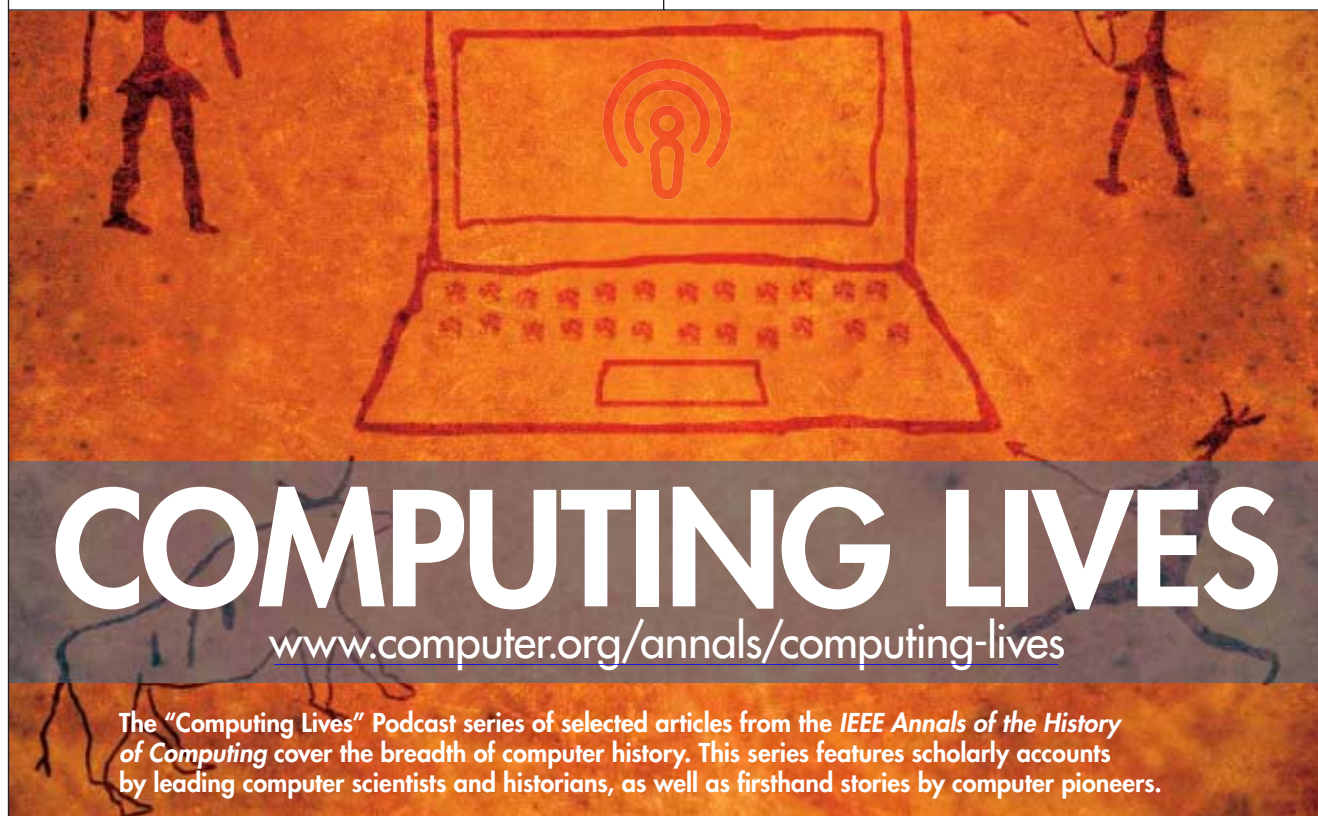
1. Z. Jelveh, "How a Computer Knows What Many Managers Don't," *The New York Times*, 9 July 2006.
2. A. Lucchetti and J. Lahart, "Your Portfolio on AutoPilot: Brokerages Roll Out Software to Automate Trading Strategies: Risks of Becoming a 'Quant'," *Wall Street J.*, 30 Sept. 2006, B1.
3. K. Burke, "Not the Man, but the Machine", 2006; www.registeredrep.com/moneymanagers/finance_not_man_machine/index.html.
4. B. LeBaron, W.B. Arthur, and R. Palmer, "Time Series Properties of an Artificial Stock Market," *J. Economic Dynamics and Control*, vol. 23, nos. 9-10, 1999, pp. 1487-1516.
5. G.P.C. Fung et al., "News Sensitive Stock Trend Prediction," *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD 02)*, LNCS 2336, Springer, 2002, pp. 481-493.
6. Y. Yoon and G. Swales, "Predicting Stock Price Performance: A Neural Network Approach," *Proc. 24th Hawaii Int'l Conf. System Sciences (HICSS-24)*, IEEE CS Press, 1991, vol. 154, pp. 156-162.
7. G. Cai and P. Wurman, "Monte Carlo Approximation in Incomplete Information: Sequential Auction Games," *Decision Support Systems*, vol. 39, no. 2, 2005, pp. 153-168.
8. D. Stauffer, "EconoPhysics—A New Area for Computational Statistical Physics?" *Int'l J. Modern Physics C*, vol. 11, no. 6, 2000, pp. 1081-1087.
9. R.J. Brachman et al., "Mining Business Data," *Comm. ACM*, vol. 39, no. 11, 1996, pp. 42-48.
10. R.P. Schumaker and H. Chen, "Textual Analysis of Stock Market Prediction Using Financial News Articles," *Proc.*

12th Americas Conf. Information Systems (AMCIS 06); www.icadl.org/intranet/papers/Textual%20Analysis%20of%20Stock%20Market.pdf.

11. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.
12. F. Tay and L. Cao, "Application of Support Vector Machines in Financial Time Series Forecasting," *Omega*, Aug. 2001, pp. 309-317.
13. M. Mittermayer, "Forecasting Intraday Stock Price Trends with Text Mining Techniques," *Proc. 37th Hawaii Int'l Conf. System Sciences (HICSS-37)*, IEEE CS Press, 2004, pp. 64-74.

Robert P. Schumaker is an assistant professor in the Information Systems Department at Iona College, New Rochelle, New York. His research interests include stock price prediction, natural-language systems, and textual analysis techniques. Schumaker received a PhD in management from the University of Arizona. He is a member of the IEEE and the ACM. Contact him at rschumaker@iona.edu.

Hsinchun Chen is the McClelland Professor of Management Information Systems at the University of Arizona. His research interests include intelligence analysis, knowledge management, and Web computing. Chen received a PhD in information systems from New York University. He is a Fellow of the IEEE and the AAAS. Contact him at hchen@eller.arizona.edu.

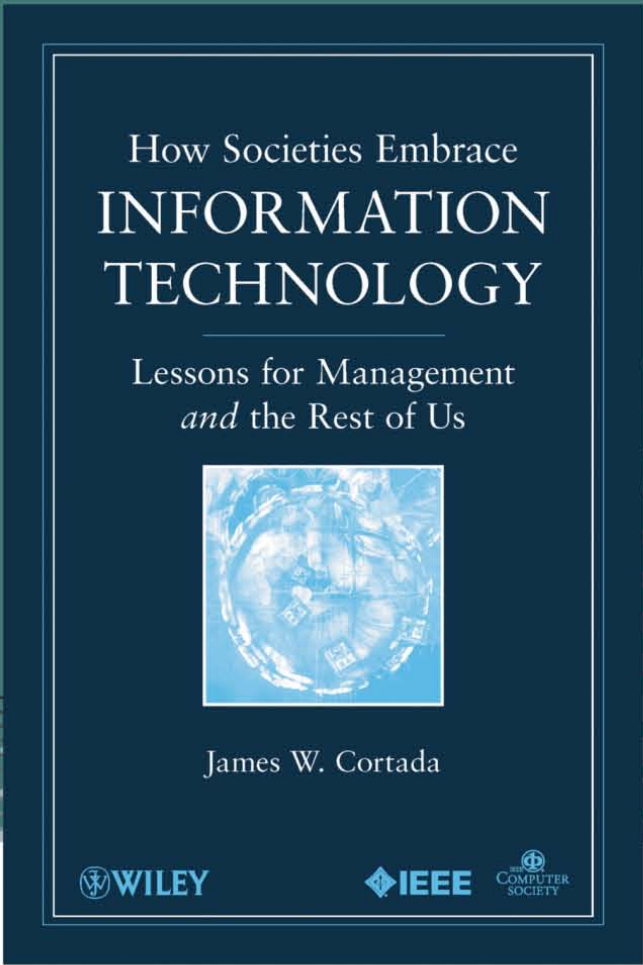


COMPUTING LIVES

www.computer.org/annals/computing-lives

The "Computing Lives" Podcast series of selected articles from the *IEEE Annals of the History of Computing* cover the breadth of computer history. This series features scholarly accounts by leading computer scientists and historians, as well as firsthand stories by computer pioneers.

NEW TITLE FROM WILEY & **CSPress**



The new book by James W. Cortada, author of *The Digital Hand*, discusses the role of corporations and governments in the over \$2 trillion annual world-wide expenditure in acquisition and deployment of IT.

The author's royalties will be donated to the IEEE Computer Society's Educational Activities Board.

978-0-470-53498-4 • November 2009
Paperback • 272 pages • \$29.95
A Wiley-IEEE Computer Society Press Publication

To Order



North America
1-877-762-2971

Rest of the World
+ 44 (0) 1243 843291

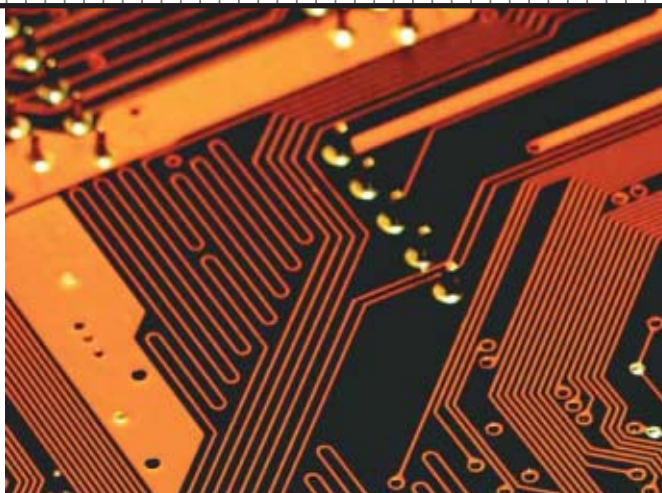
15% Off
for CS Members



IEEE  **computer society**
www.computer.org

RESEARCH FEATURE

Online Security Threats and Computer User Intentions



➔ Thomas F. Stafford and Robin Poston, *University of Memphis*

Although computer users are aware of spyware, they typically do not take protective steps against it. A recent study looks into the reasons for this apathy and suggests boosting users' confidence in installing and operating antispyware solutions as an effective remedy.

Your identity can be stolen, your account passwords compromised, your computer commandeered and converted to a spam zombie, your credit card number copied and sent to malicious third parties.¹⁻³ These are just some of the threats unwary computer users face from spyware, which constitutes a potent and growing security problem: the surreptitious installation of remote monitoring software that observes and tracks user activities and information, and reports this to outside parties over the user's Internet connection.^{2,3}

The Federal Trade Commission defines spyware as “software that aids in gathering information about a person or organization without their knowledge, and that may send that information to another entity without user consent.”⁴ Spyware guru Steve Gibson considers it to be anything that surreptitiously utilizes a computer's Internet “back channel” to communicate with an external server (www.grc.com/intro.htm).

As Table 1 shows, the harm resulting from unauthorized remote monitoring ranges from invasion of privacy, to operational losses involving degradation of computer performance, to economic losses stemming from identity theft and outright larceny. Some types of spyware simply appropriate processor cycles and network bandwidth to send out spam from user machines, while others obtain

passwords and account numbers to steal services and financial resources. The latest types of spyware, which continue to evolve, includes robust malware conglomerations that include simultaneous infection by Trojan horses, rootkits, and communications software designed to exploit peer-to-peer networking.⁵

While spyware is indeed a problem, we believe it is a symptom of a much bigger danger: user apathy. Internet users are aware of spyware but strangely reluctant to engage in safe computing practices or use effective antispyware applications.¹⁻³ One survey of 252 consumers revealed that only 22 percent consistently used spyware solutions,¹ despite their widespread availability in over-the-counter and shareware security packages.^{1-3,5,6} Lack of knowledge may be partly responsible, but so few people consistently use antispyware software that motivation is clearly the main problem.

To understand why, we examined computer users' failure to protect their online security and privacy through the prism of *protection motivation theory* (PMT), which has been widely used to explain individuals' lack of motivation to defend against threats to personal health and safety such as drunk driving, smoking, and infectious disease transmission.⁷⁻⁹ PMT emphasizes the importance of the nature of a threat to personal security coupled with self-perceived assessments of the ability to carry out an effective solution.

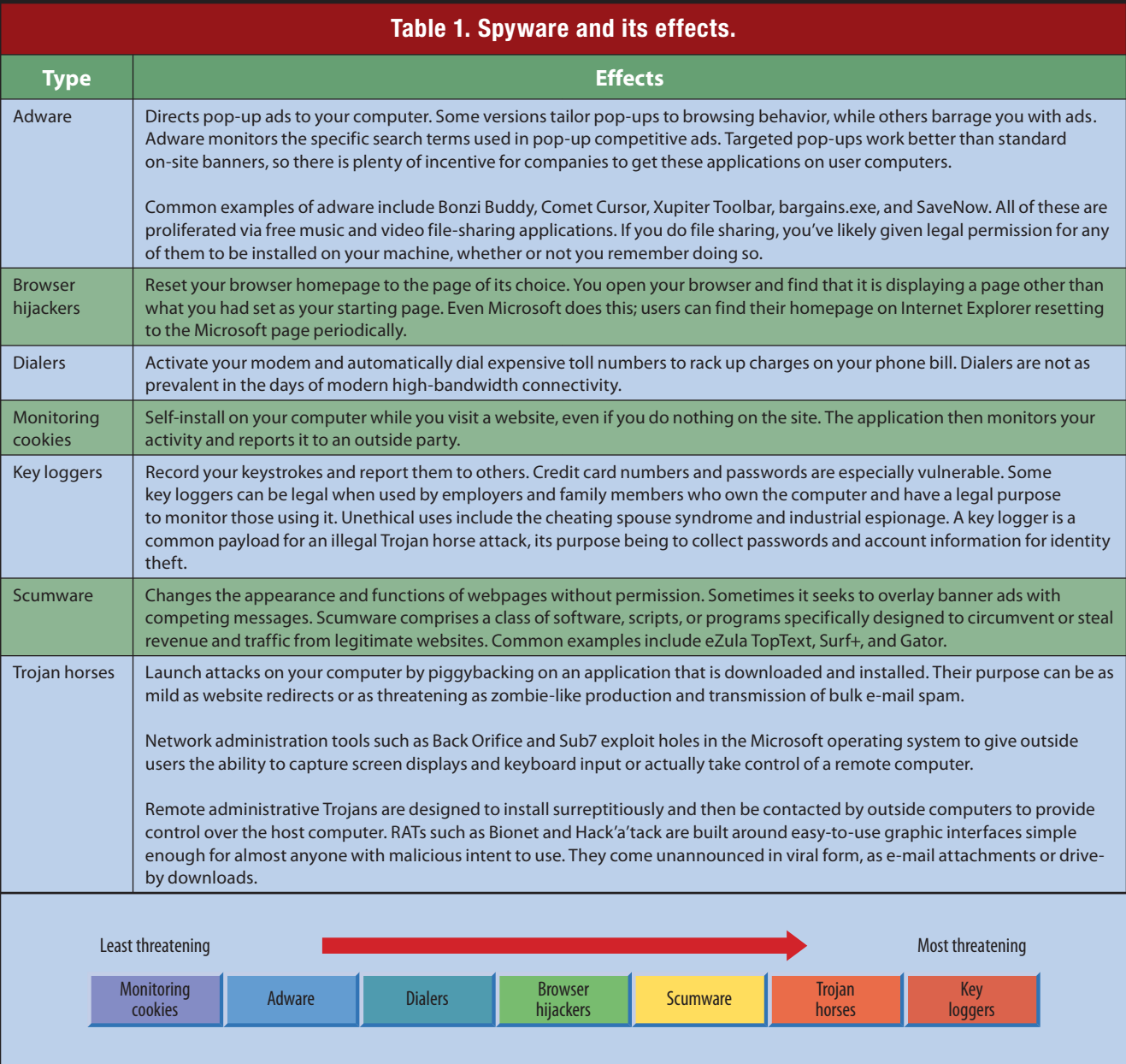


Figure 1. Users' perceived severity of spyware threats.

SPYWARE THREAT AWARENESS

Remote monitoring applications are designed to surreptitiously intrude on user privacy and compromise sensitive personal information.^{1-3,5,6} Users do not consciously install spyware—many are fooled into doing so by vaguely worded click-wrap user agreements accompanying free downloads of software such as graphic-utility and file-sharing applications. Some spyware self-installs during user visits to questionable websites.⁶ Developers of these noxious security threats brag of installations on 90 percent of online personal computers,¹ highlighting the omnipresent nature and severity of the threat.

The widespread presence of spyware threatens users

with loss of control over their personal information and computer system resources, as well as unauthorized third-party access to sensitive private data.¹⁻³ As Figure 1 shows, users perceive some types of spyware to be more dangerous than others.

When Internet service provider America Online (AOL) was preparing to introduce a spyware protection application to its users, it commissioned a study to assess customer awareness of the spyware threat and its detrimental outcomes.³ This study, which drew on a widely cited review of spyware symptoms and causes,² found that spyware was the third most recognized online threat after viruses and spam. Seventy-five percent of users claimed to be aware of spyware and 74 percent considered it a personal threat.³

RESEARCH FEATURE



Figure 2. Possible protective actions against spyware.

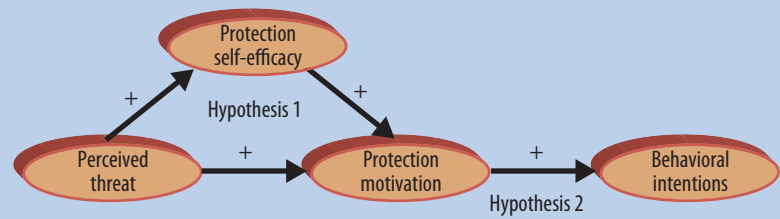


Figure 3. Protection motivation theory suggests two hypotheses to explain individuals' lack of motivation to defend against perceived threats, including spyware.

Internet users thus seem to understand that spyware is a very real danger to their personal computers and information, yet most do not take protective actions against it. The same study found that 55 percent of respondents did not use or did not know if they used spyware protection, and 72 percent had no plans to install antispyware software.³ The question is, why aren't users more motivated to defend themselves and their computers against a threat that they are aware of? The question of whether it is a matter of user apathy or lack of user confidence formed the basis of our investigation.

PROTECTION MOTIVATION

In general, people must be sufficiently motivated to take protective action against a threat.¹⁰ One way to achieve this is to stimulate emotional tension by delivering messages that induce fear in combination with messages about

an effective solution.⁷ Such "fear appeals" have been widely used for decades to sell insurance, promote socially responsible behavior, and elect political candidates to high office.^{7,8} This concept is the basis for PMT,⁸ which posits that people will take action to protect themselves only if they have knowledge of a fearful threat as well as confidence in the ability to implement the solution.¹⁰

With respect to the spyware threat, potential responses span a continuum of actions, some more technologically demanding than others, as shown in Figure 2. Our focus was on why users fail to install antispyware applications, and PMT suggests that users either are not aware of such applications and their effectiveness, or they are not convinced they can properly install and use them.

According to PMT, people confronted with a threat assess the likelihood of its impact on them and their ability to respond effectively.^{8,10,11} These judgments form the basis of their motivational attitude toward self-protection, which results in actions to adopt the solution.⁸⁻¹⁰ Appraisal of a threat induces a fear state, which increases the level of attention and subsequent need to take protective steps.⁹

In the context of computer security, spyware induces concern about the loss of privacy and sensitive personal information.¹⁻³ Users should be motivated to use antispy-

Table 2. AOL Opinion Place demographics.

Category	Questionnaire respondents (percent)	AOL members (percent)	General Internet users (percent)	US population (percent)
Gender				
Male	35	45	49	48
Female	65	55	51	52
Age				
18-24	9	16	14	12
25-34	25	18	21	20
35-44	30	27	26	21
45-54	23	24	23	15
55+	12	16	15	32
Married	58	62	66	57

Note: AOL demographic profiles are provided by and used with permission from AOL Opinion Place and are the result of ongoing in-house demographic profile studies of their membership and the general population of Internet users to support commercial research.

Table 3. Survey questions.			
Questions	Summated mean	Standard deviation	Coefficient alpha score
Perceived threat I know that I probably have a spyware problem if ... Threat 1. My computer seems to be slower than usual. Threat 2. There are more pop-up ads on my computer than usual. Threat 3. The homepage of my browser changes unexpectedly. Threat 4. Odd sites come up on my computer when I do online searches.	19.50	6.53	.874
Protection self-efficacy I would be likely to use spyware protection software if ... Cope 1. If there was no one around to tell me what to do as I used the software. Cope 2. If I had never used a software package like it before. Cope 3. If I had only a users' guide for reference. Cope 4. If I had seen someone else using it before trying it myself. Cope 5. If I could call someone for help if I got stuck. Cope 6. If someone else helped me get started. Cope 7. If I had a lot of time to use it. Cope 8. If I just had the built-in help with the software for assistance. Cope 9. If someone showed me how to use it first. Cope 10. If I had used similar packages before this one for the same purpose.	46.99	12.45	.867
Protection motivation As regards my potential use of spyware protection ... PM 1. If I continue without spyware protection, I will have problems. PM 2. If I use spyware protection, it will greatly increase my computer security.	7.59	3.85	.817
Behavioral intentions As regards the likelihood of taking protective action ... BI 1. I intend to immediately install spyware protection on my computer. BI 2. Within a week or so I will install spyware protection on my computer.	11.03	2.77	.714

ware software if they feel that the threat is severe, they are a likely target, and they can defend against the threat. PMT thus suggests two possible hypotheses:

- Hypothesis 1:* Protection self-efficacy mediates the influence of perceived threats on protection motivation.
- Hypothesis 2:* Protection motivation directly influences behavioral intention to take protective action.

Figure 3 illustrates these hypotheses.

TESTS OF HYPOTHESES

AOL provided us with data from its study on customer perceptions of spyware and their capabilities of dealing with the threat.

Sample

We obtained 1,006 completed questionnaires from AOL's research website, Opinion Place (www.opinionplace.com), which rewarded respondents with 200 American Airlines frequent flyer points. As Table 2 shows, the characteristics of AOL Opinion Place respondents are fairly representative of US Internet users.

As Table 3 shows, our analysis focused on four key variables: perceived threat, protection self-efficacy, protection motivation, and behavioral intentions. All questionnaire measures used a seven-point scale ranging from 1 = strongly disagree to 7 = strongly agree.

The first variable measured users' perceptions of the spyware problem. We developed four specific questions to assess this perception through a combination of focus group sessions and pilot testing.³ Working with a group of local working professionals who were Internet users, we developed structured questions designed to encourage the participants to share their opinions about the spyware threat. We then created a pool of potential threat-perception-scale items based on the commonly mentioned opinions raised by focus group members, and qualified

RESEARCH FEATURE

Predicting protection motivation

Variable	F (1,1004)
Perceived threat	48.211*
Protection self-efficacy	313.915*

Predicting protection self-efficacy

Variable	F (1,1004)
Perceived threat	57.329*

Predicting behavioral intentions

Variable	F(1,1004)
Protection motivation	259.063*

Relationships among variables

Relationships among variables	Strength of relationships (β)	Significance of relationships (t)	Variance explained (r ²)
Perceived threat → Protection motivation	.214	6.943*	.046
Perceived threat → Protection self-efficacy	.232	7.572*	.054
Protection self-efficacy → Protection motivation	.488	17.718*	.238
Protection motivation → Behavioral intentions	.453	16.095*	.046

* Values of F significant at *p* < .001

Figure 4. Regression results among variables. The results support both hypothesis 1 and 2.

these in a survey of university students using principal-components factor analysis to reduce the pool of items to the most characteristic descriptors.

The second variable assessed users' self-efficacy in protecting themselves from spyware, in the form of 10 questions drawn from a prominent study of self-efficacy and computer training¹¹ and modified to reflect the potential installation and use of an antispyware application. The third and fourth variables related to users' motivations in taking protective action against spyware. For both we used a two-question-scale format drawn directly from previous protection motivation research studies⁸ and adapted to the spyware context.

We summated the scale questions for each variable into composite scores for subsequent hypothesis testing. For each variable, Table 3 provides a summated means (average composite score across all subjects), standard deviation, and coefficient alpha score. Coefficient alpha scores indicate adequate reliability of the questions to measure the variable of interest when scores exceed a threshold of 0.7.

Hypothesis tests

To test hypothesis 1, we examined both direct and indirect relationships among the protection motivation, perceived threat, and protection self-efficacy variables. If the relationships among perceived threat and protection motivation and the intervening mediator, protection self-efficacy, are all positive and significant, then hypothesis 1 is supported.¹²

Using the summated question scores for each variable, we regressed protection motivation on perceived threat and protection self-efficacy, then on perceived threat only. As Figure 4 shows, the regression relationships associated with each of the three paths necessary to establish protection self-efficacy as an intervening mediator are all significant: Perceived threat fi protection motivation (*b* = .214, *t* = 6.943, *p* < .001); perceived threat fi protection self-efficacy (*b* = .232, *t* = 7.572, *p* < .001); protection self-efficacy fi protection motivation (*b* = .488, *t* = 17.718, *p* < .001). Thus, hypothesis 1 is supported.

To test hypothesis 2, we regressed behavioral intentions on protection motivation. As Figure 4 shows, there is a strong and significant relationship between protection motivation and behavioral intentions to use antispyware applications (*b* = .453, *t* = 16.095, *p* < .001). Thus, hypothesis 2 is also supported.

EVALUATION

Our study examined the problem of why computer users do not take action to protect themselves against outside parties monitoring their activities or potentially stealing their personal information or identities. The survey data collected for this study supported the research model based on PMT, which predicted that users' protection self-efficacy would mediate the relationship between their perceptions of a security threat and their motivation to protect themselves. The data also supported the notion that users with greater motivation for protection are more likely to take protective action.

Limitations

While providing a deeper understanding of the mechanisms facilitating protective action against spyware, our study was limited in certain respects. It might seem intuitive that sampling random Internet users would be a better choice than AOL participants rewarded with frequent flyer points and living in the US, but we believe this sample provided a solid foundation for testing the research model. We acknowledge that these respondents will likely differ from non-AOL users, non-American Airlines frequent-flyer members, and possibly non-US users in terms of online access and usage experiences. However, as Table 2 illustrates, there were demographic similarities between AOL and non-AOL groups. Our sample comprises data from

actual online users and reflects their perceptions and attitudes. Nonetheless, future research must sample other groups of Internet users to define boundary conditions.

In a similar vein, the survey asked participants about general spyware protection concepts rather than focusing on any one type of antispware application. Antispware technologies vary considerably, and this may differentially influence protection self-efficacy judgments. Particular antispware solutions could affect protection motivations and the resulting behavioral intentions in distinct ways. Because our study sought to gain a big picture of users' apparent apathy toward spyware protection, we chose to operationalize the variables at a broad level. This decision, however, prevented us from determining how specific antispware technologies and protection motivation attitudes influence user intentions. Consequently, whether important subtleties exist remains an open empirical issue.

We also recognize that there are two key ways to deal with spyware. One, as we have investigated, is to raise awareness of the problem and encourage the adoption of protective applications. This approach makes sense when dealing with less sophisticated users such as those found in a sample of AOL customers. Another important method is to promote safer computing behaviors to avoid spyware downloads altogether, resulting in fewer infestations.¹³ We suspect this would be equally effective as a preventive measure, but primarily with more knowledgeable computer users. Additional research is needed to examine which alternative works best with different user groups.

Finally, the use of cross-sectional data cannot provide conclusive evidence of how computer users' attitudes change over time. Although the data collection procedure is consistent with other studies that examined protection motivation attitudes,^{8,9} future research should utilize other data collection methods, such as longitudinal and experimental designs, to address this issue. Moreover, common response bias could have surfaced given the exclusive use of survey data. Although the questions exhibited reliable measurement of the variables, the relationships among variables could have been inflated.

Implications

Despite these limitations, our study has important implications. The role of self-efficacy for the use of antispware protection applications suggests a step that Internet service providers and other concerned parties can take to achieve better user responses to spyware threats online. Our findings support the notion that coaching users in the process of acquiring and installing protective software applications can increase their self-perceived capabilities to take effective action.^{10,11}

Much of what is known about self-efficacy in computing derives from user training studies,¹³ where the concept of guided learning is a critical success factor in such training

exercises. This appears to be equally important in supporting users to act against online security threats. The more comfortable users are with a protective software application, the more likely they are to install it and use it to combat spyware. The comfort level can only increase with familiarity, and this, in turn, increases through guided experiences installing antispware applications. For other applications, companies like AOL have successfully used the guided training process to increase user acceptance and use of new software applications,³ and it seems quite likely that a similar approach tailored to the specific task of downloading and installing antispware applications will achieve similar successes.

The insights from our study attest to PMT's predictive potential in the context of computer security. We found that some of the previously established relationships between the motivations to take action and the communication of threats hold. In the presence of security threats, not only is protection self-efficacy important, but so is fear of the dire consequences of not taking protective measures. Awareness of the threat and the ability to do something about it are particularly salient in understanding user behavior with respect to antispware applications. As such, we believe that the study's findings can be generalized to other types of protective actions—for example, signing up for spyware protection via an Internet service provider or avoiding actions that encourage spyware downloads.

Our research findings provide guidance for and validation of ways to motivate computer users to take action against security threats such as spyware. For example, Internet providers, software companies, and computer makers should continue to make an effort to increase users' awareness of online threats, encourage their self-efficacy in taking action, and provide them with user-friendly tools. In terms of designing effective training programs for online security, our findings suggest that teaching users how to use protective solutions—building confidence in their ability to take action—is as important as spending time conveying the consequences of not protecting computers and personal information. **■**

References

1. X. Zhang, "What Do Consumers Really Know about Spyware?" *Comm. ACM*, Aug. 2005, pp. 44-48.
2. T.F. Stafford and A. Urbaczewski, "Spyware: The Ghost in the Machine," *Comm. Assoc. for Information Systems*, vol. 14, 2004, pp. 291-306.
3. R. Poston, T.F. Stafford, and A. Hennington, "Spyware: A View from the (Online) Street," *Comm. ACM*, Aug. 2005, pp. 96-99.
4. R.R. Urbach and G.A. Kibel, "Adware/Spyware: An Update Regarding Pending Litigation and Legislation," *Intellectual Property & Technology Law J.*, July 2004, pp. 12-17.

RESEARCH FEATURE

Call for Articles

IEEE Software seeks practical, readable articles that will appeal to experts and nonexperts alike. The magazine aims to deliver reliable, useful, leading-edge information to software developers, engineers, and managers to help them stay on top of rapid technology change. Topics include requirements, design, construction, tools, project management, process improvement, maintenance, testing, education and training, quality, standards, and more.

Author guidelines: www.computer.org/software/author.htm
Further details: software@computer.org
www.computer.org/software

**IEEE
Software**

5. B. Smith, "A Storm (Worm) Is Brewing," *Computer*, Feb. 2008, pp. 20-22.
6. S. Fox, "Spyware," 6 July 2005, Pew Internet and American Life Project; www.pewinternet.org/Reports/2005/Spyware.aspx.
7. C.I. Hovland, I.L. Janis, and H.H. Kelly, *Communications and Persuasion: Psychological Studies of Opinion Change*, Yale Univ. Press, 1953.
8. R.W. Rogers, "A Protection Motivation Theory of Fear Appeals and Attitude Change," *J. Psychology*, vol. 91, 1975, pp. 93-114.
9. J.F. Tanner Jr., J.B. Hunt, and D.R. Eppright, "The Protection Motivation Model: A Normative Model of Fear Appeals," *J. Marketing*, July 1991, pp. 36-45.
10. B.J. Fogg, *Persuasive Technology: Using Computers to Change What We Think and Do*, Morgan Kaufmann, 2002.
11. D.R. Compeau and C.A. Higgins, "Computer Self-Efficacy: Development of a Measure and Initial Test," *MIS Quarterly*, June 1995, pp. 189-211.
12. R.M. Baron and D.A. Kenny, "The Mediator-Moderator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations," *J. Personality and Social Psychology*, Dec. 1986, pp. 1173-1182.
13. N. Good et al., "Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware," *Proc. 2005 Symp. Usable Privacy and Security (SOUPS 05)*, ACM Press, 2005, pp. 43-52.

Thomas F. Stafford is an associate professor in the Department of Management Information Systems, Fogelman College of Business and Economics, University of Memphis. His research interests include human-computer interaction, spyware, and the application of technology in supply chain relationships. Stafford received a PhD in management information systems from the University of Texas at Arlington and a PhD in marketing from the University of Georgia's Terry College of Business Administration. He is a member of the Association for Information Systems (AIS), the Academy of International Business, the Academy of Management, the Decision Sciences Institute, the Institute for Operations Research and Management Science, and the Global Information Technology Management Association. Contact him at tstaffor@memphis.edu.

Robin Poston is a Systems Testing Research Fellow at the FedEx Institute of Technology and an associate professor in the Department of Management Information Systems, Fogelman College of Business and Economics, University of Memphis. Her research focuses on understanding how individuals use credibility information in decision support systems, Web-based knowledge management applications, Internet-based dissemination of information, and systems testing management. Poston received a PhD in management information systems from Michigan State University. She is a member of AIS. Contact her at rposton@memphis.edu.



Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.

Computer Society Programs Serve Members



The IEEE Computer Society has been committed to fostering innovation in computing for more than six decades. With more than 40 percent of its members living and working outside the US, the Computer Society strongly encourages and facilitates international cooperation, communication, and information exchange.

In 2010, the Computer Society again offers a full catalog of periodicals that address all aspects of computer science and engineering, including 13 magazines, 12 transactions, and one letters publication. The Society also publishes cutting-edge papers and articles from hundreds of conferences, covering topics in all areas of computer science and engineering. All members receive a free subscription to *Computer* magazine as a benefit of membership.

COMPUTER SOCIETY ONLINE DIGITAL LIBRARY

Society members enjoy online access to 28 current and former Society magazines, transactions, and letters, as well as nearly 4,000 selected conference proceedings, tutorials, and scholarly books via the IEEE Computer Society Digital Library. Computer Society student members receive free access to the CSDL as a benefit of membership. Traditional single-magazine subscriptions are also available in print, online, or combined formats. Users who are not yet members of the Soci-

ety may purchase individual articles for \$19.

Visit www.computer.org/portal/web/cSDL for complete details.

VOLUNTEER OPPORTUNITIES IN PUBLICATIONS

IEEE Computer Society publications are led by volunteer editorial boards that work together with professional staff to provide the targeted, high-quality content that has been the Society's hallmark for more than 60 years.

Several titles are seeking new editors in chief for terms beginning in 2011. To find out more about these positions, see the "Society Publications Seek Volunteers for 2011-2012 Editor in Chief Terms" article.

Members can also become involved with the Society's publications program by volunteering to serve as authors and reviewers for individual magazines and journals. Visit www.computer.org/publications to learn more about opportunities at each title.

COMPUTER SOCIETY eLEARNING CAMPUS

Another free benefit of membership is the IEEE Computer Society e-Learning Campus. Computer Society members have access to more than 3,000 Web-based courses that include specialized technical classes for computing professionals, primers on office productivity, key courses in business fundamentals, and a full

suite of courses on Cisco networking technology.

Members can also access books, certification courses, and other study materials. Twice each year, a committee of volunteers reviews and selects new e-Learning Campus offerings based on survey data and usage numbers. Start exploring the campus at www.computer.org/portal/web/e-learning.

Certifications

Intended for mid-level software development and software engineering professionals, the IEEE Computer Society Certified Software Development Professional program offers the only brand-name professional credential in software development. The program requires strong performance on an examination that demonstrates mastery of a well-defined body of knowledge, as well as verification of both a solid experience base in the field and recent continuing professional education work.

The Certified Software Development Associate program is intended for beginning software developers and engineers. The CSDA is the first step toward earning full CSDP certification.

Visit www.computer.org/portal/web/getcertified for complete details including requirements, fees, and test dates.

Exams

Proficiency exams by Brainbench help demonstrate knowledge in a

COMPUTER SOCIETY CONNECTION

particular area. Members receive free access to the exams, and official certificates are available at a special member rate. Learn more at www.computer.org/portal/web/e-learning/brainbench.

Books 24x7 Collections

Computer Society professional and student members now receive free access to 600 selections from Safari Books Online. Safari offers online technical books from noted computer publisher O'Reilly and other leading publishers including Addison-Wesley, Cisco Press, FT Press, and Pren-

tice Hall. Complete details are available at www.computer.org/safari.

Another 500 technical references are available to members via Element K. Members can visit www.computer.org/portal/web/e-learning/ekbooks to access this benefit.

COMMUNITIES AND ONLINE RESOURCES

The IEEE Computer Society maintains a robust online presence via Computing Now, a one-stop source for new print and online content from the 13 peer-reviewed IEEE Computer Society magazines, as well as selec-

tions from the Society's journals and conference proceedings.

Other opportunities and services sponsored by the Computer Society include local and student chapter activities; standards groups; career resources; conference management and publications services; educational, professional, and technical panels; and a highly regarded awards program.

CONFERENCES

IEEE Computer Society technical councils, task forces, and technical committees sponsor the majority of the Society's technical meetings. The following selection of high-profile conferences is a cross-section of the many events presented by the Society this year.



1-6 March: ICDE 2010

Data engineering addresses the use of engineering techniques and methodologies in the design, development, and assessment of information systems for different computing platforms and application environments. The 26th IEEE International Conference on Data Engineering, in Long Beach, California, continues as a premier forum for presenting research results on advanced data-intensive applications and discussing issues in data and knowledge engineering. Conference participants share research solutions and cooperate to identify new issues and directions for future research and development work.

Organizers have called for submissions that address research issues in designing, building, managing, and evaluating advanced data-intensive systems and applications.

ICDE 2010 is the flagship conference of the IEEE Computer Society Technical Committee on Data Engineering. Visit the ICDE 2010 website at www.icde2010.org for full conference details, including registration information as it becomes available.

→ COSPONSORED TRANSACTIONS CALL FOR NEW EIC, REAPPOINTMENT INPUT

The IEEE Transactions on Mobile Computing steering committee seeks applicants for the position of editor in chief serving a three-year term beginning 1 January 2011. Please send materials by 1 March to Tom LaPorta, tlp@cse.psu.edu.

Marie-France Sagot, editor in chief of IEEE/ACM Transactions on Computational Biology and Bioinformatics, is currently being considered for reappointment to a second two-year term. Please send comments on her reappointment by 15 February 2010 to Metin Akay, metin.akay@asu.edu.

IEEE Transactions on Haptics editor in chief Edward Colgate is currently standing for reappointment to a second three-year term. Please send comments by 15 February 2010 to Peter Luh, peter.luh@uconn.edu.

→ SOCIETY BEGINS NOMINATIONS FOR IEEE DIVISION V DIRECTOR-ELECT


The IEEE Computer Society has begun the nominations process for candidates to serve as 2011 IEEE Division V director-elect and 2012-2013 Division V director.

Division directors represent the members of IEEE societies on the IEEE Board of Directors and the Technical Activities Board; Division V and VIII directors represent the Computer Society membership. Elections for Division V director are typically held in even-numbered years, and Division VIII elections are held in odd-numbered years. The elected representative then serves one year in the director-elect role before assuming a two-year division director term.

Past Computer Society president Michael Williams currently serves as IEEE Division V director for 2010-2011. Past Computer Society president Steve Diamond is serving as IEEE Division VIII director for 2009-2010. In a recent vote, IEEE members chose another former Computer Society president, Susan K. (Kathy) Land, CSDP, as Division VIII director-elect for 2010.

Nominations for the position of director-elect may be placed on the ballot by petition. The Computer Society Nominations Committee will propose a list of candidates by 8 January. Additional nominees may be submitted by written petition signed by one-third of the franchised Board members and received by the Computer Society Secretary no later than 26 January. A slate will be selected by the Board of Governors at its 5 February meeting. IEEE Computer Society members may propose additional names by petition in accordance with IEEE bylaws. Completed petitions must be submitted in a letter to the IEEE Board of Directors to be received at IEEE headquarters no later than noon ET Friday, 11 June.


**20-24 March:
IEEE VR 2010**

 Innovative research, groundbreaking technology, pioneering concepts, and hands-on experiences in the disciplines of virtual reality, augmented reality, and 3D user interfaces will be highlighted at IEEE Virtual Reality 2010.

Located in Waltham, Massachusetts, VR 2010 will take place in conjunction with the IEEE Symposium on 3D User Interfaces and the Symposium on Haptic Interfaces. Conference organizers have solicited papers on topics that include immersive gaming, VR systems and toolkits, augmented and mixed reality, and computer graphics techniques.

VR 2010 leads the Computer Society Technical Committee on Visualization and Graphics' annual calendar of conferences and workshops. For further details, visit <http://conferences.computer.org/vr/2010>.

**22-26 March:
ECBS 2010**

 The 17th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems is devoted to advancing the design, development, deployment, and analysis of complex systems that are largely controlled by computers.

Conference organizers have solicited papers on topics that include autonomic systems, component-based system design, ECBS infrastructure, reengineering and reuse, and verification and validation.

ECBS shares a venue with the IEEE International Conference on Engineering of Complex Computer Systems and the IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems.

Sponsored by the IEEE Computer Society Technical Committee on the Engineering of Computer-Based Systems, the 2010 conference will take

place in Oxford, UK. Visit <http://tab.computer.org/ecbs/2010> for more information.

**14-16 April:
Cool Chips XIII**

 The 2010 IEEE Symposium on Low-Power and High-Speed Chips is a high-profile international forum for presenting recent advancements in all areas of microprocessors and their applications.

This year, Cool Chips will focus on the architecture, design, and imple-

mentation of chips in three areas: low-power, high-performance processors, including "eco-processors," for multimedia, consumer electronics, healthcare, biometrics, and more; novel architectures and schemes for single core, multicore, embedded, and wireless systems; and cool software including binary translations, compiler issues, and low-power techniques.

mentation of chips in three areas: low-power, high-performance processors, including "eco-processors," for multimedia, consumer electronics, healthcare, biometrics, and more; novel architectures and schemes for single core, multicore, embedded, and wireless systems; and cool software including binary translations, compiler issues, and low-power techniques.

IPDPS 2010 participants will also have the opportunity to organize informal birds-of-a-feather sessions. Scheduled workshop topics include reconfigurable architectures, high-level parallel programming models, and system management techniques, processes, and services. Returning



The Computer Society publishes cutting-edge papers and articles from more than 300 conferences.

mentation of chips in three areas: low-power, high-performance processors, including "eco-processors," for multimedia, consumer electronics, healthcare, biometrics, and more; novel architectures and schemes for single core, multicore, embedded, and wireless systems; and cool software including binary translations, compiler issues, and low-power techniques.

Cool Chips XIII, which takes place in Yokohama, Japan, is sponsored by the IEEE Computer Society Technical Committees on Microprocessors and Microcomputers and Computer Architecture. Cosponsors include the Institute of Electronics, Information, and Communication Engineers Electronics Society, ACM Sigarch, and the Information Processing Society of Japan. For more conference information, visit www.coolchips.org.

**19-23 April:
IPDPS 2010**



The 24th International Parallel and Distributed Processing Symposium is

in 2010 is the popular PhD Forum, introduced by the IEEE Computer Society Technical Committee on Parallel Processing as an opportunity for graduate students to present their proposed or partially completed dissertation work to a broad audience of both academic and industrial researchers and practitioners.

IPDPS 2010, located this year in Atlanta, is sponsored by the TCPP in cooperation with ACM Sigarch. To learn more about IPDPS 2010, visit www.ipdps.org.

**16-19 May:
Security and Privacy 2010**



The 2010 IEEE Symposium on Security and Privacy marks the 31st annual meeting of this flagship conference. Since 1980, S&P has been the premier forum for presenting developments in computer security and electronic privacy, and for bringing together researchers and practitioners in the field. For 2010, conference organizers have solicited papers on topics that include forensics, hardware-based

COMPUTER SOCIETY CONNECTION

security, information flow, intrusion detection, security architectures, and malicious code, among others. Papers presented at the conference represent advances in the theory, design, implementation, analysis, and empirical evaluation of secure systems, both for general use and for specific application domains.

S&P, which takes place in Berkeley, California, is sponsored by the IEEE Computer Society Technical Committee on Security and Privacy in cooperation with the International Association for Cryptologic Research. See <http://oakland31.cs.virginia.edu> for complete conference details.

17-20 May:
CCGrid 2010
CCGrid

The 10th IEEE International Symposium on Cluster Computing and the Grid, the latest in a series of successful international conferences that began in 2000, provides researchers and practitioners with an opportunity to share their research and experiences in overcoming challenges in Web services and grid technology.


Several workshops and other events complement the larger conference. Workshop topics at CCGrid include resiliency in high-performance computing and challenges for

the application of grids in healthcare.

CCGrid 2010, set this year in Melbourne, is sponsored by the IEEE Computer Society Technical Committee on Scalable Computing. Taking place in conjunction with CCGrid 2010, the International Scalable Computing Challenge is also sponsored by the TCSC.

Visit www.manjrasoft.com/ccgrid2010 for program highlights and more conference information.

19-23 July:
COMPSAC and SAINT 2010

 The Internet is evolving into a pervasive and highly distributed ecosystem of connected computers, mobile devices, sensors, home appliances, and a variety of other devices.


The 2010 Symposium on Applications and the Internet and the 2010 IEEE Computer Society Conference on Computer Software and Applications will draw researchers from around the world to share new ideas and findings regarding the Internet, computer software, and their applications. Participants come from a wide spectrum of disciplines in industry, government, and academia.

SAINT organizers have called for papers on topics that include mobile

and ad hoc groupware, Internet security architectures, information integration, and universal access and interfacing.

COMPSAC organizers have called for papers that address smart software solutions in diverse areas including healthcare, energy, society, environment, and industry.

Cosponsored by the Information Processing Society of Japan, SAINT is the flagship conference of the IEEE Computer Society Technical Committee on the Internet. Visit <http://infonet.cse.kyutech.ac.jp/conf/saint10> for more information on SAINT 2010. To learn more about COMPSAC 2010, visit <http://compsac.cs.iastate.edu>. The two events share a venue in Seoul.

Proceedings from many conferences are available through the Computer Society Digital Library. IEEE Computer Society members also enjoy as much as a 25 percent discount on registration fees at Society-sponsored conferences. See www.computer.org for complete details on all IEEE Computer Society publications, conferences, symposia, technical meetings, volunteer opportunities, and other activities. 

For more information on any topic
presented in *Computer*,
visit the IEEE Computer Society
Digital Library at

 www.computer.org/csdl

Society Publications Seek Volunteers for 2011-2012 Editor in Chief Terms

The IEEE Computer Society seeks applicants for the position of editor in chief, serving a two-year term starting 1 January 2011. Prospective candidates are asked to provide (as PDF files) by **1 March** a complete curriculum vitae, a brief plan for the publication's future, and a letter of support from their institution or employer. For more information on the search process and to submit application materials for the following titles, please contact:

MAGAZINES

Computer; Jenny Stout, jstout@computer.org

IEEE Internet Computing; Jenny Stout, jstout@computer.org

IEEE Micro; Robin Baldwin, rbaldwin@computer.org

IEEE Software; Jenny Stout, jstout@computer.org

IEEE Security & Privacy; Jenny Stout, jstout@computer.org

TRANSACTIONS

For more information on the search process and to submit application materials for the following titles, please contact Kathy Santa Maria, ksantama@computer.org:

IEEE Transactions on Computers

IEEE Transactions on Visualization and Computer Graphics

QUALIFICATIONS AND REQUIREMENTS

Candidates for any Computer Society editor in chief position should possess a good understanding of industry, academic, and government aspects of the specific publication's field. In addition, candidates must demonstrate the managerial skills necessary to process manuscripts through the editorial cycle in a timely

fashion. An editor in chief must be able to attract respected experts to the editorial board. Major responsibilities include

- actively soliciting high-quality manuscripts from potential authors and, with support from publication staff, helping these authors get their manuscripts published;
- identifying and appointing editorial board members, with the concurrence of the Publications Board;
- selecting competent manuscript reviewers, with the help of editorial board members, and managing timely reviews of manuscripts;
- directing editorial board members to seek special-issue proposals and manuscripts in specific areas;
- providing a clear, broad focus through promotion of personal vision and guidance where appropriate; and
- resolving conflicts or problems as necessary.


Applicants should possess recognized expertise in the computer science and engineering community and must have clear employer support.

REAPPOINTMENTS

Other IEEE Computer Society publications have editors in chief who are currently standing for reappointment to a second two-year term. The IEEE Computer Society Publications Board invites comments upon the tenures of the individual editors.

Editors in chief standing for reappointment to terms in 2011-2012 are:

- Isabel Beichl, *Computing in Science & Engineering*
- Fei-Yue Wang, *IEEE Intelligent Systems*
- Beng Chin Ooi, *IEEE Transactions on Knowledge & Data Engineering*
- Ramin Zabih, *IEEE Transactions on Pattern Analysis & Machine Intelligence*
- Liang-Jie Zhang, *IEEE Transactions on Services Computing*

For magazines, send comments to Robin Baldwin, rbaldwin@computer.org. For transactions, send comments to Kathy Santa Maria, ksantama@computer.org. 

2 Free Sample Issues!



The magazine of computational tools and methods for 21st century science.

Send an e-mail to jbebee@aip.org to receive the two most recent issues of CISE.
(Please include your mailing address.)

<http://cise.aip.org> | www.computer.org/cise

AIP



IEEE



IEEE
computer
society

CALL AND CALENDAR

CALLS FOR ARTICLES FOR IEEE CS PUBLICATIONS

IEEE Internet Computing seeks articles for a November/December 2010 special issue on overcoming information overload issues.

Internet users today are inundated with information. They receive masses of e-mail, are interrupted by instant messages, and must remember to check social-networking sites, news sources, and company websites daily—or even many times each day. Web searches produce more hits than users can sift through.

Managing so much information is a very complex task. Syndication technology—such as RSS and Atom—and feed readers might provide some support, but issues related to the analysis, classification, evolution, and retrieval of information are open problems.

This special issue seeks original articles examining the state of the art, open problems, research results, tool evaluation, and future research directions in overcoming information overload.

Final submissions are due by **1 March**. Visit www.computer.org/portal/web/computingnow/iccfp6 to view the complete call for papers.

IEEE Security & Privacy seeks papers that explore the security and privacy opportunities and threats of cloud computing, including technical diversity and resiliency using clouds, shared infrastructure risks, applications of cloud computing for malware, and policy and compliance issues.

Articles are due by **5 March**. Visit www.computer.org/portal/web/computingnow/spcftp6 to view the complete call for papers.



IEEE Software seeks papers on the state of the art and current industrial practice in framing architectural concerns.

The guest editors of *Software's* November/December 2010 issue welcome case studies, success and failure stories in introducing viewpoints, frameworks, and models to organizations; mature and innovative approaches; and future trends.

Articles are due by **1 April**. Visit www.computer.org/portal/web/computingnow/swcfp6 to view the complete call for papers.

CALLS FOR PAPERS

EMS 2010, Int'l Conf. on Eng. Management and Service Sciences, 24-26 August, Wuhan, China; abstracts due **1 February**; www.scirp.org/conf/ems2010/CallForPapers.aspx

CALENDAR

FEBRUARY

28 Feb-3 Mar 2010: HPCA 2010, IEEE Int'l Symp. on High-Performance Computer Architecture, Bangalore, India; www.hpcaconf.org

MARCH

1-6 Mar: ICDE 2010, Int'l Conf. on

Data Engineering, Long Beach, California; www.icde2010.org

28-30 Mar: INFOS 2010, IEEE Int'l Conf. on Informatics and Systems, Giza, Egypt; <http://infos2010.fci.cu.edu.eg>

28-30 Mar: ISPASS 2010, IEEE Int'l Symp. on Performance Analysis of Systems and Software, White Plains, New York; www.ispass.org

29 Mar-2 Apr: PerCom 2010, Int'l Conf. on Pervasive Computing and Communications, Mannheim, Germany; www.percom.org

APRIL

12-16 Apr: DIGITEL 2010, IEEE Int'l Conf. on Digital Game and Intelligent Toy-Enhanced Learning (with WMUTE), Kaohsiung, Taiwan; <http://digitel2010.cl.ncu.edu.tw>

12-16 Apr: WMUTE 2010, IEEE Int'l Workshop on Wireless, Mobile, and Ubiquitous Technology in Educa-

SUBMISSION INSTRUCTIONS

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting.

Visit www.computer.org/conferences for instructions on how to submit conference or call listings as well as a more complete listing of upcoming computer-related conferences.

EVENTS IN 2010

February

9-14 HPCA 2010

March

1-6 ICDE 2010

29 Mar-2 Apr PerCom 2010

April

12-16 DIGITEL 2010

12-16 WMUTE 2010

19-23 IPDPS 2010

tion, Kaohsiung, Taiwan; <http://wmute2010.cl.ncu.edu.tw>

19-23 Apr: IPDPS 2010, IEEE Int'l Parallel & Distributed Processing Symp., Atlanta; www.ipdps.org

AUGUST

24-26 Aug: EMS 2010, Int'l Conf. on Eng. Management and Service Sciences, Wuhan, China; www.scirp.org/conf/ems2010

NOVEMBER

13-19 Nov: SC 2010, Int'l Conf. for High-Performance Computing, Networking, Storage, and Analysis, New Orleans; www.sc-conference.org

→ WMUTE 2010

The convergence of smart phones integrating high-quality media capture devices, trends in social networking, participatory media, and cyberinfrastructure provides a remarkable opportunity for making mobile social media integral to distributed learning environments.

Conference participants at the 6th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education will have the opportunity to interact and share recent research results in areas that include social media, mobile devices, user-generated content, and human-computer interaction. WMUTE 2010 shares a venue with the IEEE International Conference on Digital Game and Intelligent Toy-Enhanced Learning.

WMUTE is sponsored by the IEEE Computer Society Technical Committee on Learning Technology. The conference takes place 12-16 April in Kaohsiung, Taiwan. Visit <http://wmute2010.cl.ncu.edu.tw> for complete conference details.

Give Your Career a Boost

In today's environment, strengthening your resume is more important than ever. Whether you are an entry-level or mid-career software practitioner, we have the answer:

Distinguish yourself with one of the IEEE Computer Society's software development credentials.



"Having the CSDP helped me make the case for strengthening our software quality process, which drastically reduced our production support costs by 40%."

Phanindra Mankale, CSDP
F500 Manufacturing Company

Stand out from the others with the CSDA/CSDP



For more information, and to see how these credentials have helped other practitioners, go to: www.computer.org/getcertified

CAREER OPPORTUNITIES

PURDUE UNIVERSITY, School of ECE, Computer Engineering Faculty Position in Human-Centered Computing.

The School of Electrical and Computer Engineering at Purdue University invites applications for a faculty position at any level in human-centered computing, including but not limited to visualization, visual analytics, human computer interaction (HCI), and graphics. The Computer Engineering Area of the school (<http://engineering.purdue.edu/ECE/Research/Areas/CompEng>) has nineteen faculty members who have active research programs in areas including AI, architecture, compilers, computer vision, distributed systems, embedded systems, graphics, haptics, HCI, machine learning, multimedia systems, networking, networking applications, NLP, OS, robotics, software engineering, and visualization. Eligible candidates are required to have a PhD in computer science/engineering or a related field and a significant demonstrated research record commensurate with the level of the position applied for. Applications should consist of a cover letter, a CV, a research statement, names and contact information for at least three references, and URLs for three to five online papers. Applications should be submitted to [https://engineering.purdue.edu/](https://engineering.purdue.edu/Engr/AboutUs/Employment/Applications)

[Engr/AboutUs/Employment/Applications](https://engineering.purdue.edu/Engr/AboutUs/Employment/Applications). Review of applications will begin on 1 December 2009. Inquiries may be sent to ece-hcc-search@ecn.purdue.edu. Applications will be considered as they are received, but for full consideration should arrive by 1 January 2010. Purdue University is an equal opportunity, equal access, affirmative action employer fully committed to achieving a diverse workforce.

UNIVERSITY OF WASHINGTON, Computer Science & Engineering and Electrical Engineering, Tenure-Track and Research Faculty, Ref. AA2440.

The University of Washington's Department of Computer Science & Engineering and Department of Electrical Engineering have jointly formed a new UW Experimental Computer Engineering Lab (ExCEL). In support of this effort, the College of Engineering has committed to hiring several new faculty over the forthcoming years. All positions will be dual appointments in both departments (with precise percentages as appropriate for the candidate). This year, we have one open position, and encourage exceptional candidates in computer engineering, at tenure-track Assistant Professor, Associate Profes-

sor, or Professor, or Research Assistant Professor, Research Associate Professor, or Research Professor to apply. A moderate teaching and service load allows time for quality research and close involvement with students. The CSE and EE departments are co-located on campus, enabling cross department collaborations and initiatives. The Seattle area is particularly attractive given the presence of significant industrial research laboratories, a vibrant technology-driven entrepreneurial community, and spectacular natural beauty. Information about ExCEL can be found at www.excel.washington.edu. We welcome applications in all computer engineering areas including but not exclusively: atomic-scale devices and nanotechnology, implantable and biologically-interfaced devices, synthetic molecular engineering, VLSI systems and CAD, embedded systems, sensor systems, parallel computing, network systems, and technology for the developing world. We expect candidates to have a strong commitment both to research and teaching. ExCEL is seeking individuals at all career levels, with appointments commensurate with the candidate's qualifications and experience. Applicants for both tenure-track and research positions must have earned a PhD by the date of appointment. Please apply online at www.excel.washington.edu. Apply with a letter of application, a complete curriculum vitae, statement of research and teaching interests, and the names of at least four references. Applications received by 1 February, 2010 will be given priority consideration. Open positions are contingent on funding. The University of Washington was awarded an Alfred P. Sloan Award for Faculty Career Flexibility in 2006. In addition, the University of Washington is a recipient of a National Science Foundation ADVANCE Institutional Transformation Award to increase the participation of woman in academic science and engineering careers. We are building a culturally diverse faculty and encourage applications from women and minority candidates. The University of Washington is an affirmative action, equal opportunity employer.

Positions at the Institute for Defense Analyses Center for Computing Sciences

The Institute for Defense Analyses Center for Computing Sciences (IDA/CCS) is looking for outstanding researchers to address difficult computing problems vital to the nation's security. IDA/CCS is an independent, applied research center sponsored by the National Security Agency (NSA). Emphasis areas for IDA/CCS technical staff include high-performance computing, cryptography, and network security. Members of the technical staff come from a diverse variety of backgrounds, including computer science, computer architecture, computer/electrical engineering, information processing, and the mathematical sciences; most have Ph.D.s. Special attention is paid to the design, prototyping, evaluation, and effective use of new computational algorithms, tools, paradigms, and hardware directly relevant to the NSA mission. Stable funding provides for a vibrant research environment, and an atmosphere of intellectual inquiry free of administrative burdens.

The center is equipped with a very large variety of hardware and software. The latest developments in high-end computing are heavily used and projects routinely challenge the capability of the most advanced algorithms and architectures. IDA/CCS research staff members have always been at the forefront of computing, as evidenced by lasting, visible contributions to areas as varied as multi-threaded architectures (e.g., Horizon), novel computing systems (e.g., FPGA-based Splash and Splash-2, Processing-In-Memory chips), design and implementation of operating systems (e.g., the Linux kernel), and programming language design and implementation for high-performance computing systems (e.g., Universal Parallel C and Cinquecento).

IDA/CCS research staff work on complex topics often engaging multidisciplinary teams; candidates should demonstrate depth in a particular field as well as a broad understanding of computational issues and technology. Because the problems of interest are continually evolving, IDA/CCS recruitment focuses on self-motivation, strength of background, and talent, rather than specific expertise.

Located in a modern research park in the Maryland suburbs of Washington, DC, IDA/CCS offers a competitive salary, an excellent benefits package, and a superior professional working environment. U.S. citizenship and a Department of Defense TSSI clearance (with polygraph) are required. IDA/CCS will sponsor this clearance for those selected. The Institute for Defense Analyses is proud to be an equal opportunity employer.

Please send responses or inquiries to:

Dawn Porter
Administrative Manager
IDA Center for Computing Sciences
17100 Science Drive
Bowie, MD 20715-4300
dawn@super.org

SYSTEM ANALYST F/T in Poughkeepsie, NY. Must have Bach's deg or equiv in Comp Engg/Sci & 2yrs exp in Business Analysis & software SDLC testing. Responsible for leading test team, manage test lab, write Test Plan/ Test Case/ Defect, test video applications using like Functional, Regression, Automaton, Performance, Internationalization, Database, Smoke, Beta, Sanity, Interoperability. Exp in Premiere Pro, Test Director,

SUBMISSION DETAILS: Rates are \$445.00 per column inch (\$500 minimum). Eight lines per column inch and average five typeset words per line. Free online listing on careers.computer.org with print ad. Send copy at least one month prior to publication date to: Marian Anderson, Classified Advertising, Computer Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314; (714) 821-8380; fax (714) 821-4010. Email: manderson@computer.org.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, Computer may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...", "...1-4 years maximum experience...", "...up to 5 years experience," or "...10 years maximum experience." Computer reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. Computer assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. Computer encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

جامعة كارنيغي ميلور في قطر
Carnegie Mellon Qatar
School of Computer Science



Computer Science Faculty Positions

Carnegie Mellon University in Qatar invites applications for teaching-track positions at all levels in the field of Computer Science. These career-oriented renewable appointments involve teaching international undergraduate students, and maintaining a significant research program. Candidates must have a Ph.D. in Computer Science or related field, substantial exposure to Western-style education, outstanding teaching record and excellent research accomplishments or potential.

Specifically, we are seeking candidates with expertise in databases, data mining, web technology and human-computer interaction, or with substantial experience teaching introductory programming courses. Truly exceptional candidates in other areas also will be considered.


The position offers a competitive salary, foreign service premium, research seed grant, excellent international health coverage and allowances for housing, transportation, dependent schooling and travel.

Carnegie Mellon is internationally recognized as a leader in research and higher education. In 2004, the university established itself in Education City, a state-of-the-art campus that is home to six top universities. Collaboration opportunities with internationally-known researchers and world-class businesses are abundant.

For further information or to apply, visit
<http://www.qatar.cmu.edu/cs/positions/>



INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



Join **INRIA**, a major player
in the field
of **COMPUTER
SCIENCES**

In 2010
INRIA offers
various searcher
positions.
To apply, please
view our open job
listing at:
www.inria.fr/work



CAREER OPPORTUNITIES

WinRunner, SilkTest, LoadRunner, ClearQuest, Java, C/C++, SQL, Unix/Windows OS. Send resume: Apollo Consulting Services Corp., Recruiting (NN), 14 Catharine St, Poughkeepsie, NY 12601.

IT MANAGER, NYC (F/T). Participate in architecture design, create development specifications. Prepare project plans including development of linguistic, semantic text analysis and text classification modules. Lead team, train, review work. MS in Comp Eng/Info Sys or Comp Science, 2 yrs exp as Project Mgr. Instant Information Inc, by email: Michael.Akselrod@infongen.com.

TRANSPORT MGMT SYS UI Devel., Enroute Traffic Sys., Golden, CO. Req. BS or for. equiv in Comp. Sci & Engg + 5 yrs prgmng exp. Mail CV to D. Sanford (ref#091247) PO Box 260130, Lakewood, CO 80226

QA ANALYST - dsgn, dvlp, test & implmt applic s/w utilizing knowl of & exp w/Java, J2EE, Jmeter, QALoad, WebSphere, WebLogic, Winrunner, Quality Center & Test Director, exp in Unix, Win, 00/NT (TCP/IP, LAN, WAN). Define & dvlp QA Testing practices & procedures; Req MS in Comp Sci, Eng or rel. Mail resumes

to SmartWorks LLC, 55 Carter Dr, Ste 107, Edison, NJ 08817

TEMPLE UNIVERSITY, Department of Computer and Information Sciences, Tenure-Track Faculty. Applications are invited for a tenure-track, open rank, faculty position in the Department of Computer and Information Sciences at Temple University. Areas of interest include, but are not limited to Computer Systems, Wired and Wireless Networks, and Trustworthy and Reliable Computing. For senior rank candidates, applications should include curriculum vitae, a statement of recent achievements, and research, and teaching goals, up to three representative publications, and names and addresses of at least three references. Junior candidates should have three reference letters sent directly. Please submit applications online at <http://academic-jobsonline.org>. For further information check <http://www.cis.temple.edu>. Review of candidates will begin on February 1, 2010 and will continue until the position is filled. Temple University is an equal opportunity, equal access, and affirmative action employer.

SR. DATABASE ADMIN, F/T, NYC. Deploy, support oracle DB. Develop PL/

SQL, Java, Perl Scripts. MS in Comp Science/Applied Math/Info Tech + 2 yrs/exp. By E-mail to Visual Trading Systems LLC at IGINDEL@VTSYSTEMS.COM

UNIVERSITY OF WASHINGTON, Computer Science & Engineering, Tenure-Track, Research, and Teaching Faculty, Ref. #AA2439. The University of Washington's Department of Computer Science & Engineering has one or more open positions in a wide variety of technical areas in both Computer Science and Computer Engineering, and at all professional levels. A moderate teaching load allows time for quality research and close involvement with students. Our space in the Paul G. Allen Center for Computer Science & Engineering provides opportunities for new projects and initiatives. The Seattle area is particularly attractive given the presence of significant industrial research laboratories as well as a vibrant technology-driven entrepreneurial community that further enhances the intellectual atmosphere. Information about the department can be found on the web at <http://www.cs.washington.edu>. We welcome applicants in all research areas in Computer Science and Computer Engineering including both core and inter-disciplinary areas. Areas of interest include (but are not limited to) security, computer engineering, and systems. We expect candidates to have a strong commitment both to research and to teaching. The department is primarily seeking individuals at the tenure-track Assistant Professor rank; however, under unusual circumstances and commensurate with the qualifications of the individual, appointments may be made at the rank of Associate Professor or Professor. We may also be seeking non-tenured research faculty at Assistant, Associate and Professor levels, postdoctoral researchers (Research Associates) and part-time and full-time annual lecturers and Sr. Lecturers. Applicants for both tenure-track and research positions must have earned a doctorate by the date of appointment; those applying for lecturer positions must have earned at least a Master's degree. Research Associates, Lecturers and Sr. Lecturers will be hired on an annual or multi-annual appointment. All University of Washington faculty engage in teaching, research and service. Please apply online at <http://norfolk.cs.washington.edu/apply> with a letter of application, a complete curriculum vitae, statement of research and teaching interests, and the names of four references. Applications received by February 1, 2010 will be given priority consideration. Open positions are contingent on funding. The Uni-

ROCHESTER INSTITUTE OF TECHNOLOGY

COMPUTING AND INFORMATION SCIENCES
DEPARTMENT CHAIR OF SOFTWARE ENGINEERING
BEGINNING JULY 1, 2010

Rochester Institute of Technology's B. Thomas Golisano College of Computing and Information Sciences (GCCIS) invites applications and nominations for the position of department chair in the Software Engineering department (IRC#35729).

Successful candidates must possess credentials that merit appointment at the rank of Full Professor with tenure at RIT and have:

- qualifications and skills for playing a strong leadership role in the department as well as in the college
- a tangible research and external funding track record
- experience in curriculum development at the graduate and undergraduate levels
- experience in program administration

A Ph.D. in a related computing discipline is required; salary is commensurate with experience. Specialization in any area of software engineering will be considered, with preference for software quality assurance, secure software systems, engineering of web-based software systems, and embedded and real-time systems. We are seeking individuals who are committed to contributing to RIT's core values, honor code, and statement of diversity.

The Golisano College of Computing and Information Sciences is home to the Computer Science, Software Engineering, Information Sciences and Technologies, Interactive Games & Media, and Networking, Security, and Systems Administration departments, as well as the Ph.D. program in Computer and Information Sciences. The college has 105 faculty and over 2400 undergraduate and 600 graduate students.

Candidates should visit <https://mycareer.rit.edu> and refer to IRC#35729 for specific information about the position and the application process. Refer to www.rit.edu for information about RIT and the B. Thomas Golisano College of Computing and Information Sciences.

R·I·T

-Selected as one of the top colleges to work for by
The Chronicle of Higher Education (2008, 2009)

versity of Washington was awarded an Alfred P. Sloan Award for Faculty Career Flexibility in 2006. In addition, the University of Washington is a recipient of a National Science Foundation ADVANCE Institutional Transformation Award to increase the participation of women in academic science and engineering careers. We are building a culturally diverse faculty and encourage applications from women and minority candidates. The University of Washington is an affirmative action, equal opportunity employer.

ELECTRONIC DATA SYSTEMS, LLC (EDS, HP Enterprise Services), is accepting resumes for the following positions: **SERVICES INFORMATION DEVELOPER IN RANCHO CORDOVA, CA.** (Ref. # EDSRCSA1). Conceptualize, design, construct, test, & implement portions of business & tech IT solutions through application of appropriate SW devlpmt life cycle methodology. Requires Bachelor's or foreign degree equivalent in Electronic Engg, Comp Sci, Maths, Info Sys, Comp Engg, Electrical Engg, or related field + 5 yrs post-baccalaureate, progressive exp in job offered, or as programmer analyst, technical lead, SW engineer, systems architect, or related

occupation. Object Oriented Analysis; UML design patterns; JSP; Struts Framework; JavaScript; and JDBC. Please mail resumes with reference # to: Ref. # EDSRCSA1, Jim York, Applications Manager, EDS, HP Enterprise Services, 10888 White Rock Road, Rancho Cordova, CA 95670. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

SERVICES INFORMATION DEVELOPER IN RANCHO CORDOVA, CA. (Ref. # EDSRCSA1). Electronic Data Systems, LLC (EDS, HP Enterprise Services), is accepting resumes for Services Information Developer in Rancho Cordova, CA. (Ref. #EDSRCSA1). Conceptualize, design, construct, test, & implement portions of business & tech. IT solutions through application of appropriate SW devlpmt life cycle methodology. Requires Master's or foreign degree equivalent in Comp Sci, Maths, Info Sys, Comp Engg, Electrical Engg, or related + 3 yrs exp in job offered, or as programmer analyst, analyst programmer, SW engineer, consultant, or related. Object Oriented Analysis; UML design patterns; Java Server Pages; Struts Framework; JavaScript; & JDBC. Please mail resumes with reference # to: Ref. # EDSRCSA1, Jim York, Applications Manager, EDS,

HP Enterprise Services, 10888 White Rock Road, Rancho Cordova, CA 95670. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

PROGRAMMER ANALYST: dsgn, dvlp, test & implmt applic s/w utilizing Oracle 8i/9i/10g, SQL, PL/SQL, VB.Net, ASP.Net, C#, Java, XML, Toad, Cognos, Linux, Win NT/2000/2003; Req MS Comp Sci, Eng, or equiv. Mail resumes to Strategic Resources International, 777 Washington Rd, Ste 2, Parlin, NJ 08859.

HEWLETT-PACKARD COMPANY is accepting resumes for Software Designer in San Francisco, CA (Ref. #SFSWD11). Design, develop, maintain, test, & perform quality & performance assurance of system SW products. Please mail resumes with reference # to Ref. # SFSWD11, Hewlett-Packard Company, 19483 Pruneridge Avenue, MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

CITCO TECHNOLOGY MANAGEMENT INC. has an opening in Ft. Lauderdale, FL for Systems Administrator to Install,



FACULTY POSITIONS IN COMPUTER SCIENCE / COMPUTING SYSTEMS / INFORMATION SYSTEMS WITH NANYANG TECHNOLOGICAL UNIVERSITY

Nanyang Technological University (NTU), Singapore is ranked globally as one of the best universities in the World. Under the University's College of Engineering, the **School of Computer Engineering (SCE)-NTU**, established in 1988, offers undergraduate training leading to a BEng (Hons) in Computer Engineering and Computer Science, as well as graduate training leading to MSc, MEng and PhD. A research intensive institution with a strong R&D infrastructure and networked alliance with industry and academia, the School offers its academic staff the opportunity to pioneer cutting-edge research in a wide spectrum of technological areas.

SCE comprises four divisions; Division of Computer Communications (CCM), Division of Computer Science (CSC), Division of Computing Systems (CPS) and Division of Information Systems (IS), and is home to over 2,052 students as well as 100 academic staff from across the globe.

In light of the rapid growth of the Information Technology arena, high-calibre PhD holders with a proven track record in research, and teaching at a university level are invited to apply for suitable appointments as **Associate Professor (A/P)** or **Assistant Professor (Ast/P)** in the following areas:

- **High Performance Computing or Distributed Systems (A/P or Ast/P in CSC)**
 - **Artificial Intelligence or Computational Intelligence (A/P or Ast/P in CSC)**
- **Audio, Speech & Signal Processing (A/P or Ast/P in CPS)**
 - **Bioinformatics (Ast/P in IS)**
 - **Machine Learning & Intelligent Agents (Ast/P in IS)**
 - **Agents, Services Computing & Text Mining (A/P or Ast/P in IS)**

Candidates for appointment at an **Associate Professor** level must possess an outstanding track record of research through publication in top ranking journals, obtaining grants and academic leadership, as well as a willingness and demonstrated ability to teach at the undergraduate and graduate levels. Candidates for appointment at the **Assistant Professor** level must demonstrate strong research potential and a willingness and ability to teach at the undergraduate and graduate levels. Successful candidates are expected to carry out research in one of the research centres hosted by SCE, as well as teach MSc, MEng and BEng Computer Engineering/Computer Science programmes offered by the School.

Based on the qualifications and experience, successful candidates can look forward to an excellent remuneration package, and start-up grants to pursue research interests in the broad field of Computer Engineering/Computer Science.

Further information about the school can be obtained at <http://www.ntu.edu.sg/sce>. Informal enquiries and submission of application forms can be made to SCEHR@ntu.edu.sg. Guidelines for application submission and application forms can be obtained from <http://www.ntu.edu.sg/ohr/Career/SubmitApplications/Pages/default.aspx>.

Closing Date: 15 March 2010

www.ntu.edu.sg

CAREER OPPORTUNITIES

configure & customize CA Service Desk r11, Knowledge Tools & Dashboard in a Windows 2003, SQL 2000 & Active Directory environment. Send resumes to employment@citco.com. Please reference Job code CTM31 in email subject line.

THE UNIVERSITY OF ALABAMA, Department of Computer Science. The University of Alabama, Department of Computer Science, invites applications for a new assistant professor position to begin August 16, 2010. Candidates must have an earned Ph.D. in computer science or a related field, with solid evidence of superior research and scholarship accomplishments that are appropriate for the desired level of appointment, as well as quality teaching abilities. Applicants from all areas of computer science will be considered. Those who specialize in software engineering, database systems, operating systems, or networking are particularly encouraged to apply. High priority areas for us include model-driven engineering (e.g., domain-specific modeling) with specific application to software product lines and mobile software development. The University of Alabama, located in Tuscaloosa, is considered the Capstone of higher education and is also the largest institution in the State. The Department of Computer Science, housed in the College of Engineering, currently has twenty-three faculty members (16 tenured/tenure-track), roughly 200 undergraduates in an ABET accredited B.S. degree program, and approximately 60 M.S. and Ph.D. stu-

dents. Outstanding applicants should send curriculum vitae and the names and addresses of at least three references to: Faculty Search Committee, Department of Computer Science, Box 870290, The University of Alabama, Tuscaloosa, AL 35487-0290. E-mail: faculty.search@cs.ua.edu. E-mail submissions are also encouraged. For additional information, please visit <http://cs.ua.edu> or contact the Search Committee at faculty.search@cs.ua.edu. Review of applications will begin January 29, 2010 and will continue until the position is filled. The University of Alabama is an equal opportunity/affirmative action employer. Women and minorities are particularly encouraged to apply.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Roseville, CA. *Global Deal Consultant.—*Resp. for consulting teams re all Custmr Ops related topics in presales and bid-phase thru implmnt & transitt. Reqs: Adv. Knwldg of operatnl processes relating to the Svc Engmt bus; Adv. Fin. knwldg; strong knwldg of project acctg processes & procedures relating to Global Deal Bus; Sys & tool knwldg incl. SAP, billing & invoicing systems; knwldg of global invoicing, structures & understgd of global tax implicatns; Proj Mgmt Knwldg; Contract structure knwldg & understgd. Also reqs: Bach deg or foreign equiv. in Acctg or rel. & 2 yrs exp in job offered or rel. Send resume and refer to Job#ROSJMA2. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014.

No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Cupertino, CA. Quality Assurance Engineer.—Reqs. experience in: testing web-based enterprise applctns; products built using Java, a Web interface & a database back-end; automatr of complex applctns; Agile methodology exp.; Web Svcs testing using Java and .Net clients. Also reqs: Bach degree or foreign equiv in Engrng or rel field of study. & 2 yrs exp in job offered or rel. Send resume and refer to Job#CUPAPR2. Please send resumes with job number to Hewlett-Packard Company, 19483 Pruneridge Ave., MS 4206, Cupertino, CA 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

PROGRAMMER ANALYST. Design & dvlp transparent, scalable & portable distributed/parallel structured s/w sys. Dvlp & Implement C/C++/Linux engines capable of interacting & controlling several dozen devices. Analyze data in real time from several devices in a multi-threaded C/C++ sys. Configure POSTGRES/MYSQL database & write PYTHON scripts. Dvlp factory automation broker servers using ASYST libraries (SECS/GEM interface). Dvlp JAVA/.NET GUI apps to interact w/ Users. Req: Masters in Comp Sci, Comp Engrng or Elec Engrng. 40 hr/wk. Job/Interview Site: Brea, CA. EMAIL resume to: MTSI Inc. at Jobs110903@mtsiinc.com.

ADVERTISER INFORMATION | JANUARY 2010 • COMPUTER

Advertiser	Page	Advertising Sales Representatives	Midwest/Southwest	Product:
Carnegie Mellon University in Qatar	73		Darcy Giovingo	US East
IDA Center for Computing Sciences	72		Phone: +1 847 498 4520	Dawn Becker
INRIA	73		Fax: +1 847 498 5911	Phone: +1 732 772 0160
Nanyang Technological University	75		Email: dg.ieeemedia@ieee.org	Fax: +1 732 772 0164
Rochester Institute of Technology	74			Email: db.ieeemedia@ieee.org
Advertising Personnel			Northwest/Southern CA	US Central
Marion Delaney			Tim Matteson	Darcy Giovingo
IEEE Media, Advertising Dir.			Phone: +1 310 836 4064	Phone: +1 847 498 4520
Phone: +1 415 863 4717			Fax: +1 310 836 4067	Fax: +1 847 498 5911
Email: md.ieeemedia@ieee.org			Email: tm.ieeemedia@ieee.org	Email: dg.ieeemedia@ieee.org
Marian Anderson			Japan	US West
Sr. Advertising Coordinator			Tim Matteson	Lynne Stickrod
Phone: +1 714 821 8380			Phone: +1 310 836 4064	Phone: +1 415 931 9782
Fax: +1 714 821 4010			Fax: +1 310 836 4067	Fax: +1 415 931 9782
Email: manderson@computer.org			Email: tm.ieeemedia@ieee.org	Email: ls.ieeemedia@ieee.org
Sandy Brown			Europe	Europe
Sr. Business Development Mgr.			Hilary Turnbull	Sven Anacker
Phone: +1 714 821 8380			Phone: +44 1875 825700	Phone: +49 202 27169 11
Fax: +1 714 821 4010			Fax: +44 1875 825701	Fax: +49 202 27169 20
Email: sb.ieeemedia@ieee.org			Email: impress@impressmedia.com	Email: sanacker@intermediapartners.de

WEB TECHNOLOGIES

Web 3.0: The Dawn of Semantic Search

➔ James Hendler, Rensselaer Polytechnic Institute



Emerging Web 3.0 applications use semantic technologies to augment the underlying Web system’s functionalities.

In the past two January editions of this *Computer* column, I’ve had the pleasure of writing about the status of the Semantic Web, and particularly of its applied use in Web applications, increasingly coming to be known as Web 3.0. I’m happy to say development and deployment continue apace, and that for those of us who know where to look, we see a lot of progress. Of course, if we were really being successful, you wouldn’t have to know what rocks to look under—it would be everywhere. Or would it?

AN INFRASTRUCTURE TECHNOLOGY

One of the difficulties in explaining Web 3.0 is that, unlike the original Web browser or later Web 2.0 systems, Semantic Web technology tends to be an infrastructure technology. While Web companies are working to produce new and scalable tools, academic researchers are pushing the size and speed of Semantic Web back-end operations.

Corporate development

Web developers are learning that they can build an application from a combination of traditional databases with RDF (Resource Description Framework) triple stores, the databases of the Semantic Web.

Traditional databases provide scaling for the back-end dynamics that are well-developed and clear; semantic databases provide new functionality that requires Web linking, flexible representation, and external access APIs. Relational databases provide the computational beef, the triple stores the secret Web 3.0 sauce.

In fact, you’ve probably visited a website sometime in the past few weeks that was built this way, but as hardly anyone is using any kinds of “Web 3.0 inside” labels, it’s not surprising that you didn’t know.

The first generation of enterprise Web 3.0 systems uses behind-the-scenes “structural” semantics to extend their current capabilities—for example, taxonomies with simple properties that can be used to relate terms to each other or to integrate terminologies from multiple sites. This sort of “controlled vocabulary” has been around for a long time, but emerging technologies allow it to be more easily integrated with Web development (J. Hendler, “Web 3.0 Emerging,” *Computer*, Jan. 2009, pp. 111-113).

In addition, new standards make it possible to find consulting and tool-development companies that can help provide the scalable back ends needed to make the systems succeed. One company I work with has a cus-

tommer who has hired them to develop a “trillion triple” store that can keep up with its complex application’s real-time needs.

Academic research

While details about corporate use of the Semantic Web and the architecture to support it are still under wraps, the academic community is also looking more seriously at scalable reasoning and large-scale back-end applications.

Researchers are exploring the scaling of both triple-store capabilities and inference algorithms for Semantic Web languages. The best paper at the 2007 VLDB conference was on using a DBMS for Semantic Web data management (D.J. Abadi et al., “Scalable Semantic Web Data Management Using Vertical Partitioning,” *Proc. 33rd Conf. Very Large Data Bases*, VLDB Endowment, 2007, pp. 411-422), and that work has led to a number of new techniques for accessing and optimizing Semantic Web data.

At last year’s International Semantic Web Conference (*The Semantic Web—ISWC 2009*, LNCS 5823, Springer, 2009), researchers presented findings on using parallel architectures for performing reasoning over semantic Web data at scale, including the design of both a

WEB TECHNOLOGIES

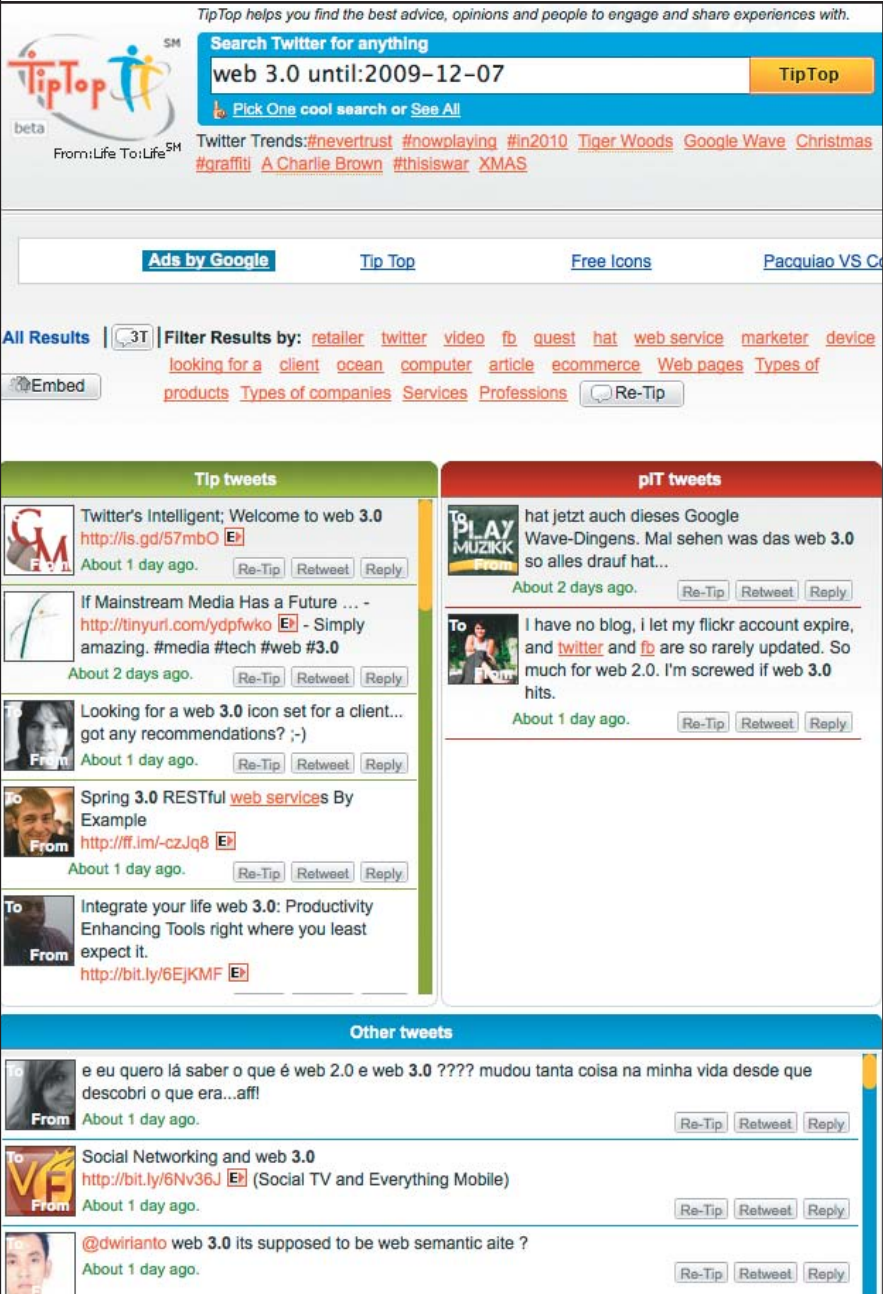


Figure 1. Real-time search engine TipTop combines language technologies with search to classify Twitter results into positive responses (green), negative responses (red), and other opinions (blue).

MapReduce mechanism (J. Urbani et al., “Scalable Distributed Reasoning Using MapReduce,” pp. 634-649) and a cluster-based technique (J. Weaver and J. Hendler, “Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples,” pp. 682-697) for computing the inference-based closures for more than a billion triples. Other work looked

at more efficient reasoning in the space of OWL reasoning (J. Du et al., “A Decomposition-Based Approach to Optimizing Conjunctive Query Answering in OWL DL,” pp. 146-162) at larger scales.

SEMANTIC SEARCH

Web companies haven't been waiting around for research results, and

they're starting to deploy specialized algorithms to meet their own needs. In fact, some new technologies are beginning to emerge from the Semantic Web infrastructure, and in 2010 we'll see more of these companies providing greater functionality to their users.

The most important area where we'll see these technologies on the Web is in the growing area of semantic search engines. These include systems that try to augment general searches as well as systems that are trying to literally change the search experience.

While the internal details of most of these systems are still proprietary, in general they appear to combine a pragmatic approach to natural-language processing with a lightweight semantics that lets them better collect and process information about specific areas. A complete survey is beyond the mandate of this column, but a few applications will suffice to highlight some of the differences between these and traditional systems.

More informative results

One semantic search capability is the attempt to provide more informative results than are typically returned by a regular search engine. Rather than simply identifying a useful page, these systems try to pull the information from those pages that might be what a user is looking for, and to make this immediately apparent.

For example, a search for “James Hendler” with Sensebot (www.sensebot.net), one of the newer search engines, will return a set of results such as

[James Hendler](#) (born April 2, 1957) is an artificial intelligence researcher at Rensselaer Polytechnic Institute, USA, and one of the originators of the [Semantic Web](#).

[SOURCE: [James Hendler facts - Freebase](#) (www.freebase.com/view/en/james_hendler)]

Sensebot uses language technologies to identify specific assertions about the object being searched for—my name in this case—and to provide a source for those assertions: It found this sentence in the Freebase open-database system.

Further search suggestions

A second capability offered by semantic search is to try to help a user identify further searches that may be more useful, and which can identify related searches to help users hone in on what they're looking for.

Probably the best known example of this capability is offered by Microsoft's Bing search engine. For example, a Bing search for "IEEE Computer" returns, along with regular search results, results for a list of related queries such as "IEEE PCS" or "IEEE Computer Security Conference" that might lead a user to more detailed information.

These expansions vary in quality based on how much data Bing has on the particular thing being searched for and sometimes can be quite impressive. For example, a search for the actress "Gates McFadden" returns a sidebar of related results for Brent Spiner, Lavar Burton, Will Wheaton, and some of her other *Star Trek: The Next Generation* costars.

"Affective" Web content

Another illustration of things to come in semantic search is the combination of language technologies with search to discover "affective" aspects of Web content, especially in the blogosphere or in Twitter feeds.

An example of such technology is the real-time search engine TipTop (<http://feeltiptop.com>). Figure 1 shows the results of a TipTop search for "web 3.0," which sorts several recent Twitter messages that mention the topic into positive responses (green), negative responses (red), and other opinions (blue).

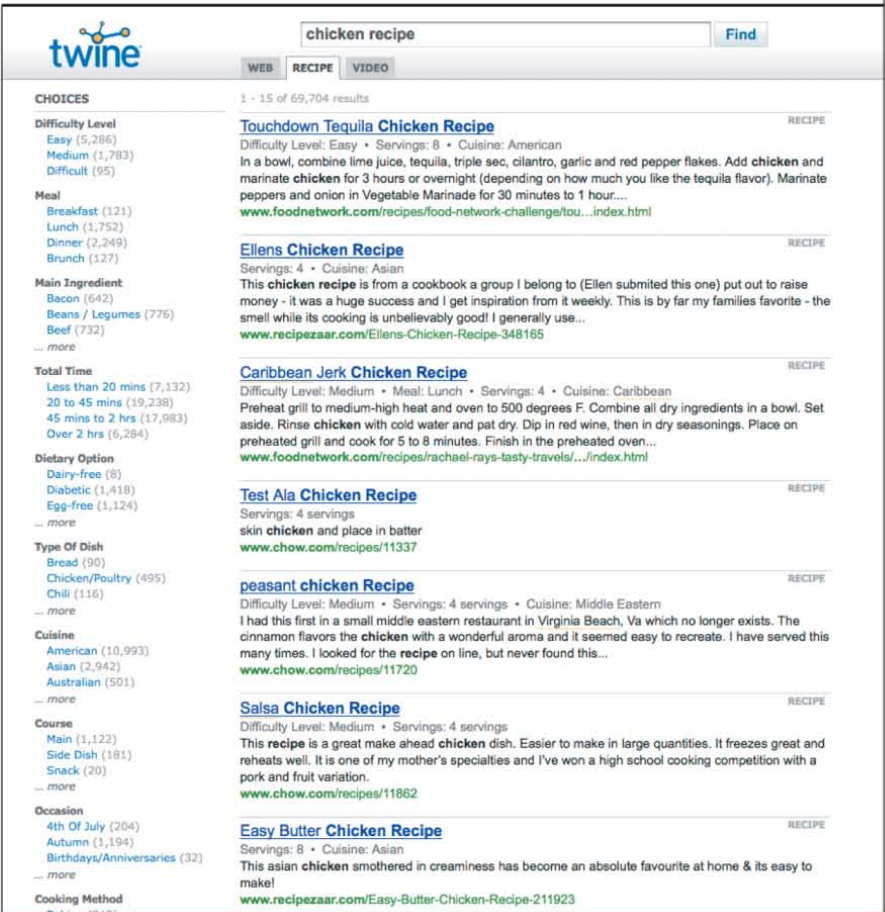


Figure 2. Search engine T2 draws on domain knowledge to find and categorize search results.

Domain knowledge

An important use of semantics in search is to draw on domain knowledge in areas where searches are difficult.

At ISWC 09, Nova Spivack, CEO and founder of Radar Networks (developers of Twine), gave some examples of this capability in his company's forthcoming T2 semantic search engine. Figure 2 shows the results of a search for "chicken recipe." The engine uses semantic technologies not only to find recipes from sites but also to filter them by several categories including cooking time, dietary options, and cuisine.

Semantic search techniques that use domain knowledge clearly would change the search experience if widely deployed. However, it will take time and a combination of human and machine effort to cover the enor-

mous diversity of Web domains. T2's solution to this problem is to provide the means for people to create these mappings using social, wiki-like mechanisms, thus extending the search engine's reach.

Matching people and needs

Beyond simple keyword matching, another use of semantics and language technologies is to find matches between people and their needs.

A good example of this is Applied Informatics' TrialX (<http://trialx.com>), which won the 2009 Semantic Web Challenge (<http://challenge.semanticweb.org>). This application uses advanced medical ontologies to combine electronic health records with user-generated information to match people with potentially helpful clinical trials. Thus far, TrialX has successfully matched more

WEB TECHNOLOGIES

than 3,000 participants to appropriate trials, helping both users and researchers in the pursuit of new medical treatments.

Another example of semantic matching is offered by Bintro (www.bintro.com), a classifieds website that matches users to whatever they are looking for—jobs, volunteer organizations to join, business partners, and so on—without them having to fill out a form or hope that someone notices them in a sea of static listings, or doing multiple keyword searches. Bintro uses a combination of semantic techniques to determine the meaning of a user’s descriptions and then, through an asynchronous match process, finds other users with similar descriptions. Thus, a person looking

for a position as a “childcare provider in the New York City area” could be matched with someone advertising for a “nanny in Manhattan.”

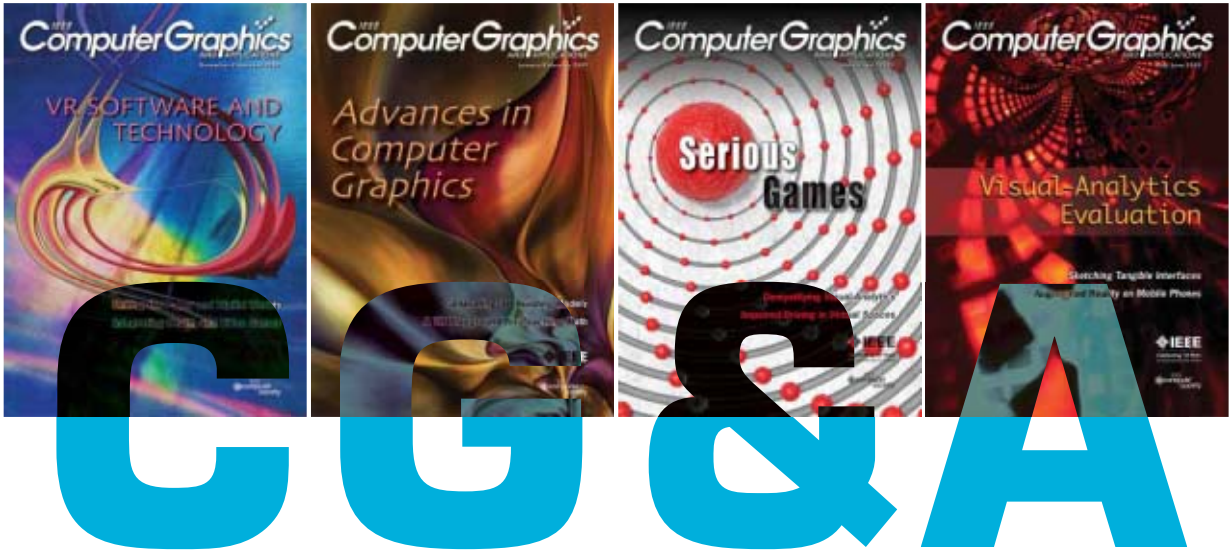
Like most semantic search applications, Bintro takes advantage of Semantic Web standards to build independent ontologies and thereby rapidly move into new domain areas.

Emerging Web 3.0 applications are enhancing search engines in interesting new ways. Architecturally, they all have one thing in common—they use semantics to augment the underlying Web system’s functionalities. When the semantics aren’t applicable, or where they fail to add value, the underlying application looks like a traditional

Web or Web 2.0 site. But where the semantics can be useful, the new functionality adds some exciting oomph. And on the Web, a little oomph can lead to a lot of money. **■**

James Hendler is the Tetherless World Chair of Information Technology and Web Science at Rensselaer Polytechnic Institute, as well as an advisory or board member for numerous Semantic Web companies including Radar Networks and Bintro. Contact him at hendler@cs.rpi.edu.

Editor: Simon S.Y. Shim, Dept. of Computer Engineering, San Jose State Univ., San Jose, CA; simon.shim@sjsu.edu



IEEE Computer Graphics and Applications bridges the theory and practice of computer graphics. From specific algorithms to full system implementations, CG&A offers a unique combination of peer-reviewed feature articles and informal departments. CG&A is indispensable reading for people working at the leading edge of computer graphics technology and its applications in everything from business to the arts.

Visit us at www.computer.org/cga

EMBEDDED COMPUTING

Mobile Supercomputers for the Next-Generation Cell Phone



➔ Mark Woh, Scott Mahlke, and Trevor Mudge, *University of Michigan*
➔ Chaitali Chakrabarti, *Arizona State University*

AnySP demonstrates that power efficiency can be achieved on a fully programmable processor in the context of a future mobile terminal supporting 4G wireless and high-definition video coding.

Mobile devices have proliferated at a spectacular rate, with more than 3.3 billion active cell phones in the world. Soon, improvements to today's smart phones, such as high-bandwidth Internet access, high-definition video processing, and interactive video conferencing will be commonplace. The International Telecommunications Union has proposed fourth-generation (4G) wireless tech-

nology to increase bandwidth to maximum data rates of 100 Mbps for high-mobility situations and 1 Gbps for stationary and low-mobility scenarios like Internet hot spots (www.ieee802.org/secmail/pdf00204.pdf). This translates into an increase in computational requirements of 10 to 1,000 times over previous third-generation (3G) wireless technologies, with a power budget of approximately 1 W for all the computation. Other forms of signal processing, such as

high-definition video, are also up to 100 times more compute-intensive than current mobile video. Figure 1 shows the peak processing throughput and power budgets of 3G and 4G protocols. Conventional processors cannot meet these protocols' power-throughput requirements.

Research solutions, such as VIRAM and Imagine, can achieve the performance requirements for 3G, but generally exceed the power budgets of mobile terminals. The signal-processing on demand architecture (SODA) improved upon these solutions and could meet both the power and throughput requirements for 3G wireless (Y. Lin et al., "SODA: A Low-Power Architecture for Software Radio," *Proc. 33rd Ann. Int'l Symp. Computer Architecture*, 2006, pp. 89-101).

For 4G wireless protocols, the computational efficiency of mobile computer systems must be increased to greater than 1,000 Mops/mW. 4G uses three central technologies: orthogonal frequency-division multiplexing (OFDM), low-density parity check (LDPC) code, and multiple-input multiple-output (MIMO) techniques.

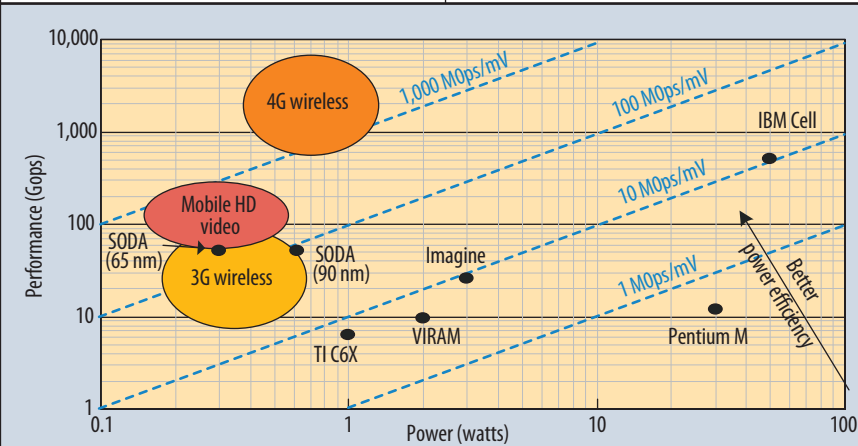


Figure 1. Peak processing throughputs and power budgets of 3G and 4G protocols. Conventional processors cannot meet these power-throughput requirements.

EMBEDDED COMPUTING

Fast Fourier transforms (FFTs) are key because they route signals from the baseband to the subcarriers. LDPC codes provide superior error-correction capabilities; however, parallelizing the LDPC decoding algorithm is more challenging because of the large amount of data shuffling. MIMO is based on the use of multiple antennae for both transmission and signal reception and requires complex signal-detection algorithms. The need for higher bandwidth and increased computational complexity are the main reasons for the two-orders-of-magnitude increase in processing requirements when moving from 3G to 4G.

milliWatt power budgets that today's cell phones require. However, such heterogeneous organizations are inefficient for companies to use in developing, building, and maintaining software. Further, as the amount of functionality integrated into their mobile terminal increases, hardwired solutions waste silicon area and power with many single-use hardware blocks.

Next-generation mobile computer system designs must address three issues: efficiency, programmability, and adaptivity. The existing computational efficiency of 3G solutions is inadequate and must be increased for 4G. As a result, straightforward scaling

Next-generation mobile computer system designs must address three issues: efficiency, programmability, and adaptivity.

High-definition video is also an important application that these platforms must support. Figure 1 shows that the performance requirements of HD video exceed those of 3G wireless, but are less than those for 4G wireless. However, the power budget dedicated to HD video is generally smaller. Moreover, the data access complexity in HD video is much higher than wireless, since algorithms operate on two- or three-dimensional blocks of data. Thus, HD video applications push designs to have more flexible, higher-bandwidth memory systems. HD video is just one example of a growing class of applications with diverse computing and memory requirements that next-generation mobile devices must support.

NEXT-GENERATION DESIGN STRATEGIES

3G mobile computer systems employ a combination of general-purpose processors, digital-signal processors, and hardwired accelerators to provide the giga-operations-per-second performance on

of 3G solutions by increasing cores or data-level parallelism will not suffice. Programmability provides the opportunity for a single platform to support multiple applications and even multiple standards within each application domain. Last, hardware adaptivity is necessary to maintain efficiency as the core computational characteristics of the applications change.

3G solutions rely heavily on the widespread amounts of vector parallelism in wireless signal-processing algorithms, but lose most of their efficiency when vector parallelism is unavailable or constrained, as happens with HD video.

Designing efficient architectures for future mobile systems requires analyzing the anticipated workloads. While performing detailed analysis of the computation kernels for 4G wireless and HD video encoding and decoding (h.264), we elicited five key insights:

- Opportunities for single-instruction, multiple data (SIMD) parallelism vary widely across

the algorithms. Some have large inherent vectors, up to 1,024 elements in length. However, most algorithms have small to moderate vectors.

- Algorithms with smaller vector lengths frequently contain a high degree of identical threads, where each thread performs the same instructions, but on discontinuous data.
- A large percentage of temporary values generated during the computation are short-lived and need not be saved to a register file.
- There is a small set of arithmetic instruction pairs that occur with high frequency.
- Each algorithm repeatedly uses a small set of predetermined data-shuffling patterns.

ANYSP MOBILE SUPERCOMPUTER

To address these challenges, we highlight the AnySP advanced signal-processing architecture proposed by researchers at the University of Michigan, Arizona State University, and ARM Limited (M. Woh et al., "AnySP: Anytime Anywhere Anyway Signal Processing," *Proc. 36th Ann. Int'l Symp. Computer Architecture, 2009*, pp. 128–139). AnySP seeks to create a fully programmable architecture that supports 4G wireless communication and HD video decoding.

Such a design would need to reach the computation efficiency levels of nearly 1,000 Mops/mW that only ASIC solutions have achieved previously. Programmability is recognized as a first-class design constraint, thus no fixed-function hardware blocks are employed.

To overcome the typical pitfalls of relying on SIMD parallelism across a wide variety of algorithms, a configurable SIMD datapath is created. This supports three execution scenarios: wide vector computation (64 lanes), multiple independent narrow vector computation threads, and chained computation subgraphs on moder-

ately wide vector computation. This inherent flexibility lets the data path be customized to the application while still retaining the high execution efficiency that SIMD offers by reducing control overhead. AnySP also attacks the traditional inefficiencies of SIMD computation: register file power, data shuffling, and reduction operators.

Figure 2 shows the AnySP processing element (PE) architecture, which consists of integrated SIMD and scalar data paths. The SIMD data path in turn consists of eight groups of 8-wide SIMD units, which can be configured to create SIMD widths of 16, 32, and 64. Each of the 8-wide SIMD units comprises groups of flexible functional units (FFUs). The FFUs contain the functional units of two lanes connected through a simple cross bar. Eight SIMD register files (RFs) feed the SIMD data path and each 8-wide RF has 16 entries. The data shuffle—or swizzle—network aligns data for the FFUs. It can support a fixed number of swizzle patterns of 8-, 16-, 32-, 64-, and 128-wide elements. Finally, a multiple-output adder tree can sum groups of 4, 8, 16, 32, or 64 elements, then store the results in a temporary buffer.

The local memory consists of 16 memory banks. Each bank is an 8-wide SIMD containing 256 16-bit entries, totaling 32 Kbytes of storage. Each 8-wide SIMD group has a dedicated address generation unit (AGU). When not in use, the AGU can run sequential code to assist the dedicated scalar pipeline. The AGU and scalar unit share the same memory space as the SIMD data path. To accomplish this, the design includes a scalar memory buffer that can store 8-wide SIMD locations. Because many of the algorithms access data sequentially, the buffer acts as a small cache that helps to avoid multiple vector-bank accesses.

CONFIGURABLE MULTISIMD WIDTH SUPPORT

The individual algorithms in the applications that we studied had varying SIMD widths. However, inde-

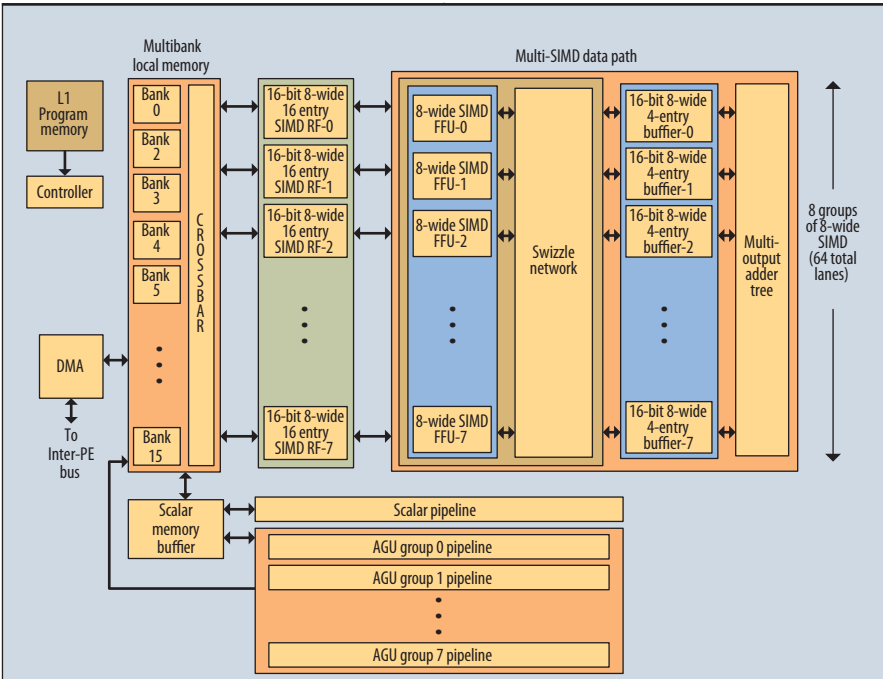


Figure 2. AnySP processing element. This processing element architecture consists of integrated SIMD and scalar data paths.

pendent threads were not dominant. Rather, the system would run the same task many times for different sets of data. Each task was independent of others, running the exact same code and following almost the same control path with the only difference being the set of memory addresses accessed.

To support these types of kernel algorithms, AnySP was designed as a multi-SIMD-width architecture. Each group of 8-wide SIMD units has its own AGU to access a different data stream. The 8-wide groups can also be coalesced to create SIMD widths of 16, 32, or 64. This feature lets the system exploit data and thread parallelism together for large and small SIMD-width algorithms.

Small SIMD-width algorithms like intraprediction and motion compensation from h.264 video decoding can process multiple macroblocks at the same time while exploiting the 8-wide and 16-wide SIMD parallelism within the algorithms. Meanwhile, large SIMD-width algorithms like FFT and LDPC can use the full 64-width SIMD.

TEMPORARY BUFFER AND BYPASS NETWORK

AnySP implements temporary register buffers and a bypass network to reduce power consumption and the number of RF accesses. The temporary register buffers are implemented as a partitioned RF. The main RF contains 16 registers, but the design also adds a second partition containing four registers, making 20 registers total. This small, partitioned RF shields the main RF from accesses by storing values that have short lifetimes.

The bypass network is a modification to the writeback stage and forwarding logic. Typically, in processors, data forwarded to eliminate data hazards is also written back to the RF. In the bypass network, the compiler explicitly manages the forwarding and writing to the RF to eliminate unnecessary RF writes.

FLEXIBLE FUNCTIONAL UNITS

In typical SIMD architectures, power and performance are lost when

EMBEDDED COMPUTING

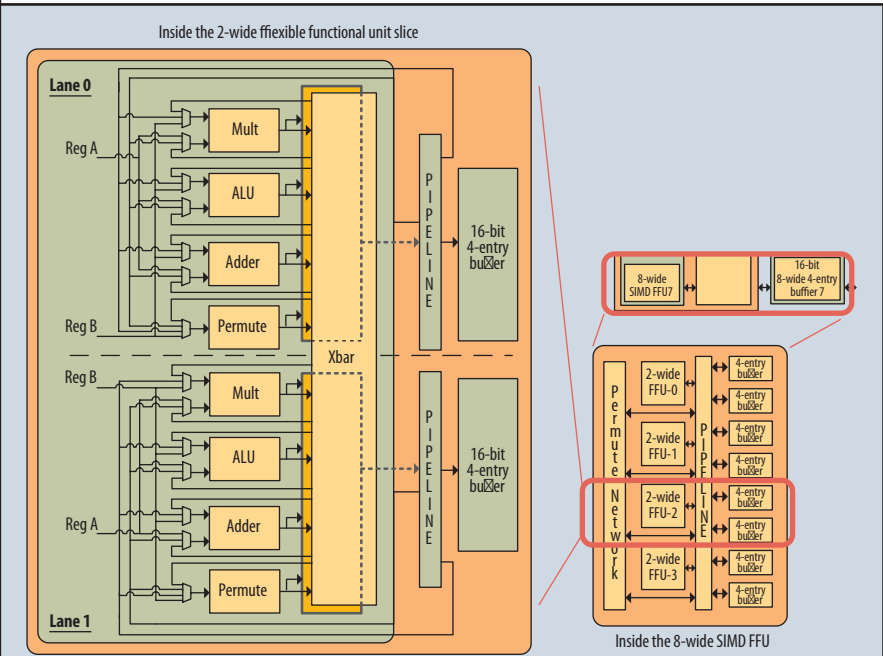


Figure 3. Design of a flexible functional unit that supports chaining of neighboring lanes to pipeline the execution of 2-deep computation subgraphs.

the vector size is smaller than the SIMD width as a result of underutilized hardware. AnySP adds another level of configurability to the data path by using FFUs as the core computation units. When SIMD utilization is low, the data path can be configured to chain the FFUs from neighboring lanes back-to-back. This effectively turns two SIMD lanes into a 2-deep execution pipeline. Two different instructions can be chained through the pipeline, and data is passed between them without writing back to the RF.

As Figure 3 shows, each 8-wide SIMD group is built from four 2-wide FFUs. Algorithms with SIMD widths smaller than 64 benefit from this structure. In chained-execution mode, the functional units among two internal lanes can be connected through a crossbar network. Overall, FFUs improve performance and reduce power by adding more flexibility. AnySP only chains pairs of lanes together, but the technique can be expanded to chain larger numbers of lanes.

SWIZZLE NETWORK

The number of distinct swizzle

patterns needed for a specific algorithm is small, fixed, and known in advance. Previous research has explored building application-specific crossbars for SIMD processors (P. Raghavan et al., “A Customized Crossbar for Data-Shuffling in Domain-Specific SIMD Processors,” *Proc. Architectures for Computing Systems* [ARCS 2007], LNCS 4415, Springer, 2007, pp. 57-68), but these lack flexibility because they cannot support new swizzle operations for applications that emerge postfabrication. AnySP uses an SRAM-based swizzle network that adds flexibility while maintaining the performance of a customized crossbar. The proposed network is similar to the work of N. Goel, A. Kumar, and P.R. Panda (“Power Reduction in VLIW Processor with Compiler Driven Bypass Network,” *Proc. 20th Int’l Conf. VLSI Design* [VLSID 07], ACM Press, 2007, pp. 233-238) in that the X-Y style crossbar lays out the input buses horizontally and the outputs vertically.

Each point of intersection between the input and output buses contains

a pass transistor controlled by a flip-flop. Multiple sets of swizzle configurations are stored in the SRAM cells, allowing zero-cycle delay for changing the swizzle pattern. By storing multiple configurations, many control wires can be removed, and the network’s area and power consumption can be reduced while still operating within a single clock cycle. For crossbar sizes larger than 32 × 32, power is dramatically lower than the MUX-based alternative and can run at almost twice the frequency. For example, a 128 × 128 SRAM-based swizzle network consumes less than 30 percent of the power consumed by an equivalent MUX-based crossbar.

Though only a certain number of swizzle patterns can be loaded at a time without reconfiguration, this approach provides a viable solution because only a small set of swizzle patterns are needed for each algorithm. The swizzle network has lower power and provides more functionality than the permutation networks found in typical SIMD architectures by also supporting multicasting capabilities along with the swizzle patterns.

MULTIPLE OUTPUT ADDER TREE SUPPORT


Many SIMD architectures have special SIMD summation hardware to perform reduction-to-scalar operations. To compute this, adder trees sum up the values of all lanes and store the result in the scalar RF. While this worked for 3G algorithms, many of the video-decoding algorithms needed sums shorter than the SIMD width. In the AnySP architecture, the adder tree allows for partial summations of 4 through 64 elements, which are then written back to the temporary buffer unit.

A 4-PE AnySP system running at 300 MHz with an ARM Cortex-M3 serving as the control processor met the throughput requirements of 100 Mbps 4G wireless while consuming 1.3 W at

90 nm. This falls short of the 1,000 Mops/mW efficiency target, but close enough to meet it in 45-nm process technology. H.264 video decoding at 30 fps is achieved with 60 mW at 90 nm, meeting the requirements for mobile HD video. The power breakdown of AnySP shows that the SIMD functional units dominate power consumption, followed by the register file and the rest of the data path. AnySP was designed to demonstrate that power efficiency can be achieved on a fully programmable processor in the context of a future mobile terminal supporting 4G wireless and HD video coding. Programmability is essential moving forward to provide a hardware substrate that allows the software to evolve naturally.

AnySP features a configurable SIMD data path that supports wide and narrow vector lengths; flexible

functional units, which can chain together narrow SIMD instructions using neighboring SIMD lanes; temporary buffers and a bypass network that reduce register and memory accesses; an SRAM-based swizzle network that reduces the power and latency of data-shuffling operations; and a flexible multiple-output adder tree, which speeds up video applications.

Industry will continue to build heterogeneous systems consisting of programmable processors and hardwired ASICs, but it is already trying to reduce the number of distinct intellectual property blocks in designs to reduce cost and manage complexity. We expect features such as those in AnySP to slowly integrate into mainstream mobile architectures, achieving a fully programmable, mobile supercomputer. 

Mark Woh is a PhD candidate in the Department of Electrical Engineering and Computer Science at the University of Michigan. Contact him at mwoh@umich.edu.

Scott Mahlke is an associate professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. Contact him at mahlke@umich.edu.

Trevor Mudge is Brecht Family Professor of Engineering in the Department of Electrical Engineering and Computer Science at the University of Michigan. Contact him at tnm@umich.edu.

Chaitali Chakrabarti is a professor of electrical engineering at Arizona State University. Contact her at chaitali@asu.edu.

Editor: Tom Conte, College of Computing, Georgia Institute of Technology; conte@cc.gatech.edu

Computer

Innovative Technology for Computer Professionals

Welcomes Your Contribution

**Computer
magazine
looks ahead
to future
technologies**



IEEE
computer
society

- **Computer**, the flagship publication of the IEEE Computer Society, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.
- Articles selected for publication in **Computer** are edited to enhance readability for the nearly 100,000 computing professionals who receive this monthly magazine.
- Readers depend on **Computer** to provide current, unbiased, thoroughly researched information on the newest directions in computing technology.

**To submit a manuscript for peer review,
see *Computer's* author guidelines:**

www.computer.org/computer/author.htm

INDUSTRY PERSPECTIVE

The Economic Case for Open Source Foundations

➔ **Dirk Riehle**, *Friedrich-Alexander-University of Erlangen-Nürnberg*



By establishing a successful open source platform, software firms can compete more effectively across technology stacks and thereby increase their addressable market.

An open source foundation is a group of people and companies that has come together to jointly develop community open source software. Examples include the Apache Software Foundation, the Eclipse Foundation, and the Gnome Foundation.

There are many reasons why software development firms join and support a foundation. One common economic motivation is to save costs in the development of the software by spreading them over the participating parties. However, this is just the beginning. Beyond sharing costs, participating firms can increase their revenue through the provision and increased sale of complementary products. Also, by establishing a successful open source platform, software firms can compete more effectively across technology stacks and thereby increase their addressable market. Not to be neglected, community open source software is a common good, creating increased general welfare and hence goodwill for the involved companies.

OPEN SOURCE FOUNDATIONS

The Linux operating system and the Apache webserver are popular

examples of open source projects that are in widespread industry use. They started out as volunteer projects without any commercial backing. When the industrial significance of these projects became apparent during the 1990s, interested software developers and firms decided to put the future of the software on more solid ground by creating nonprofit organizations.

Such an organization, commonly called a foundation, serves as the steward of the projects under its responsibility. It provides financial backing and legal certainty, making the survival of the software less dependent on the individuals who initially started it. There are many variants of foundations like trade associations and consortia. Each of them has its own matching legal structure, depending on the specific goals of the founders. This article uses the term foundation to denote all of them.

The foundation represents the community of developers, which is also why the software is called *community open source* (D. Riehle, "The Economic Motivation of Open Source: Stakeholder Perspectives," *Computer*, Apr. 2007, pp. 25-32; E. Capra and A. Wasserman, "A Framework for Evaluating Managerial Styles in Open

Source Projects," *Proc. 4th Int'l Conf. Open Source Systems* [OSS 2008], Springer, 2008, pp. 1-14).

Community open source is different from single-vendor open source, which is open source software that is being developed by a single firm. Firms behind single-vendor commercial open source expect direct revenue from selling the software and services for it (D. Riehle, "The Commercial Open Source Business Model," *Proc. 15th Americas Conf. Information Systems* [AMCIS 2009], AIS Electronic Library, 2009). This is typically not the case with communally owned open source, as competition is likely to keep revenues down.

However, there are several economic reasons why software firms join and support foundations to develop community open source: Some members expect cost savings for products built on the community open source software, others expect increased revenue and sales from complementary products, and yet others want to grow their addressable market.

ORGANIZATIONAL RESPONSIBILITIES

The main purpose of a foundation is to act as the steward of the software being developed and to ensure

its long-term survival. A foundation has various responsibilities, including the following:

- organize the project community;
- actively market the software;
- clarify and manage intellectual property rights;
- set strategic directions for the software;
- respond and remain accountable to its members; and
- run all relevant back-office processes.

Open source foundations are usually open to everyone to join; however, a membership fee may apply. Many of their processes are similar to those of traditional software associations and will not come as a surprise. What is different, however, is the provision of the main product as open source and the resulting intellectual property implications.

INTELLECTUAL PROPERTY MECHANISMS

Some eschew open source out of fear of a loss of intellectual property. Open source foundations solve this problem by providing well-defined processes that clarify any intellectual property rights issues associated with the software. In the end, the open source project becomes just like every other software and is provided under an open source license that spells out its usage conditions.

In software development, three main categories of intellectual property rights must be considered:

- copyright (to the source code and related texts),
- trademarks, and
- patents.

The contributors to the project provide the relevant intellectual property. Most foundations define the relationship between a contributor and the project using a so-called contributor agreement. Any legal

entity that wishes to contribute to the project, whether a member of the foundation or not, must sign this agreement.

A common practice of open source foundations is to own the copyright to all source code and related texts. Thus, the contributor agreement is set up so that the contributor, be it a company like IBM or a volunteer programmer, signs over the copyright of any current or future contributions to the foundation. (In a weaker form, sometimes only a relicensing right is required.)

Open source foundations provide well-defined processes that clarify any intellectual property rights issues associated with the software.

Using this mechanism, the foundation becomes the sole owner of the copyright. It is important that there be only a single owner: Decision making is with the foundation rather than a distributed group of diverse copyright holders. The foundation can now define and enforce the license terms under which the project is made available to the public and can defend the software in court.

The choice of the license depends on the foundation's goals. Most foundations choose a liberal license to allow for the widest variety of use circumstances of the software by its members. Such a liberal license typically allows embedding the software in other software packages without requiring the open sourcing of these other packages.

An important practice of a foundation is to ensure that no source code is contributed from another open source project with an incompatible license. The specific fear is that the contribution of incompatible code would require an undesired

change of license, as might happen, for example, with the contribution of GPL-licensed code to Apache-licensed code. The GPL license is the original reciprocal ("viral") open source license that requires all derived code to have the same license as well. "Keeping the code clean" is a prime directive at many foundations.

Naturally, the foundation also becomes the owner of the software trademarks and acts to enforce them. Thus, the foundation becomes the trustee of both the source code and its trademarks.

Finally, the contributor agreement clarifies the use of software patents. Source code implements software patents. Even if the foundation owns the copyright to the source code, without further measures, users of the software may still have to pay royalties to the holders of the patents implemented by the software. This can become particularly nasty if royalty requests surface only after the software has been put to use in a user organization. For this reason, the contributor agreement typically requires contributors to provide a general (perpetual, unrestricted, royalty-free) usage grant of the patent to all users of the open source software. This protects users from unanticipated patent royalty requests.

SHARING DEVELOPMENT EXPENSES

There are many economic reasons to start, join, or support an open source foundation. The original and still most widely known reason is cost savings realized by standardizing on one platform and sharing its development expenses.

Consider the situation of Unix and Linux desktops in the late 1990s: Several competing windowing systems existed, each with incompatible desktop applications, and all of them configured and deployed differently, depending on the Unix or Linux distribution they came with. By then, Unix had lost the competition for

INDUSTRY PERSPECTIVE

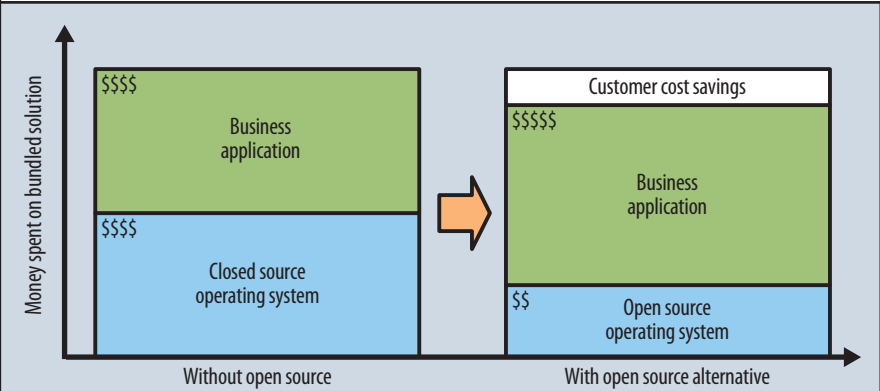


Figure 1. The support of open source software lets vendors sell at a higher price.

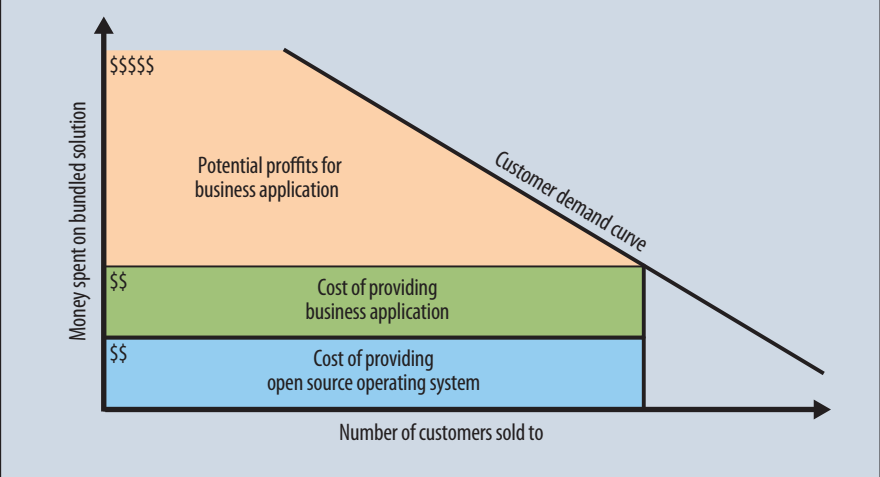


Figure 2. The support of open source software lets software črms sell to more customers.

the users' desktop to Microsoft Windows. Graphical user interfaces for Linux were a pure cost position for the distributors.

In this situation, any good-enough desktop software would do for the involved software firms. Distributors like Red Hat and SUSE (Novell) as well as IBM and HP decided not to compete on the merits of their desktop configuration but rather to support a common desktop environment. This led to the continued development and consolidation of the GNOME and KDE desktop environments, formally supported by the GNOME foundation and the KDE e.V., respectively. These two foundations remain volunteer efforts, however, with strong corporate support.

It is not always the software firms that start or grow a software

foundation. Cost savings through community open source can have multiple roots. Sometimes, customers join forces to create a foundation and require that any software development work by a contractor utilize and develop the open source software further. Currently, the US healthcare industry is undertaking steps in this direction.

PROFITS PER SALE IN A GIVEN MARKET

Beyond cost savings in research and development and in user organizations, original software development firms can use open source foundations to their competitive (economic) advantage.

The initial thrust behind company contributions to Linux and the Apache Software Foundation projects

focused on supporting an alternative to more expensive closed source solutions. If, for example, a company is selling a business application that also needs an operating system to run, more money will be available for the business application vendor if no money is spent on the operating system license and its maintenance fees. Hence, for the customer, an open source alternative saves money, while for the business application vendor more money becomes available, at the expense of the closed source operating system vendor, who misses a sale. Figure 1 illustrates this economic situation.

An early example of this mechanism is IBM's support of Linux. Realizing that OS/2, IBM's then-competitor to Microsoft's Windows, was losing in the marketplace, IBM threw its weight behind Linux and related open source projects. Having an alternative to Windows meant that IBM could keep Microsoft's license fees in check when selling to customers.

In general terms, replacing a high-cost closed source component of the technology stack with a lower-priced open source component increases pricing flexibility for the vendors of the other components in the stack. It also reduces costs for customers and makes more money available for other purchases.

INCREASED SALES IN A GIVEN MARKET

A second consequence of the increased pricing flexibility is that a software developer can sell to more customers than before. Some customers are more price-sensitive than others. Figure 2 illustrates this as the customer demand curve. Going down the demand curve from left to right, the price for the business application plus operating system bundle goes down, and more customers are willing to buy. In simplified terms (a nontransparent market), the developer stops selling only if the price has come down to its own total cost.

Thus, replacing the more expensive closed source operating system with a lower-cost open source alternative reduces the lowest possible price point for the bundle. This lets a vendor sell to more customers, which leads to more profits.

Higher profits on a given sale and more profits by selling to more customers are two important reasons why a software firm may support open source software that is complementary to its own product line. From the firm's perspective, supporting the open source software is a subsidy, paid out of increased profits from its own product. Basically, the open source alternative lets the firm shift revenues from a complementary product, owned by someone else, to its own product.

GROWING THE ADDRESSABLE MARKET

The size of the market a software firm can sell into depends on the platforms on which it is based. If a vendor builds on a platform that customers aren't willing to operate, the firm's products will not be considered. Thus, the choice of the platforms a firm's product runs on is crucial. As indicated, an open source platform is economically more beneficial than a closed source platform. Thus, the software development firm should support an appropriate platform and encourage other vendors to do the same. More and better applications will grow the value of the platform to customers. With growing acceptance of the platform, more customers will be operating it, first increasing the total size of the market and then the size of the market that the software firm's products can address.

Figure 3 illustrates the dynamics of shrinking a closed source platform to the advantage of an open source platform. The money leaving the market around the closed source platform enters the market for products built on top of the open source platform. As customers review the choice of

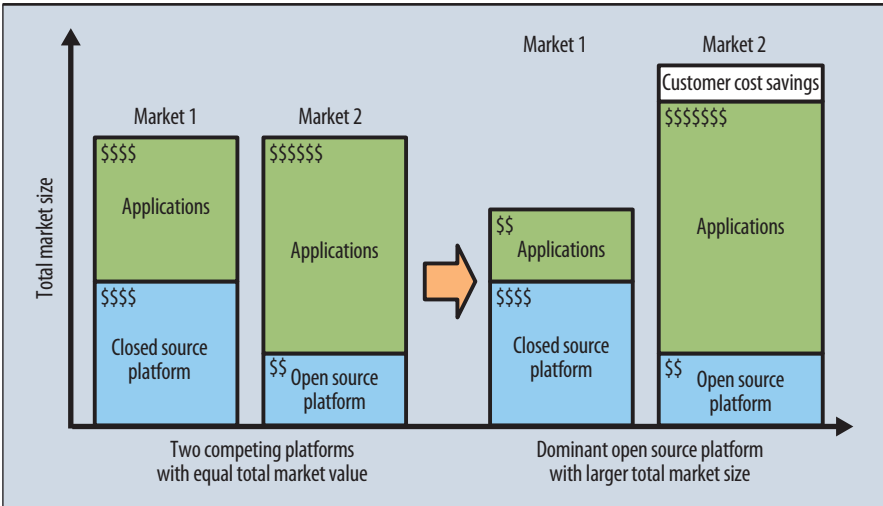


Figure 3. Growing an open source platform increases the total market size.

products available, they prioritize purchases anew in accordance with what's available and how big their IT budget is. This dynamic is particularly attractive to the providers of mission-critical applications, which typically get higher purchasing priority than less important, more incremental applications.

Participating in the development of the open source platform is of strategic interest to a software firm. It ensures visibility of the firm to potential customers and promises high technical quality of its software products. Gaining a strong position in the foundation and development processes of the software creates a significant positional advantage over later competitors.

There are more platforms or layers in the technology stack than some might think. The obvious platforms are operating systems and middleware solutions. Beyond this, many more potential platforms exist, addressing vertical as much as horizontal slices of the stack. Whether it is platforms for business accounting or medical imaging, automotive software buses or electronic patient records, we can expect a wealth of new domain-specific open source platforms to appear in the coming years.

A firm should consider creating community open source and sup-

porting an open source foundation if it is not only competing within the same stack, but across stacks. Linux, the Apache projects, and the Eclipse platform can all be viewed as software platforms on which revenue-generating applications are built. These platforms compete with closed source alternatives, for example, the Microsoft set of platforms, namely Windows, ASP.NET, and Visual Studio.


A reliable platform attracts other software vendors that might base their own products, whether provided as open source or not, on this platform. The increasing richness of functionality around a given platform benefits everyone: Customers cannot go wrong in deciding for this platform. Moving customers from a not-supported platform to a firm's own platform increases the size of the addressable market, which is likely to lead to more sales.

Every software development firm today should ask which open source foundation to support or, if necessary, to found. The benefits are clear: Done right, the firm can expect cost savings, increased profits per sale, a higher number of sales, and a larger addressable market. The question then becomes one of investment: How much to

INDUSTRY PERSPECTIVE

invest and what return to expect. At present, we lack economic models and decision processes to answer these questions.

The open source research group at the Friedrich-Alexander-University of Erlangen-Nürnberg and its collaborators are working on this question. In addition, we are looking at the processes and tools used by open source foundations and in open source software development in general. In collaboration with the Open Source Business Foundation, an international nonprofit

organization located in Nuremberg, Germany, we are making our research findings available to industry. Finally, we are interested in supporting public policy decisions with economic and technical insight to help increase general welfare through high-quality community open source. 

Dirk Riehle is the professor for open source software at the Friedrich-Alexander-University of Erlangen-Nürnberg. Contact him at dirk@riehle.org.

Editor: Sumi Helal, Department of Computer and Information Science and Engineering, University of Florida; helal@cise.ufl.edu

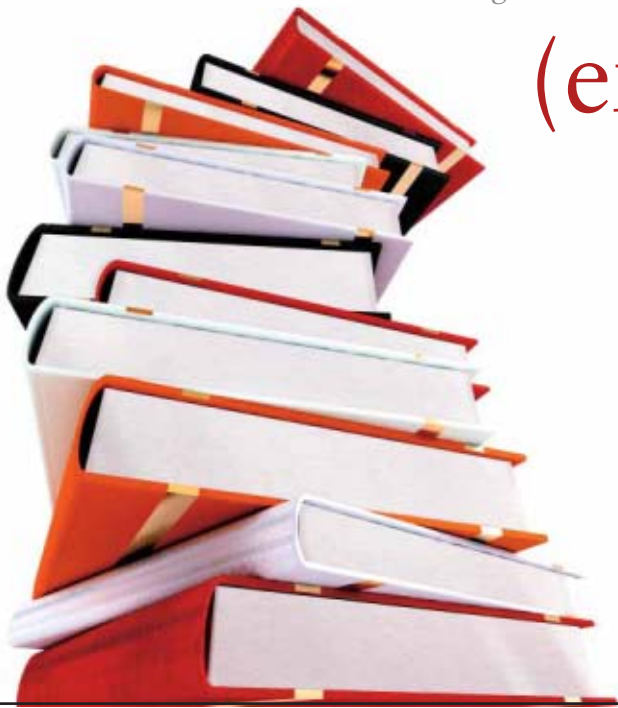
Readers are encouraged to use the message board at www.computer.org/industry_perspective to post comments, offer feedback, or ask questions.

cn Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.

“All writers are vain,
selfish and lazy.”

—George Orwell, “Why I Write” (1947)

(except ours!)



The IEEE Computer Society Press is currently seeking authors. The CS Press publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. It offers authors the prestige of the IEEE Computer Society imprint, combined with the worldwide sales and marketing power of our partner, the international scientific and technical publisher Wiley & Sons.

For more information contact Kate Guillemette, Product Development Editor, at kguillemette@computer.org.

 **CS Press**
www.computer.org/cspress

 **WILEY**
Publishers Since 1807

GREEN IT

Proxying: The Next Step in Reducing IT Energy Use

➔ **Bruce Nordman**, *Lawrence Berkeley National Laboratory*
➔ **Ken Christensen**, *University of South Florida*



Proxying is a simple and effective means of allowing network hosts to sleep while maintaining network presence.

At the core of Green IT is reducing the energy use of the electronic devices that help us to acquire, store, process, and display information. The energy use of PCs, set-top boxes, and even most servers is mostly driven by their operating patterns and low utilization. This results in annual energy use being dominated by the power required simply to make the devices be present and available, not by the incremental energy for actual useful computations. Thus, the greatest savings can be attained by reducing the power needed to maintain functional presence when idle, not by making the active operation more efficient.

PCs in the US use about \$7 billion of electricity per year (plus several billion dollars more for displays). Most of this energy use occurs when no one is present and the PC is idle. A major and increasing reason for PCs to be left on is to be continuously available for use on the network—that is, to maintain “network presence.” A host that fails to do this will not be reachable or addressable on the network, will not be manageable, and will lose some application state. While today’s PCs have a reliable sleep mode with quick wake-up, they lose network connectivity while asleep. The key is

to combine the compelling energy-saving characteristic of sleep while not sacrificing network presence.

NETWORK PRESENCE

Presence for communications is not new. Telephones maintain presence on the network between calls, to awaken any time a call comes in, and televisions listen for a remote-control signal to wake up. In these cases, a device that did not respond appropriately would be understood as failing a basic function.

PCs are increasingly used in ways that call for continuous presence on the network. This is part of the industry-wide transition from a two- to a three-state power model. At one time, electronic products were simply on or off. Beginning several decades ago, some devices added a third basic power state: sleep. Maintaining state and network connectivity are becoming the defining characteristics of sleep. Responsiveness to network activity will in the future likely be the key differentiation between sleep and off. A typical sleeping PC consumes less than 10 percent of its power when on (even if idle).

Presence on the network can be understood as having three “layers” (these are not intended to be the Open System Interconnection layers, but rather abstractions):

- link-layer presence for maintaining local network connectivity—for example, Wi-Fi or Ethernet;
- network-layer presence for supporting end-to-end transport; and
- application-layer presence for functionality.

Maintaining an Ethernet link while asleep (or even off) has been common in PCs for years; maintaining a Wi-Fi link in sleep is not currently done. Basic network connectivity requires active participation in numerous core Internet protocols, such as the Address Resolution Protocol (ARP) to maintain reachability. Application requirements vary: Some include activity for a system while it is asleep, but all require awareness of when the host needs to wake up for that application.

PROXYING

Proxying is the use of a low-power entity to maintain presence on the network for a high-power device like a PC. The component that does this, the proxy, can be internal to a device (for example, in the network interface chip within a PC), in an immediately adjacent network switch or router, or even another PC on the subnet. The proxy enables a host

GREEN IT

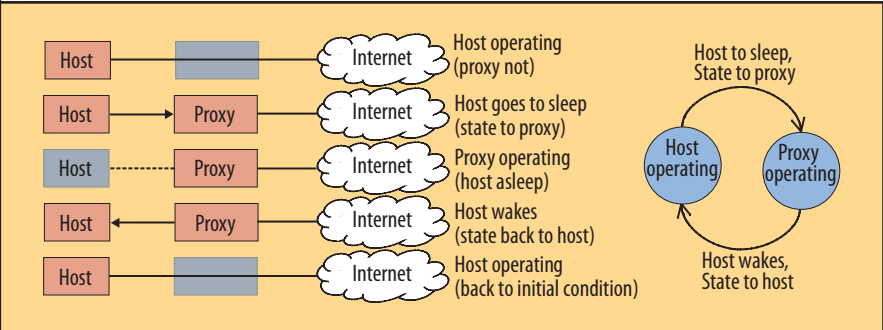


Figure 1. Proxy operation. A proxy enables a host to transition into and out of sleep transparently to the network. When the host is operating normally, the proxy function is not engaged.

to transition into and out of sleep transparently to the network. Use of a proxy requires no infrastructure changes such as changing existing protocols, or maintaining state in routers or switches.

As Figure 1 shows, a PC going to sleep signals the proxy that it needs to begin operation and then passes key information about its network state to the proxy. The host ceases to receive network traffic, and the proxy does the following on behalf of the host:

- maintain the network link,
- respond to packets,
- maintain presence state, and
- generate packets

Most incoming packets require no action and are thus simply ignored. The proxy can handle the majority of the remaining packets with routine responses. When the proxy finds network traffic that it cannot handle (that needs the host’s attention), it wakes the host. As the host wakes, it might receive some information from the proxy—for example, the packet that triggered the wake.

Once the mechanics of proxying are in place, the actual operations that the proxy undertakes are not necessarily complicated, but failure to carry them out is catastrophic for network presence.

Systems that are presently left on for many hours while not in use (often 24/7) can save large amounts

of energy by instead being asleep most of the time and waking only when needed to execute required tasks. However, hosts that are today powered down when not in active use will still benefit; doing so will not save energy, but the devices will gain additional functionality through network presence they previously lacked when in sleep. Thus, the addition of proxying functionality can benefit all PCs.

STANDARDIZATION

Functions that act on a network usually require standardization to enable interoperability. Proxying involves coordination among operating systems, applications, and the proxy itself. The EPA Energy Star program recognized the value of proxying in its Version 5.0 Specification for Computers but stipulated that it should be defined by a standard. Then, major PC companies, including AMD, Apple, Intel, Microsoft, and Sony came together under the auspices of the Ecma International standards organization to create such a standard, which they completed in November 2009.

The Ecma proxying committee, TC38-TG4 (www.ecma-international.org/memento/TC38-TG4.htm), considered various typical consumer and enterprise usages, covering all three of the aspects of network presence: link (Ethernet and Wi-Fi), network (IPv4 and IPv6), and several applications.

For network connectivity for IPv4 networks, the standard includes key protocols such as ARP, the Dynamic Host Configuration Protocol (DHCP), and the Internet Group Management Protocol (IGMP). Application-level proxying needs vary: Some require only basic link and network presence; some need only basic wake-up mechanisms—for example, waking on a Transmission Control Protocol (TCP) SYN packet; and for others, the proxy must undertake specific actions.

The Ecma standard addresses several application-oriented features, including the Simple Network Management Protocol (SNMP), remote wake-up, and remote access for IPv6 with Teredo. The standard also covers service discovery with multicast DNS (mDNS) and link-local multicast name resolution (LLMNR).

IMPLEMENTATION

Several research efforts have focused on proxying. Most notable is the Somniloquy project at the University of California, San Diego, and Microsoft Research; it is a proxy residing on USB-attached Gumstix hardware with both Ethernet and Wi-Fi connectivity. Somniloquy specifically explored the needs of applications for proxying.

Intel Research Berkeley conducted extensive trace analysis to uncover what a proxy needs to do, and it also implemented a simple proxy not requiring coordination with the host. Intel Labs conducted a technical session at Intel Developer Forum 2009 on proxying and also demonstrated Wi-Fi and remote wake-up (using the Session Initiation Protocol) capabilities.

Proxying is also making its way into various products. In mid-2009, Apple introduced proxying functionality for its PCs, with the proxy residing in Apple-brand access points or in a nonsleeping Mac on the same subnet. More recently, Apple has begun shipping systems with the proxy functional internal to

the PC. The Apple implementation focuses on Bonjour (mDNS), which enables media sharing among other functions.

Microsoft's Windows 7 includes support for network interface chips that can respond to ARP and neighbor discovery (ND) queries.

FUTURE CHALLENGES

As noted above, the proxy can be either internal to the sleeping device or external to the sleeping host. In the long run, an internal proxy offers ease of coordination with the host operating system and ensures that the device can reliably proxy within complex configurations. In the near term, however, external proxying can utilize existing network equipment and proxy for existing PCs and so be useful and save energy much more quickly—a single piece of network equipment might contain proxies for many attached hosts. Thus, both approaches are necessary.

The next step for proxying is wide deployment in enterprises. For general proxying, deployment of external proxies in commercial businesses first has several advantages:

- it can be implemented with existing equipment, and a single IT person can enable the functionality for many PCs;
- a business that saves energy for dozens or hundreds of PCs will see economic savings that are much larger than a household would; and
- many business network environments and applications are more uniform than those of households in general are, reducing the number of complications or problems in implementation.

Proxying should also be useful for battery-powered devices to reduce the amount of communications, computation, and required on-time, therefore extending battery life.

Proxying raises the topic of what

network and application protocols and behaviors are truly critical for a modern host to support. This same question has arisen in Internet Engineering Task Force discussions of what very simple hosts—such as lights, communicating thermostats, and so on—need to support to enable good operation and interoperability. This may provide an extra push for simplicity in the network realm.

For applications that are not compatible with the proxying standard, deployment will provide incentives to update the application to accommodate proxying. The biggest frontier for proxying is likely to be the handling of specific applications and application-oriented protocols.

Several promising directions for proxying to help reduce the energy use of IT equipment include:

- Proxying for security-related protocols such as Internet Protocol Security (IPSec). The challenge here is how to transfer security-related state (passwords and so on) to and from a proxy and how the proxy can be trusted.
- Proxying P2P protocols to allow hosts running them to power down when not actively downloading, uploading, or streaming content. Currently, P2P hosts must remain fully powered on all the time. P2P hosts already use significant energy and this use is expected to grow, making this an area of urgent interest.
- Adding more application-level functionality into a proxy—for example, the ability to store and share content.
- Enabling proxies to coordinate activities and thus be able to consolidate IT services to further reduce energy use.
- Extending proxying to non-IT equipment such as residential appliances that are rapidly being connected to the Internet as network hosts.

Network connectivity proxying offers a chance to both save large amounts of energy and add functionality, at very low cost. As implementation requires coordination of numerous entities and organizations, basing it on an industry standard greatly increases the chances that it will succeed and be widely deployed sooner. Proxying offers the opportunity for hosts and applications to expose their power state when desired, or to hide low-power states from the network when that is more appropriate. This brings power states and power management to higher-layer protocols, which historically have lacked them.

Bruce Nordman is a researcher in the Energy Analysis Department at Lawrence Berkeley National Laboratory. Contact him at bnordman@lbl.gov.

Ken Christensen is a professor and director of the undergraduate program in the Department of Computer Science and Engineering at the University of South Florida. Contact him at christen@cse.usf.edu.

Editor: Kirk W. Cameron, Dept. of Computer Science, Virginia Tech; cameron@cs.vt.edu

build
your
career

IN COMPUTING

www.computer.org/buildyourcareer

THE PROFESSION

Continued from page 96

than 70 minutes of television daily. For seven and eight-year-olds, viewing time rises to an average of almost two hours” (tinyurl.com/y8bapkn). How will such children learn to tell fact from fiction, coercion from instruction, and good from bad? And now videogames are being made for young children.

Conceptual reality is the basis of culture. The richness of culture springs from the depth of contemplation, and from the ability to analyze perceptions and choose from a range of responses.

Living in society means that each member’s subjective reality deals with that of others. Interaction is complex, springing from perceptual reality and involving various degrees of attention, contemplation, and response, and various numbers of interactants from time to time.

Traditionally, when people were in company they interacted in various ways. They learned how to interact successfully in their childhood when they interacted with other children and with parents and teachers, who fostered the development of good

is said to bring (tinyurl.com/yk4jxrr).

The other kind of interactive reality relates to society’s perception of the world we live in. One individual’s subjective reality is not the same as another’s. To interact successfully, people must reconcile each other’s subjective reality. This is easily done if we are in the same place, speak the same language, and are prepared to give and take. But in the long term and in matters of detail, a shared, valid physical reality is much more difficult to achieve. And technologists must understand physical reality if they are to successfully change it.

The difficulty of understanding physical reality means that small subsocieties of experts—scientists, mainly—must concentrate on measuring and modeling their field of physical reality. This must be done numerically and mathematically, which is where digital technology comes in strongly. Their findings are available for technologists to exploit, and for interested others to learn from.

Different components of society at large will apply the findings of scientists in different ways and at different levels of understanding. One common misunderstanding is that science is composed of theories that might or might not be true. In reality, science is an ongoing endeavor and theories accepted by a community of scientists are, if the community is working properly, the truth as it is understood so far. But physical reality is so complex and changeable that scientists continually work, through measurement and mathematical contemplation, to improve their theories, just as breeders work to improve their stock.

CLIMATE CHANGE

The weird thing is that people use digital machinery that has only been made possible through the work of scientists to try to discredit the work of scientists. The most obvious case of this is the recent work of the climate

Digital technology tends to hide social reality from the individuals using it.

In the past, science and technology have enriched conceptual reality by providing more to contemplate. For humans, contemplation is facilitated and extended by language, which provides the means to make fine distinctions and to better remember past experience.

By contrast, the aim of television and much digital technology is to capture and keep attention and to promote unthinking reaction, for example, when shopping in a supermarket. This diminishing of contemplation erodes personality and individuality. The pity is that digital technology could be used to extend the opportunity for personality development, in particular through DVDs and the Internet, by giving users individual control over the representations they watch and the vocabulary they use to exercise that control (for example, see *The Profession*, March 2008, pp. 104, 102-103).

INTERACTIVE REALITY

People do not usually live in isolation. Indeed isolation has been used as a punishment, and solitary confinement is arguably a form of torture (tinyurl.com/c4feho).

interactive skills. Good interactive skills were those that considered others as equals with rights and duties to be respected. This was a healthy social reality.

Television and digital technology are changing social reality for many. The overloading of perceptual reality and the stunting of conceptual reality bring a selfishness that lessens respect for others, and even respect for law and order. This is particularly evident in the marketing of consumer products that typically promote sensual satisfaction.

Digital technology tends to hide social reality from the individuals using it. Much is made of social networking on the Internet, but that social reality is gaunt compared to networking in physical proximity.

Readers might have noticed my use of TinyURL.com to save space in my essays, a very simple, impersonal facility. What a contrast to the more recent bit.ly that not only shortens your URLs but will “track the performance of your bit.ly links in real time” and provide “the complete history of your bit.ly links.” This seems to me to be offering the same kind of social unreality or clutter that e-mail

change deniers (tinyurl.com/yfjs5j5; tinyurl.com/ykbnkjs).

The issue of climate change is extremely important and multifaceted (see, for example, tinyurl.com/yzjjf7t). During the December 2009 United Nations Climate Change Conference in Copenhagen (en.cop15.dk), a huge amount of reporting took place, much of it speculation. The turmoil in America is particularly significant because on the one hand, the “Environmental Protection Agency has formally declared that greenhouse gases endanger human health” (tinyurl.com/ygbnrh3), while at the same time, “Only 45 per cent of the 1,041 adults surveyed on December 2-3 believed global warming was a proven fact” (tinyurl.com/ye8l9hs). And there is also turmoil in Australia (tinyurl.com/yl3kfpd).

When I last wrote in this column about climate change (Feb. 2005, pp. 104, 102-103), my emphasis was on the need for the profession to support increased collection of data. Since then, William F. Ruddiman’s book, *Plows, Plagues, and Petroleum: How Humans Took Control of Climate*, has been published (tinyurl.com/yjte7sw). This book takes a look at the Earth’s climate on a scale of millennia in a very convincing way.

The main influence on the Earth’s climate is insolation. This is cyclic in a complex way because of three variations in the Earth’s orbit: eccentricity, axial tilt, and precession (tinyurl.com/jd7cl). The cycle is of long ice ages separated by relatively brief interglacial periods. We are at the end of the most recent interglacial period, and temperatures started declining 10 millennia ago and should still be doing so. The temperature change has stalled variously, for example, eight millennia ago when agriculture with plows was developed, and is now going up when it should be going down. This conclusion is based on data extracted from ice cores.

There are uncertainties about the details of this argument, but the scale

of time considered puts the quibbles of climate change deniers focusing on the last decade or so into stark perspective (tinyurl.com/yhmysxt). Further, the people who argue for a gradual adoption of countermeasures must be told, first, that projections of the early stages of the lead-up to Copenhagen underestimated the rate of change in many ways. Second, there is a real danger of a “sticking point” being reached, that is, of positive feedback setting in—and it might have already done so—against which even completely eliminating anthropogenic warming factors would be ineffectual.

Many have reached the consensus that an international agreement is urgently needed either out of Copenhagen or consequently. The role of the computing profession will be vital in the likely measures prescribed by such an agreement. Perhaps the most important is verifying that the agreed measures are being taken and evaluating how effective they are.

An important component of adapting to climate change is building up the capabilities and productivity of Third World countries. A large part of this must come through accelerated education and training, and digital technology provides an essential component.

In all the many roles for the computing profession in coping with climate change is that of the system engineer. As a former engineer, I wonder whether governments at Copenhagen and after will look at the benefits of agreeing internationally to make marketing costs a use of after-tax profits rather than a pre-tax business expense. Such a measure would move the emphasis of economics away from consumption toward construction. To be practical, it would have to be done in stages, though this implementation technique is much more familiar to system engineers than to politicians.

However, a more important task for computing professionals will be to reverse the degeneration of realities that poor use of digital technology is supporting. Computers and the Internet should be used to promote balanced subjective reality in individuals, equitable social reality in communities everywhere, and a deeper understanding of physical reality in all levels of society. **□**

Neville Holmes is an honorary research associate at the University of Tasmania’s School of Computing and Information Systems. Contact him at neville.holmes@utas.edu.au.

cn Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>

Call for Papers | General Interest

IEEE *Micro* seeks general-interest submissions for publication in upcoming issues. These works should discuss the design, performance, or application of microcomputer and microprocessor systems. Of special interest are articles on performance evaluation and workload character-

ization. Summaries of work in progress and descriptions of recently completed works are most welcome, as are tutorials. *Micro* does not accept previously published material.

Check our author center (www.computer.org/micro/author.htm) for word, figure, and reference limits. All submissions pass through peer review consistent with other professional-level technical publications, and editing for clarity, readability, and conciseness. Contact *IEEE Micro* at micro-ma@computer.org with any questions.

IEEE
micro

THE PROFESSION

The Varieties of Reality

➔ **Neville Holmes**, *University of Tasmania*



Digital technology is being used to distort and corrupt reality.

The phrase virtual reality has an oxymoronic flavor, as acknowledged in Wikipedia (tinyurl.com/edg7z). In essence, it is an extension of drawing and painting, though digital technology is making it a drastic extension. Photography had a better right to the term, but digital technology has now given the lie to the old saying “the camera never lies.”

More recently the phrase *augmented reality* has appeared in Wikipedia (tinyurl.com/2buf25) and popular writing on computing. An essay in *The Atlantic* by Jamais Cascio (tinyurl.com/yh53r6d) discussed how this technology could “strike a fatal blow to American civil society” and presumably to other societies as well.

This attack on reality seems to be a theme nowadays in digital technology, and it’s hard to say where it’s likely to go in the long run. A recent news item describes work toward “a world where your contact lenses double as a personal computer display, superimposing information in front of you” (tinyurl.com/yjab5nt).

The professional issues here are many and various, and deserve concentrated evaluation by computing professionals.

WHAT IS REALITY ?

The main definition of “reality” in the *Oxford English Dictionary*, second edition, is “the quality of being real or having an actual existence.” The difficulty here is that, for any individual, quality stems from observation and evaluation. Thus, reality is subjective for individuals.

Digital technology affects subjective reality by changing what individuals experience and what they make of what they observe. Otherwise, reality is an interactive construct. Physical reality is built by the consensus of those actively concerned in defining and understanding particular classes of things. Social reality is built by the interaction of people living within a physical reality that they exploit and change.

For example, a particular science is continually developed by collaboration of specialists in the area of reality specific to that science, and a particular technology seeks to change social reality in an area of social activity by exploiting the findings of scientists.

Digital technology sits behind both science and technology. After all, language is the digital technology behind human social development, and the digital machinery we now use so widely has a profound effect on both social and physical reality.

SUBJECTIVE REALITY

People are individuals because everyone has a different personality. Personality changes through experience. Experience is the combination of what we perceive and what we make of it.

Perceptual reality comprises what our senses tell us about ourselves and our surroundings. Conceptual reality comprises attention, contemplation, and response.

History tells us how human society and its technologies have changed the content of perceptual reality, though that content has changed much more in developed countries. In particular, photography, radio, television and all-conquering modern digital technology have for many in the developed world completely changed the balance of what we perceive from predominantly actual to predominantly representational. A representation has a reality of its own, but that is not the reality of what is represented. Listening to rock music on your iPhone is not at all the same as listening while attending a rock concert.

The implications of this are profound, especially for the very young. Two years ago, researchers in Australia found that “three and four-year-olds on average watch more

Continued on page 94

ANNOUNCING A NEW STUDENT MEMBER PACKAGE FOR 2010!

Join IEEE and the IEEE Computer Society and enjoy FREE access to the Computer Society Digital Library for only \$40

Now is the best time to become part of the world's leading technical community and benefit from numerous networking and real-world learning opportunities. And, student members have access to the Computer Society Digital Library (CSDL).

Whether you are looking for the latest research on today's hottest topic or quick answers to a problem, CSDL has the information you need. In addition to over 3,500 conference publications, CSDL includes

- Access to *Computer* magazine—featuring cutting-edge research and articles written by leading experts in the field
- All 27 Computer Society peer-reviewed periodicals covering the spectrum of computing and information technology—with access to the complete archives

Student members also receive

- Access to development software from Microsoft, including Visual Studio Team System, Vista Business Edition, and Expression Web Designer
- Access to 600 selections from Safari® Books Online, featuring technical and business titles from leading publishers such as O'Reilly Media, Addison Wesley, and Cisco Press
- Access to 3,000 courses powered by Element K® and available in numerous languages
- Valuable networking opportunities through membership in your local chapter

There has never been a better time to join both IEEE and the IEEE Computer Society.

Become a student member today for just \$40 by visiting
www.computer.org/stuoffer



Running in Circles Looking for a Great Computer Job or Hire?



The IEEE Computer Society Career Center is the best niche employment source for computer science and engineering jobs, with hundreds of jobs viewed by thousands of the finest scientists each month - in **Computer** magazine and/or online!



careers.computer.org

<http://careers.computer.org>

- > Software Engineer
- > Member of Technical Staff
- > Computer Scientist
- > Dean/Professor/Instructor
- > Postdoctoral Researcher
- > Design Engineer
- > Consultant

The IEEE Computer Society Career Center is part of the *Physics Today* Career Network, a niche job board network for the physical sciences and engineering disciplines. Jobs and resumes are shared with four partner job boards - *Physics Today* Jobs and the American Association of Physics Teachers (AAPT), American Physical Society (APS), and AVS: Science and Technology of Materials, Interfaces, and Processing Career Centers.

IEEE
computer
society