

Computer

Innovative Technology for Computer Professionals

June 2007


**The Camino
Real,
p. 6**

**Apple's
iPhone,
p. 94**

**Collaboration
Environments,
p. 108**

<http://www.computer.org>



IEEE  computer society

Your potential. Our passion.™
Microsoft

Moore's Law. Fermat's Theorem. Euler's Function. _____ 's Law.



Adam Siepel
Cornell University



Magdalena Balazinska
University of Washington



Josh Bongard
University of Vermont



Yixin Chen
Washington University
in St. Louis



Luis Von Ahn
Carnegie Mellon University

Who's next? The Microsoft® Research Faculty Fellowship Award recognizes exceptionally talented computing academics with the freedom to explore at the edge of computing. Please join us in congratulating this year's Fellows. To find out more, please visit <http://research.microsoft.com/nff>

Microsoft
Research



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

Associate Editors in Chief
Bill N. Schilit
Kathleen Swigger
University of North Texas

Computing Practices
Rohit Kapur
rohit.kapur@synopsys.com

Perspectives
Bob Colwell
bob.colwell@comcast.net

Research Features
Kathleen Swigger
kathy@cs.unt.edu

Special Issues
Bill N. Schilit
schilit@computer.org

Web Editor
Ron Vetter
vetterr@uncw.edu

2007 IEEE Computer Society President
Michael R. Williams
president@computer.org

Area Editors
Computer Architectures
Steven K. Reinhardt
Reservoir Labs Inc.
Databases/Software
Michael R. Blaha
Modelsoft Consulting Corporation
Graphics and Multimedia
Oliver Bimber
Bauhaus University Weimar
Information and Data Management
Naren Ramakrishnan
Virginia Tech
Multimedia
Savitha Srinivasan
IBM Almaden Research Center
Networking
Jonathan Liu
University of Florida
Software
Dan Cooke
Texas Tech University
Robert B. France
Colorado State University
H. Dieter Rombach
AG Software Engineering

Column Editors
Broadening Participation in Computing
Juan E. Gilbert
Embedded Computing
Wayne Wolf
Princeton University
Entertainment Computing
Michael R. Macedonia
Michael C. van Lent
How Things Work
Alf Weaver
University of Virginia
In Our Time
David A. Grier
George Washington University
IT Systems Perspectives
Richard G. Mathieu
James Madison University
Invisible Computing
Bill N. Schilit
The Profession
Neville Holmes
University of Tasmania

Security
Jack Cole
US Army Research Laboratory
Software Technologies
Mike Hinchey
Loyola College Maryland
Standards
John Harauz
Jonic Systems Engineering Inc.
Web Technologies
Simon S.Y. Shim
SAP Labs
Advisory Panel
James H. Aylor
University of Virginia
Thomas Cain
University of Pittsburgh
Doris L. Carver
Louisiana State University
Ralph Cavin
Semiconductor Research Corp.
Ron Hoelzeman
University of Pittsburgh

Mike Lutz
Rochester Institute of Technology
Edward A. Parrish
Worcester Polytechnic Institute
Ron Vetter
University of North Carolina at Wilmington
Alf Weaver
University of Virginia

CS Publications Board
Jon Rokne (chair), Mike Blaha, Angela Burgess, Doris Carver, Mark Christensen, David Ebert, Frank Ferrante, Phil Laplante, Dick Price, Don Shafer, Linda Shafer, Steve Tanimoto, Wenping Wang

CS Magazine Operations Committee
Robert E. Filman (chair), David Albonesi, Jean Bacon, Arnold (Jay) Bragg, Carl Chang, Kwang-Ting (Tim) Cheng, Norman Chonacky, Fred Douglass, Hakan Erdogmus, David A. Grier, James Hendler, Carl Landwehr, Sethuraman (Panch) Panchanathan, Maureen Stone, Roy Want

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

Lee Garber
Senior News Editor
Margo McCall
Associate Editor
Yu-Tzu Tsai
Assistant Editor
Bob Ward
Membership News Editor
Bryan Sallis
Publication Coordinator

Design and Production
Larry Bauer
Cover art
Dirk Hagner

Administrative Staff
Publisher
Angela Burgess
aburgess@computer.org
Associate Publisher
Dick Price
Membership & Circulation Marketing Manager
Georgann Carter
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, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903. 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

June 2007, Volume 40, Number 6

IEEE Computer Society: <http://computer.org>

Computer: <http://computer.org/computer>

computer@computer.org

IEEE Computer Society Publications Office: +1 714 821 8380

PERSPECTIVES

19 The Case for Flexible NIST Security Standards

Feisal Keblawi and Dick Sullivan

A public dialogue on the role and reach of NIST as a rule maker and as a standards writer for federal cybersecurity is essential to understanding the limits of security rule making in the present federal environment.

COMPUTING PRACTICES

27 Best Practices for Automated Traceability

Jane Cleland-Huang, Raffaella Settini, Eli Romanova, Brian Berenbach, and Stephen Clark

Automated traceability applies information-retrieval techniques to generate candidate links, sharply reducing the effort of manual approaches to build and maintain a requirements trace matrix as well as providing after-the-fact traceability in legacy documents.

COVER FEATURES

37 An Open Source Environment for Cell Broadband Engine System Software

Michael Gschwind, David Erb, Sid Manning, and Mark Nutter

The Cell Broadband Engine provides the first implementation of a chip-multiprocessor with a significant number of general-purpose programmable cores targeting a broad set of workloads.

49 Benefits from Isolation in Commodity Multicore Processors

Nidhi Aggarwal, Parthasarathy Ranganathan, Norman P. Jouppi, and James E. Smith

The integration of components on a multicore chip should also be accompanied by features that help isolate effects of faults, destructive performance interference, and security breaches.

RESEARCH FEATURES

60 iMouse: An Integrated Mobile Surveillance and Wireless Sensor System

Yu-Chee Tseng, You-Chiun Wang, Kai-Yang Cheng, and Yao-Yu Hsieh

The integrated mobile surveillance and wireless sensor system (iMouse) uses static and mobile wireless sensors to detect and then analyze unusual events in the environment.

68 Password-Based Authentication: Preventing Dictionary Attacks

Saikat Chakrabarti and Mukesh Singhal

Researchers have engineered several protocols to prevent attacks, but we still need formal models to analyze and aid in the effective design of acceptable password protocols geared to prevent dictionary attacks.

Cover design and artwork by Dirk Hagner

ABOUT THIS ISSUE

Technology scaling and power trends have led to the widespread emergence of multicore processors as the predominant hardware paradigm. Multiple cores are being integrated on a single chip and made available for general-purpose computing. In this issue, we look at an implementation of a chip-multiprocessor with a significant number of general-purpose programmable cores targeting a broad set of workloads. Another article reviews several multicore architectures designed for fault isolation. Other topics include automated traceability, NIST security standards, the iMouse, and password-based authentication.

Flagship Publication of the IEEE Computer Society

CELEBRATING THE PAST

- 6 In Our Time
The Camino Real
David Alan Grier
- 9 32 & 16 Years Ago
Computer, June 1975 and 1991
Neville Holmes

NEWS

- 12 Technology News
Searching the Visual Web
Sixto Ortiz Jr.
- 15 News Briefs
Linda Dailey Paulson and George Lawton

MEMBERSHIP NEWS

- 81 Call and Calendar
- 84 IEEE Computer Society Connection

COLUMNS

- 90 How Things Work
Binary Arithmetic
Neville Holmes
- 94 Entertainment Computing
iPhones Target the Tech Elite
Michael Macedonia
- 96 Invisible Computing
Social Scripting for the Web
Tessa Lau
- 100 Software Technologies
An Era of Change-Tolerant Systems
Shawn Bohner
- 103 Security
Discription: Internal Hard-Disk Encryption for Secure Storage
Laszlo Hars
- 108 The Profession
Supporting Resource-Constrained Collaboration Environments
Sean M. Price

DEPARTMENTS

- 4 Article Summaries
- 11 Computer Society Information
- 76 Career Opportunities
- 80 Advertiser/Product Index
- 86 IEEE Computer Society Membership Application
- 89 Bookshelf



NEXT MONTH:
3D
Visualization



BPA WORLDWIDE
COPYRIGHT © 2007 BY THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS INC. ALL RIGHTS RESERVED.
LIBRARIES ARE PERMITTED TO PHOTOCOPY BEYOND THE LIMITS OF US COPYRIGHT LAW FOR PRIVATE USE OF PATRONS: (1) THOSE POST-1977 ARTICLES THAT CARRY A CODE AT THE BOTTOM OF THE FIRST PAGE, PROVIDED THE PER-COPY FEE INDICATED IN THE CODE IS PAID THROUGH THE COPYRIGHT CLEARANCE CENTER, 222 ROSEWOOD DR., DANVERS, MA 01923; (2) PRE-1978 ARTICLES WITHOUT FEE. FOR OTHER COPYING, REPRINT, OR REPUBLICATION PERMISSION, WRITE TO COPYRIGHTS AND PERMISSIONS DEPARTMENT, IEEE PUBLICATIONS ADMINISTRATION, 445 HOES LANE, P.O. BOX 1331, PISCATAWAY, NJ 08855-1331.

ARTICLE SUMMARIES

The Case for Flexible NIST Security Standards

pp. 19-26

Feisal Keblawi and Dick Sullivan

Recently, the US National Institute of Standards and Technology (NIST) began issuing new kinds of information system security (ISS) standards. Responding to the 2002 Federal Information Security Management Act (FISMA), these mandatory standards regulate ISS processes in federal civilian agencies and require standardized security controls in all related federal information systems.

Experience shows that federal standards aligned with established commercial practices generally succeed. The authors seek to initiate a dialogue on the role and reach of NIST as a rule maker and as a standards writer for federal cybersecurity that will result in a new understanding about the limits of security rule making in the present federal environment.

Best Practices for Automated Traceability

pp. 27-35

Jane Cleland-Huang, Raffaella Settini, Eli Romanova, Brian Berenbach, and Stephen Clark

Traceability helps determine that researchers have refined requirements into lower-level design components, built them into the executable system, and tested them effectively. It further helps analysts understand the implications of a proposed change and ensures that no extraneous code exists.

Unfortunately, many organizations fail to implement effective traceability. Because manual traces are often created ad hoc, they tend to be inconsistent and often incomplete. Automated traceability methods aggressively tackle these problems by decreasing the effort needed to construct and maintain a set of traceability links and by providing traceability across a much broader set of documents.

An Open Source Environment for Cell Broadband Engine System Software

pp. 37-47

Michael Gschwind, David Erb, Sid Manning, and Mark Nutter

New computer architectures usually arise in response to tectonic shifts in technology and market conditions. As the era of pure CMOS frequency scaling ends, architects must again respond to massive technological changes by more efficiently exploiting density scaling.

The Cell Broadband Engine provides the first implementation of a chip-multiprocessor with a significant number of general-purpose programmable cores targeting a broad set of workloads, including intensive multimedia and scientific processing.

Isolation in Commodity Multicore Processors

pp. 49-59

Nidhi Aggarwal, Parthasarathy Ranganathan, Norman P. Jouppi, and James E. Smith

Technology scaling and power trends have led to the widespread emergence of chip multiprocessors as the predominant hardware paradigm. From a system viewpoint, CMPs provide higher levels of integration, typically including multiple processing cores, caches, memory controllers, and even some I/O processing—all in a single socket.

Multiple cores will provide unprecedented compute power on a single chip. However, integration of several components on a chip must be accompanied by features that enable isolation from fault effects, destructive performance interference, and security breaches.

iMouse: An Integrated Mobile Surveillance and Wireless Sensor System

pp. 60-67

Yu-Chee Tseng, You-Chiun Wang, Kai-Yang Cheng, and Yao-Yu Hsieh

With their environment-sensing capability, wireless sensor networks can enrich human life in applications such as healthcare, building monitoring, and home security.

The iMouse system integrates WSN technologies into surveillance technologies to support intelligent mobile surveillance services. The authors suggest several ways to improve or extend iMouse. One option is to facilitate mobile sensor navigation by, for example, integrating localization schemes to guide mobile sensors instead of using color tapes. A second option is to exploit coordination among mobile sensors, especially when they're on the road.

Password-Based Authentication: Preventing Dictionary Attacks

pp. 68-74

Saikat Chakrabarti and Mukesh Singhal

The most common verification technique is to check whether the claimant possesses information or characteristics that a genuine entity should possess. For example, we can authenticate a phone call by recognizing a person's voice and identify people we know by recognizing their appearance.

But the authentication process can get complicated when visual or auditory clues aren't available to help with identification.

Because they're cheap and convenient, passwords have become the most popular technique for authenticating users trying to access confidential data stored in computers, even though such authentication is vulnerable to several forms of attack. Password protocols preventing offline dictionary attacks need more than heuristic arguments to provide a guarantee of security.

HOT

CHIPS

19

ADVANCE PROGRAM

HOT CHIPS 19

A Symposium on High-Performance Chips
August 19-21, 2007, Memorial Auditorium,
Stanford University, Palo Alto, California

HOT CHIPS brings together designers and architects of high-performance chips, software, and systems. Presentations focus on up-to-the-minute real developments. This symposium is the primary forum for engineers and researchers to highlight their leading-edge designs. Three full days of tutorials and technical sessions will keep you on top of the industry.

Sunday August 19	Morning Tutorial Ralph Wittig Peter Alfke	Approaches to System Design for the Working Engineer Xilinx David Witt Texas Instruments Xilinx Shephard Siegel Mercury Computer Systems	Organizing Committee Chair John Sell Microsoft Vice Chair Don Draper Rambus Finance Lily Jow HP Publicity Kevin Krewell NVIDIA Gail Sachs Telairity Advertising Don Draper Rambus Fely Krewell Spansion Sponsorship Amr Zaky Broadcom Publications Gordon Garb Sun Micro Registration Ravi Rajamani Oracle Sujata Ramasubramanian Intel Local Arrangements Lance Hammond Apple Webmaster Alexis Cordova CTO Yusuf Abdulghani Apple
	Afternoon Tutorial Norm Jouppi HP Labs	Enterprise Power and Cooling: A Chip-to-Data Center Perspective Chandrakant Patel HP Labs Parthasarathy Ranganathan HP Labs	
Monday August 20	IBM Power6 • Fault-Tolerant Design of the IBM POWER6™ Microprocessor • System Performance Scaling of IBM POWER6™ Based Servers • The 3rd Generation of IBM's Elastic Interface (EI-3) Implementation on POWER6™	IBM	Steering Committee Don Alpert Camelback Arch. Lily Jow HP Allen Baum Intel Pradeep Dubey Intel John Mashey Techviser Howard Sachs Telairity Alan Jay Smith UC Berkeley
	Keynote I: Vernor Vinge Computer scientist, science-fiction writer, author of <i>True Names</i> and <i>Rainbows End</i>		
	Multi-Core and Parallelism A • NVIDIA GeForce 8800™ GPU • NVIDIA GPU Parallel Computing Architecture • Performance of Non-Graphics Applications on the GeForce 8800™	NVIDIA NVIDIA UIUC	Program Committee Program Co-Chairs John Mashey Techviser Rajeevan Amirtharajah UC Davis Program Committee Forrest Baskett NEA Ralph Wittig Xilinx John Montrym NVIDIA Christos Kozyrakis Stanford U Chuck Moore AMD Mitsuo Saito Toshiba Alan Jay Smith UC Berkeley Marc Tremblay Sun Micro Jan-Willem van de Waerd NXP Semiconductors Doug Burger UT Austin Norm Jouppi HP Labs Dileep Bhandarkar Microsoft
	Multi-Core and Parallelism B • Radeon R600, a 2nd Generation Unified Shader Architecture • Teraflop Prototype Processor with 80 Cores • Design and Implementation of the TRIPS Prototype Chip • Tile Processor: Embedded Multicore for Networking Multimedia	AMD Intel UT Austin Tilera	
	Embedded and Video • SH-X3: SuperH Multi-Core for Embedded Systems • An HD Image Processor for Low-Cost Entertainment Products • A Professional H.264/AVC CODEC Chip-Set for HDTV Broadcast	Renesas TI NTT	Founder Bob Stewart SRE
	Panel: CMOS is Dead ... Long Live CMOS Moderator: Norm Jouppi	HP Labs	
Tuesday August 21	Technology and Software Directions • Multi-terabit Switch Fabrics Enabled by Proximity Communication • Thyristor RAM: A High-Speed High-Density Embedded Memory • Raksha: A Flexible Architecture for Software Security	Sun Micro T-RAM Semi Stanford U	
	Wireless • A 4Gbps Wireless Uncompressed 1080p 60 GHz HD Transceiver • A 2x2 MIMO Baseband for Wireless Local-Area Network (802.11n)	SiBeam Broadcom	
	Keynote II: Multicore and Beyond: Evolving the X86 Architecture Phil Hester CTO, AMD		
	Networking • A Packet Processing Chipset • Chesapeake: A 50Gbps Network Processor and Traffic Manager • A System-on-a-Chip with Integrated Accelerators • Focalpoint II, A Low-Latency, High Bandwidth Switch/Router Chip	Cisco Bay Micro Intel Fulcrum Micro	
	Mobile PC Processors and Chipsets • Power Management Features in Penryn 45nm Core2™ Duo • Next Generation Mobile X86 Processor • nForce 680i and 680 Platform Processors	Intel AMD NVIDIA	
	Big Iron • Victoria Falls - Scaling Highly-Threaded Processor Cores • The Next-Generation Mainframe Microprocessor	Sun Micro IBM	

IEEE
computer
society

This is a preliminary program; changes may occur. For the most up-to-the-minute details on presentations and schedules, and for registration information, please visit our web site where you can also check out HOT Interconnects (another HOT Symposium being held following HOT CHIPS):
Web: <http://www.hotchips.org> Email: info2007@hotchips.org

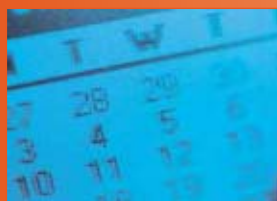
1997 2007
SSCS
10th Anniversary

 IEEE

A Symposium of the Technical Committee on Microprocessors and Microcomputers
of the IEEE Computer Society and the Solid State Circuits Society

The Camino Real

David Alan Grier
George Washington University



Teaching a computer science course to a group of Peruvian engineers provides valuable insights into the development of computing technology in Latin America.

I have never felt so far from civilization. The plane ride had been long enough, a restless all-night flight from Miami to Lima, Peru, but it had only taken us part of the way. The next segment of the trip was a bruising 14-hour drive over the 14,000-foot Ticlio Pass. Once we arrived at our destination, we faced one final journey: the hike into the dark tunnels of the lead-zinc mine.

The trip was part of a consulting job, an assignment that had seemed intriguing when it was offered. The mine's owner, a friend of a student, wanted me to come to Peru and teach a computer science course to his engineering staff. In particular, he wanted to develop software to prepare the daily production plans for the mine.

AN UNDERGROUND LAB

When I accepted the job, I joked that all I knew about Peru was a story that it had been considered the site of El D'Orado, the fabled city of gold that had to surrender its wealth to Spanish invaders. As I walked into the mine, I received the first hints that cities of gold were not places of leisure but civilizations built with hard labor. The mine itself was Miltonic, the "dungeon horrible" of *Paradise Lost*,

"As far removed from God and light of Heaven as from the centre thrice to th' utmost pole."

My headlamp peered weakly into the fog of dust and diesel fumes that filled the air. I heard the crash of ore in the distance as front loaders dropped ore down deep shafts to rail cars that waited on levels below us. I had been told to keep my eye on the tunnel roof, as pieces could free themselves from the earth's grip and fall on the heads of unsuspecting miners. I was so anxious about this charge that I failed to watch my steps and regularly stumbled over rubble on the mine floor.

After a half hour of walking through the darkness, we came across a culvert, a pipe 30 or 35 inches in diameter. My guide, a tall, slender engineer named Tito after the Yugoslav dictator, indicated that I should get on my knees and begin to crawl. "This is an unstable area," he said, "so we put in the pipe."

As I moved through the tube, I found that the culvert turned upward and that I had to press my back against the top of the pipe to keep from slipping. After proceeding in this manner for 75 or 80 feet, I emerged into a little chamber with a door. Behind the door—in the heart of the mine—was the engineering shop. It

was a cavern sheathed in corrugated iron, lit with fluorescent lights, and filled with machine tools.

"This is where we run the mine," said Tito as he advised me to turn off my headlamp. "We want to put a computer station here and connect it to the mine office."

As I looked around the room, I saw shelves of drawings and chests of tools. I heard the dripping of water on the iron ceiling and the whir of an electric fan. "So this is technology transfer," I thought. "This is what it means to be at the frontier."

COMPUTING TECHNOLOGY IN SOUTH AMERICA

The countries of South America were late recipients of computing technology. The most industrialized South American countries—Argentina, Brazil, and Chile—purchased their first computers in 1962 and 1963. The number grew steadily but slowly. When a researcher from MIT conducted a survey in 1972, he identified fewer than 2,800 computers on the entire continent. Of these, 75 percent were located in Brazil and Argentina. Peru had only 100 machines, less than 4 percent of the total. "Most of the installations in Latin America are running conventional data-processing applications," the researcher noted, "deviating very little from the North American pattern."

At the time, Peru had a simple economy. The country's largest businesses were engaged in mining and agriculture. In contrast, both Argentina and Brazil had growing industries. In Buenos Aires, Siam Di Tella was manufacturing automobiles. The Brazilian government had just moved part of its aircraft industry to a private firm, Empresa Brasileira de Aeronáutica S.A. (now known as Embraer). Yet, neither of these countries was ready to start a computer firm. "No computers, aside from tabulating equipment, are manufactured in [the region]," wrote a visiting professor. "All must be imported."

Through the 1970s and early 1980s, observers of the region noted many factors that were limiting the growth of a high-technology industry. A report by

the National Research Council pointed to “persistent internal economic and social problems,” which included such things as “obsolete land ownership systems, a markedly unequal income distribution, inequities of ethnic integration, segmentation of production and marketing systems, inequities for various socioeconomic groups in education and training, and restricted social mobility.”

By the mid-1980s, most of these observers noticed a change in the region. “A visit to Latin America has always seemed like a journey into another era,” wrote a visiting scholar in 1985. “Whereas it used to be that there were 30, 40, even 50 years of difference, no more than a decade separates the US and Latin America when it comes to computers.” To clarify his point, he added, “Computer use is clearly on the rise in Latin America and at an accelerated pace.”

SETTING UP SHOP

The new computer for the mine’s machine shop was a conventional desktop system. It had accompanied us on the truck ride over the mountain pass, carefully snuggled beneath the truck seats. Noting that it was one of the more powerful machines then available, I asked Tito where he had purchased it. I knew that the Peruvian government had severely limited all imports. Tito smiled. “Puerto del Este,” he said. “Paraguay.”

If you believed the economic statistics, Puerto del Este was the high-tech capital of the region, the largest importer of high-tech goods in South America. However, the city was only a transshipment point. Goods arrived on one flight only to leave on the next. The merchants of Puerto del Este were smugglers. They had learned how to circumvent the tariffs of Brazil, Argentina, Chile, and Peru.

We carried the new computer to the machine shop and installed it in a plastic box to give it some kind of protection from the water. We snaked the network cable out through a ventilation shaft. It was a longer path than the route through the culvert, but it

IEEE Annals of the History of Computing

The *IEEE Annals of the History of Computing*, the IEEE Computer Society’s historical magazine, publishes articles on the development of computing technology and computing institutions. *Annals* has published stories of traditional computing in native South American culture, such as “The Logical-Numerical System of Inca Quipus” by Marcia Ascher in the July 1983 issue and the introduction of the electronic computer into the region, including Jim Cortrada’s January 2004 article, “How Did Computing Go Global?”

was less likely to be broken by shifting rocks. The total distance between the computer in the mine and its host in the engineers’ office was slightly longer than the maximum length specified by the machine manufacturer, but it didn’t seem to matter. The two machines were able to communicate with only occasional difficulty.

With this network, Tito and his colleagues wanted to plan the day’s production. For each day, they wanted to determine which sections of the mountain should be blasted, which tunnels should be reinforced, which piles of rock should be moved to the rail cars, which routes the miners should take. Such plans could be created with an hour of work, but—like the mountain itself—they were never stable.

Events that occurred over the course of a day could invalidate the plans. A pile of ore might prove to contain too little or too much zinc. A front loader might fail. Two or three times a day, the engineers would have to redraft the day’s plans while mining equipment stood idle and other problems begged for attention.

Tito wanted to eliminate such delays with the computer we had installed in the subterranean machine shop. He wanted a program that could create revised plans without the intervention of the engineering staff.

CONDUCTING THE PROGRAMMING COURSE

Tito and I required a couple of days to write the planning program. It used a simple search algorithm. This program used a database that described the state of the mine: the quantity and quality of ore needed, the tunnels that were

open, the piles of ore available, the areas ready to be blasted, and the equipment that was operational. The program would then search through the various possible plans. It was not a very efficient program, but it didn’t need to be efficient as it required no intervention by the engineers. The system generally could create a new plan in 6 to 8 minutes.

After we got the system running, I had to explain how the programs worked and how the engineers could modify it. We held a brief programming course in the engineers’ office, a large plywood building with windows that opened to long vistas of the high jungle. We created a classroom space by pushing tables to one side, moving a small computer to a desk, and taping paper to the wall. I stood at the front of the room with a list of information that I wanted to convey: the database’s structure, the search algorithm’s nature, and the programming language’s syntax.

We had planned two hours for the class, but two hours stretched to three, three to four, and four to five. Part of the added time came from the need to translate the ideas for those who could not follow my English. The remainder came from the desire to test all the ideas as soon as I presented them.

As I began to describe the first idea, one of the engineers walked to the computer, took a seat, and began to program. Even if I had tried, I doubt that I could have stopped him. The others gathered around the screen and added their comments, criticized the results, and joked about any mistakes.

We ended the class at the sixth hour, when the declining sun made it difficult

IN OUR TIME

to see the notes that I had written on the wall. We finished the day tired but happy with what we had accomplished.

A TRIP BACK IN TIME

We had dinner in town that night, a village far away from the modern world. Women in bowler hats squatted in front of the church and sold roasted meats: chicken, pork, and guinea pig. Men watched a cockfight in an alley. People stood in a queue at the post office waiting to use the public radio phone, the only way to communicate with Lima and the outside world.

We took our meal at the town's most prominent restaurant, a building that was little better than the engineers' office. The floor was concrete, the sides were open to the weather, and the table was nothing more than a strip of plywood sitting on galvanized pipe. In our honor, the owner entertained us with his favorite American music, a worn tape of Jimmy Rogers tunes. "My pocket-book is empty and my heart is filled with pain," sang the scratchy voice. "I'm a thousand miles away from home just waiting for a train."

We talked about many things that night, but the conversation often turned to the isolation these men and women felt. They complained about how far they were from Lima and from the centers of computer science. They said that it was very hard to get technical information and that they had to spend hours trying to glean new ideas from the few manuals and books that they had been able to procure. They were pleased that I had been able to visit them, that I had brought them a stack of computer books, and that I had spent the time to give them a few insights into the current state of computer development.

A NEW CAMINO REAL

Only rarely can we recognize that we are standing at a point of inflection, a time when the twig is bent and a tree begins to follow a new curve. I left the mine honestly believing that I could return in two or four or eight years and find that nothing had

changed. If anything, I believed that the pull backwards was stronger than the push forward, that the efforts of terrorists to destroy "bourgeois civilization" were more likely to succeed than the work of the computer industry to expand the benefits of a cyber-infrastructure.

For a time, the network was our royal road that carried information from an engineering shack in the bowels of the Andes to offices in the industrialized world.

Digital networks started to reach South America that summer. Bitnet, the network based on IBM protocols, had reached Argentina just before I had arrived in Peru. "The advances of knowledge, science, and the arts know no boundaries, because they belong to all of humanity," raved Argentine President Raul Alfonsine. "When any nation, whatever its motivation, inhibits free access to information from within or outside itself, it is doubtless condemning itself to isolation and underdevelopment."

A network connection reached Peru six or eight months after my visit. For a time, the network was our Camino Real, our royal road that carried information from an engineering shack in the bowels of the Andes to offices in the industrialized world. Tito posed technical questions from his machine in the high mountain valleys, and I replied from my computer in Washington, D.C. The mine's general manager sent production reports to the Lima office, and the Lima office responded with directives. When they got access to the computer, the mine's employees mailed electronic letters to their families in Lima, and the families returned news of the outside world.

After a year or two, the mining engineers began to realize that they no longer needed to travel the difficult

path up the mountain but could use the network connection to operate the mine. They could review the daily production plans in the relative comfort of Lima and evaluate the day's accomplishments without being anywhere near the cramped and dirty shop in the mine. One by one, the engineers relocated to Lima. They visited the mine only when they needed to inspect the operations. Tito was the first to return, but he was soon followed by the others.

Unlike the original Camino Real, the computer network was not a simple path in the mountains, a single road from the Peruvian capital to a distant source of wealth. It put Tito and the other mining engineers in touch with the world's industries, industries that offered special opportunities to Spanish-speaking engineers. Just as they had taken the pathway from the mine to the capital, Tito and his friends traveled the computer network to new jobs with software firms in El Norte.

Tito was again the leader. He found a software support job with Siemens AG. From his home in Lima, he worked with clients across South America. The other mining engineers soon followed the same path. One went to IBM, another to Novell, a third found a career with Computer Associates, and a fourth became a software retailer. Once they had lived in an isolated world of dust and grime, of tiny villages and expansive forests, until their machine shop, deep beneath the Andes Mountains, had become a terminus on the digital Camino Real. ■

David Alan Grier is the editor in chief, IEEE Annals of the History of Computing, and the author of When Computers Were Human (Princeton University Press, 2005). Grier is an associate professor in the Center for International Science and Technology Policy at the George Washington University. Contact him at grier@gwu.edu.

1975 • 1991 • 1975 • 1991 • 1975 • 1991 • 1975 • 1991

32 & 16 YEARS AGO

JUNE 1975

SOCIO-ECONOMIC ISSUES (p. 9). “In a challenging keynote address, Jay W. Forrester called upon computer people to direct their problem-solving capabilities beyond the industry toward the larger socio-economic issues facing the country today. Dr. Forrester, Germeshausen Professor at MIT’s Alfred P. Sloan School of Management and a world-renowned authority in the field of computer science and the dynamics of change, spoke at the opening session Monday, May 19, of the National Computer Conference in Anaheim, California.”

“We are now paying the price for utilizing short-run advantages over the last 100 to 200 years, Forrester said. Since we have no methodology for understanding the long term, we’ve indulged in ‘quick fixes’—for example, in handling the energy shortage. In trying to solve problems without understanding them, the very actions taken frequently prove to be essential to the continued development of these problems.

“Computer people are uniquely qualified to analyze the nature of our socio-economic system because of a background of capability that is now becoming available. The confluence of three major streams—practical management and politics, feedback control concepts, and electronic digital computers—[has] opened up possibilities in systems dynamics, which is Forrester’s field.”

STRUCTURED PROGRAMMING (p. 21). “The dramatic decline in computer hardware costs in recent years has brought the costs of software development into sharp focus. But important efforts to address software technology have been underway during the same period—notably in the 60’s by Böhm, Jacopini, Dijkstra, Parnas, and others in the university environment, and in the 70’s by such people as Mills and Baker, who defined specific elements of a methodology directed toward reducing software costs and improving software quality. This methodology, which according to Mills and Baker was successfully applied to the *New York Times* information bank project, is generally referred to as *structured programming* (SP)—but includes such methods as top down design and chief programmer teams.

“Subsequently—and inevitably—all this attention prompted questions as to whether SP is in fact feasible and economically practical on an industrywide basis, and if so to what extent.”

MULTIPROCESSING (p. 80). “A multiprocessing system capability designed to increase throughput of Xerox Sigma 9 computers by up to 300 percent for a fully expanded system has been announced by the Data Systems Division of Xerox Corporation.

“From two to four of the Sigma 9 CPUs can be tightly coupled under the new system, which runs under the Xerox Control Program-Five (CP-V) operating system.”

“One CPU is designated the primary processor and handles input/output operations, schedules and executes user programs, and provides services to all user requests. All other CPUs in the system are called secondary processors and function as computing peripherals, executing assigned user program tasks. In the event of a malfunction of the primary processor, a fail-over system allows any one of the secondary processors to take over its tasks.”

“Prices for Xerox Sigma 9 multiprocessing systems, including two, three, or four CPUs, range from \$1 to \$3 million, with delivery in the fourth quarter of this year.”

LEGISLATION PROCESSING (p. 86). “The State of California will utilize a Varian minicomputer to draft new legislation and maintain state codes. The computer will be installed in the offices of the Legislative Council, an arm of the California State Legislature that performs legal research and assists in developing preliminary versions of new laws.

“The function of the computer will be to expand the capabilities of the overall data processing center by allowing the remote Varian to relieve Systems/370 of processing tasks. By permitting use of non-370 compatible terminals, utilization can be made of the special characteristics of certain terminals.”

A SOLAR HOME (p. 87). “Plans to custom design and build a private residence utilizing an advanced type solar energy system and a ‘comfort computer’ have been announced by Stanmar, Inc. of Sudbury, Mass.

“The home, built in a community of the buyer’s preference in Greater Boston or the outlying suburbs, will have its domestic tap water heated and its living space heated and cooled with an electronically controlled, solar augmented comfort system. The solar system to be used in the home is being developed by Raytheon Company of Lexington, Mass.”

CHARGE CARD SYSTEM (p. 87). “Continental Bank now offers Chicago-area retailers a fully-integrated computer system for approving charge card purchases and checks.

“The system—which operates with Master Charge or a retailer’s check cashing card, or could be used with a retailer’s own charge card—enables merchants to electronically verify a card’s validity and credit limit through in-store terminals connected to host computers at Continental. It rejects revoked Master Charge and private label check-cashing and charge cards, which have exceeded their maximum credit limits, or are revoked or restrained for other reasons. As an added feature, the terminals complete charge card sales slips.”

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

32 & 16 YEARS AGO

JUNE 1991

DOCUMENT PREPARATION (pp. 7-8). "Lilac is an experimental document preparation system designed to provide the best of both the WYSIWYG and the document compiler worlds. It does this by offering both WYSIWYG editing and language-based document description as two views side by side on the screen... The page view is a WYSIWYG editor showing a close approximation to the printed output. The source view shows a program-like description of the document in a special-purpose language. This language supports subroutines, variables, and conditional execution and is designed to encourage the use of subroutines to embody structure. Both views are editable, but Lilac is designed with the expectation that most editing will occur in the page view."

DISCRETE-EVENT SIMULATION (p. 33). "Asynchronous distributed discrete-event simulation of cyclic circuits has the potential to address problems in digital hardware design, queuing networks, and banking transactions. Until now, no reported algorithm offered freedom from deadlock and acceptable performance. The Yaddes algorithm, on the other hand, is mathematically correct and free from deadlock."

"The Yaddes approach opens the possibility of modeling as discrete-event systems challenging problems from such disciplines as banking, railway and mobile phone networks, sociological interactions, human decision-making processes, aircraft simulation, and weather forecasting."

LITERATE PROGRAMMING (p. 60). "The literate paradigm offers a platform for code generation that promotes readability and understandability. As it exists today, this platform is ill-suited for adoption by most software developers. The lack of a proper interface into the software development life cycle, the lack of an environment that minimizes the overhead associated with the literate paradigm, and the awkwardness of current literate languages discourage the adoption of literate techniques. Furthermore, to offer a practical, usable methodology for software development, the literate paradigm must encompass the entire life cycle, not just the implementation phase."

DATABASE ATOMICITY (p. 63). "Database performance depends on both external and internal functionality. External functionality encompasses such issues as data models and query languages. Internal functionality involves file structure, indexing, and query optimization. ... A third area, often taken for granted, is crucial in a user's interaction with the database. This area involves the transaction constraints imposed to achieve *recovery atomicity* and *concurrency atomicity*.

"In a database with recovery atomicity, when a user initiates a transaction, the transaction either executes in

its entirety or has no effect whatsoever on the database. Thus, even if the underlying hardware or software fails, application programs do not corrupt the database with incorrect results. The concurrency property, on the other hand, assures users that concurrent execution of another transaction does not affect their own applications. To achieve concurrency atomicity, a transaction processing system controls execution of transactions so that their interleaved executions are equivalent (in their effect) to some serial execution."

VIRTUAL MACHINES (p. 81). "Parsytec Inc. introduces the Multiple Virtual Machine Architecture as the basis for its transputer-based multicluster and supercluster series of parallel processing machines. The architecture's open, modular design lets users configure processing power and I/O capabilities for any application."

"Multiple virtual machines run under Helios, the Multitool transputer development system, and cross compilers using MS-DOS, the Apple operating system, and Sun Unix. Language support includes C, Par.C, Fortran, Pascal, Occam, and Ada."

A HEADY COMPUTER (p. 82). "Park Engineering has developed Compcap, a 1-lb. computer you can wear on your head. This portable can assist workers in the field who need high mobility."

"The PC-compatible Compcap, which reputedly offers the speed and processing power of a 386 desktop, comes in two models: a hard hat that incorporates the electronics, and a soft band worn around the head or hat with electronics built into a belt or vest. The device offers 4 Mbytes of RAM with memory-card interface, custom keypad interface, and barcode reader options."

MASSIVE PARALLELISM (p. 97). "Projecting sustained processing speed about 15 years ahead, [Steve] Nelson [Cray's vice president for technology] showed it increasing from a few gigaflops at present to about one teraflops. At the same time, the cost may drop to a few hundred dollars per megaflops (from around \$10,000). He used a chart to show a mid 90s 'knee,' with the rate of change increasing sharply."

"A lot of people know that, when we can field the massively parallel designs, we will have the opportunity to make a quantum jump in performance on those problems that can be mapped in massively parallel machines," Nelson said.

"The best way to exploit this technology, he thinks, is to hook a massively parallel system to a conventional supercomputer, like the Y-MP series."

PDFs of the articles and departments from Computer's June 1991 issue are available at www.computer.org/computer.

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 14 magazines, 9 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 over 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 and certification.

Next Board Meeting: 9 Nov. 2007, Cancún, Mexico



EXECUTIVE COMMITTEE

President: Michael R. Williams*

President-Elect: Rangachar Kasturi*

Past President: Deborah M. Cooper*

VP, Conferences and Tutorials: Susan K. (Kathy) Land (1ST VP)*

VP, Electronic Products and Services: Sorel Reisman (2ND VP)*

VP, Chapters Activities: Antonio Doria*

VP, Educational Activities: Stephen B. Seidman†

VP, Publications: Jon G. Rokne†

VP, Standards Activities: John Walz†

VP, Technical Activities: Stephanie M. White*

Secretary: Christina M. Schober*

Treasurer: Michel Israel†

2006–2007 IEEE Division V Director: Oscar N. Garcia†

2007–2008 IEEE Division VIII Director: Thomas W. Williams†

2007 IEEE Division V Director-Elect: Deborah M. Cooper*

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 2007: Jean M. Bacon, George V. Cybenko, Antonio Doria, Richard A. Kemmerer, Itaru Mimura, Brian M. O'Connell, Christina M. Schober

Term Expiring 2008: Richard H. Eckhouse, James D. Isaak, James W. Moore, Gary McGraw, Robert H. Sloan, Makoto Takizawa, Stephanie M. White

Term Expiring 2009: Van L. Eden, Robert Dupuis, Frank E. Ferrante, Roger U. Fujii, Anne Quiroz Gates, Juan E. Gilbert, Don F. Shafer

EXECUTIVE STAFF

Associate Executive Director: Anne Marie Kelly

Publisher: Angela R. Burgess

Associate Publisher: Dick J. Price

Director, Administration: Violet S. Doan

Director, Finance and Accounting: John Miller

COMPUTER SOCIETY OFFICES

Washington Office. 1730 Massachusetts Ave. NW, Washington, DC 20036-1992

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

Email: hq.ofc@computer.org

Los Alamitos Office. 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314

Phone: +1 714 821 8380

Email: help@computer.org

Membership and Publication Orders:

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

Email: help@computer.org

Asia/Pacific Office. 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: Leah H. Jamieson

President-Elect: Lewis Terman

Past President: Michael R. Lightner

Executive Director & COO: Jeffry W. Raynes

Secretary: Celia Desmond

Treasurer: David Green

VP, Educational Activities: Moshe Kam

VP, Publication Services and Products: John Baillieul

VP, Regional Activities: Pedro Ray

President, Standards Association: George W. Arnold

VP, Technical Activities: Peter Staeker

IEEE Division V Director: Oscar N. Garcia

IEEE Division VIII Director: Thomas W. Williams

President, IEEE-USA: John W. Meredith, P.E.

TECHNOLOGY NEWS

Searching the Visual Web

Sixto Ortiz Jr.

Google's recent purchase of YouTube is a strong signal that video is the next big wave of content to hit the Web. Photos and other images are also becoming much more prevalent. This trend has developed in part because so many users now have the broadband connections necessary to receive and transmit multimedia files.

The increase in online multimedia content means the types and amount of information users can find on the Internet is expanding rapidly. However, this information is largely useless unless it can be searched effectively and efficiently.

Today's image- and video-search techniques don't examine the content of a multimedia file but instead rely on metadata, captions, or other textual information—not always accurate or thorough—that content authors or providers only sometimes attach to multimedia files.

This is largely ineffective because five or six words of metadata or file information can't begin to approach the complexity of content in a video file, explained James McQuivey, a vice president with market analysis firm Forrester Research.

Researchers are thus looking for better ways to search visual content, which engines such as Google and Yahoo! don't offer currently.

There is a powerful financial incentive for developing successful image and video search engines: Industry watchers expect advertisements that appear with video-search



results to yield more revenue during the next few years.

Research firm eMarketer predicts that in the US, online video advertising will grow from \$410 million last year to \$775 million this year to \$1.3 billion in 2008, and will account for 11.5 percent of overall online advertising by 2010, compared to just 6 percent by the end of 2008, as Figure 1 shows.

Thus, companies such as AOL and Microsoft have recently either purchased small image- or video-search firms or licensed their technologies.

Nonetheless, visual search must still clear some hurdles before it is ready for prime time.

DRIVING VISUAL SEARCH

Several factors are encouraging companies to develop effective visual-search technology. For example, the Web is much livelier than it was even a few years ago because the amount of image and video content it contains has exploded.

For example, according to AccuStream iMedia Research, a market analysis firm, the number of professionally produced video streams transmitted online has grown from 1 billion in 2000 to 24 billion in

2006 and will grow to 29.8 billion this year and 48.9 billion in 2010.

Inadequacy of text-based video search

The traditional examination of metadata, other forms of text, or even file suffixes (such as .avi or .mpg) associated with video to satisfy search queries is generally ineffective because, for example, metadata tends to include only information such as a short content summary, the author, and the file's creation date, said Calafia Consulting senior partner and search-engine expert Danny Sullivan.

"This," he stated, "is a pretty poor way to describe a video."

Video is usually accompanied by very little textual information to begin with because it takes time and effort for content providers to add such material, explained search-engine expert Greg Notess, a Montana State University professor and reference librarian.

Potential advertising revenue

The chief economic force behind search technology in general is the advertising revenue it generates for engine providers.

Most major providers sell advertising to companies. Their advertisements, which include hyperlinks to other Web sites, are listed next to search results related to the product being promoted. For example, a search for information on "USB drives" will yield results accompanied by advertisements for such products.

Industry observers expect video search will become so popular that it will attract a significant amount of advertising revenue.

TAKING DIFFERENT APPROACHES

Companies are using different visual-search strategies.

Photo search

Several search engines help users find photographs.

Polar Rose. Polar Rose’s engine uses 3D modeling to perform face recognition on photos. This lets searchers look for photos of specific individuals on the Internet.

Polar Rose wants to distribute a browser plug-in to users and APIs for photo-storage sites. Users could tag any photo of a person with identifying and other information, and participating photo-storage sites would get a copy of the data. This is designed to let visitors to these sites find almost any other image of a person who has had even just one photo tagged.

Polar Rose’s technology uses a series of different-sized and diversely oriented filters to examine faces in a photograph, explained founder and chief technology officer Jan Erik Solem.

The filters detect points that are important parts of prominent facial features and then extract parameters such as eye, nose, and lip shape and cheekbone positioning. The technology generates a mathematical representation of each face, extrapolated via algorithms into three dimensions to provide more detail and accuracy, he noted. The system then stores the information in a database.

When someone wants to search for photos of the person, Polar Rose’s engine uses techniques such as data mining to look for images of faces with the same mathematical signature, Solem said.

Riya. Riya’s engine searches photos of people or objects on users’ desktops or the Web in response to textual queries. It looks for images via contextual recognition. It thus not only recognizes features via 3D modeling similar to that of Polar Rose but also looks for other clues in queries—such as clothes or jewelry a person is wearing.

Moreover, the engine can recognize text on a sign, shirt, or elsewhere in a photo via optical character recognition and thus can work with textual cues.

Vima Technologies. Vima’s technology analyzes images based on about 150 parameters and creates mathematical signatures for them,

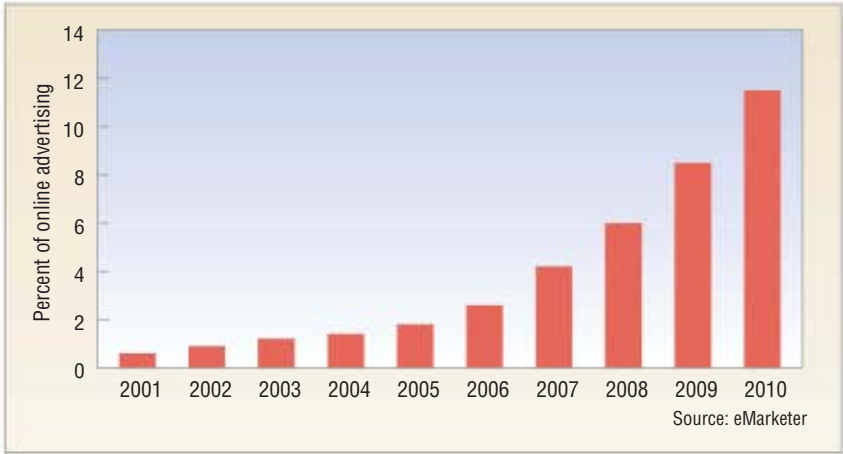


Figure 1. Research firm eMarketer predicts that in the US, online video advertising will grow steadily as a percentage of all online advertising during the next few years.

said David J. Liu, the company’s vice president of engineering

During searches, the engine uses various techniques—like data mining or the analysis of factors such as color and shape—to provide multiple possible responses when a user tries to find a specific image within a large personal collection.

As users select the possibilities that are like or unlike what they’re looking for, Vima’s engine analyzes them. It takes into account complex image features and textual content and uses learning algorithms to figure out which image the searcher is looking for within a few iterations.

Vima’s engine utilizes these same techniques to recognize and screen out pornographic images that are on users’ computers or that they are downloading from the Web.

Speech recognition in video search

Two companies are pursuing a technique that converts speech in video clips to searchable text.

Blinkx. Blinkx’s engine lets users search for specific video files from TV programs, the Web, and other sources. It uses speech recognition to transcribe video feeds into searchable text.

Blinkx, which companies such as Microsoft are working with, captures and stores on its servers video from many sources. The technology uses neural networks and hidden

Markov modeling, a probabilistic statistical approach, to analyze the audio, identify what it means, and convert the speech into searchable text. Various signal-processing techniques make recognition of speech in noisy environments more accurate.

The servers then index the analyzed files and use Blinkx’s patented Context Clustering Technology to group similar types of content together for easier searching.

The analyzed audio is associated with the corresponding parts of the video. Users can thus find what they want within the video after searching the audio transcription.

Podzinger. The Podzinger search engine is similar to Blinkx except that it works with podcasts.

Other approaches

Vendors are exploring several other visual-search techniques.

IBM’s Marvel. Marvel is a Web-based engine that searches image and video content. The technology’s multimedia-analysis engine works with MPEG-7, formally named the Multimedia Content Description Interface, which the Moving Picture Experts Group developed and the International Organization for Standardization adopted.

MPEG-7 provides a rich set of metadata tags and other elements for describing multimedia content.

TECHNOLOGY NEWS

Marvel's engine uses these descriptions in the search process.

3VR Security. The security industry extensively uses digital video recorders (DVRs) to capture images taken by surveillance cameras. However, the industry would find the video more useful if it was searchable, said Tim Ross, executive vice president and cofounder of 3VR Security, a vendor of video-surveillance-related products.

3VR's Searchable Surveillance System combines a DVR with the company's SmartSearch engine. The engine hosts face-recognition, motion-analysis, and other applications. It then uses proprietary algorithms to create metadata from input—such as video, image, and security-sensor signals—gathered from the applications.

The 3VR technology stores the metadata from activities the system detects in an event database, which investigators can search. The system associates the metadata elements with the corresponding video file so that investigators can go to the appropriate part of the video upon finding an event of interest.

Cataloging. Cataloging is the addition of enhanced metadata to a video file to, for example, insert searchable, specialized information about the content, according to Montana State's Notess.

The Dabble search engine, which collects data from video-hosting sites such as YouTube and Google, uses cataloging. Instead of capturing and storing video content, the company collects data about video files from about 160 online hosting sites, explained Dabble founder and chair Mary Hodder.

Dabble then catalogs information about the material—such as a description of its contents, its author, its theme, or its topic—as metadata. The community of users then adds details to this information.

Dabble's algorithms analyze queries and videos to determine the most relevant responses to user searches, Hodder noted. Users can then link to the recommended files.

Dabble has indexed 10.8 million videos so far and hopes to include material from additional hosting sites in the near future, she said.

STANDING IN THE WAY

Despite its great promise, visual search faces numerous obstacles to dependable effectiveness and widespread adoption. For example, using speech transcription to search for video content won't work for clips without speech.

One technical challenge is the ability to accurately interpret video or images. The hurdle is sifting through the many possible interpretations of the content they contain and inferring exactly what the user is looking for.

The main obstacle here, particularly for video, is conducting effective analysis of the huge volume of data the files contain, explained Polar Rose's Solem. Every second of video contains 20 to 25 times the data in a still image, he noted.

Also, said Forrester's McQuivey, video contains many types of information such as color, contrast, character movement, audio, camera motion, and metadata. Each aspect requires a unique type of analysis, which search engines must conduct simultaneously while examining video files, he added.

Said Notess, the large number of objects, colors, and themes in an image makes describing it difficult for searchers, even using large amounts of text.

To help them, he noted, search engines face the challenge of providing interfaces that let users specify the aspect or aspects via which they want to search files.

Generally, though, users provide relatively few parameters to narrow the search process. And this doesn't describe many images or video scenes well enough to result in an effective search.

Notess said that not many users will be willing to pay for searching video or images, so he expects that most visual search

engines will continue to be free to visitors.

Therefore, the engines' business model will depend largely on advertising. In addition, if an engine points a searcher to a commercial video or image, Notess said, the owner of the content could share the revenue with the search provider if the visitor buys the material.

Because visual search could become a big business, some industry experts predict market consolidation during the next five years, with large companies buying smaller firms.

In the future, video search could be used with other services, including those available online. For example, someone could take a photo of a speaker at a conference, submit it to an Internet service, and obtain information about the person.

According to Forrester's McQuivey, "The key to video search lies in the intelligence of the algorithms developed to either convert speech to searchable text or to deduce what type of content a video contains from visual cues."

But, he added, search providers will also have to develop a business model for aggregating large amounts of video content for many sources, to provide the number and variety of files necessary to yield meaningful query responses.

The providers must come up with a business model with financial incentives that encourage content providers, particularly those with copyrighted material, to become part of the process, he explained.

Until aggregation occurs, he said, visual searches won't be effective enough to interest people beyond specific niches. "That's why the business models are an important piece that has to be developed." ■

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

Researchers Develop Low-Cost Holographic Display

A team of MIT scientists has developed a prototype for a small, inexpensive, holographic video system that works with consumer computer hardware such as PCs or gaming consoles, thereby enabling users to view images in three dimensions.

The Mark III display could enhance participation in video games and virtual worlds, which currently are displayed mainly in two dimensions. The technology could also let doctors better view medical images such as those produced by magnetic resonance imaging. It could also help designers of complex objects such as cars.

To create a holographic video, the Mark III's software produces a 3D model of objects within a scene, explained V. Michael Bove Jr., principal research scientist and director of the MIT Media Lab's Consumer Electronics Laboratory. The software then calculates how the device must process laser beams to create a 3D hologram that looks like the model from all viewing angles. Holograms result from a diffraction pattern that occurs when light waves interfere with one another after passing through a modulator.

Based on the software calculations, the Mark III sends an electronic signal into its modulator, which then encodes a laser beam

into various intensities and frequencies. When projected onto a foggy piece of glass, the light recreates the desired 3D scene as a hologram.

The graphics processor in a user's PC, gaming console, or other device produces the signal necessary to show a series of holographic images to viewers in video form.

The Mark III is the third generation of holographic video displays that MIT has developed since the early 1990s. The Mark I and II required specialized hardware to produce video signals, offered low-resolution images, were as big as a dinner table, and were tricky to work with, Bove said.

The new system processes 3D images via a standard graphics processor rather than specialized hardware.

Also, a new high-bandwidth, acousto-optic modulator that works with sound waves replaces a stack of acousto-optic modulators. It is thus smaller and less expensive.

The Mark III currently offers only monochromatic images, and its viewing volume is equivalent to an 80-mm cube, too small for practical applications such as PCs. Multiple modulators, one for each primary color, or one very fast modulator could provide color images, according to Bove.

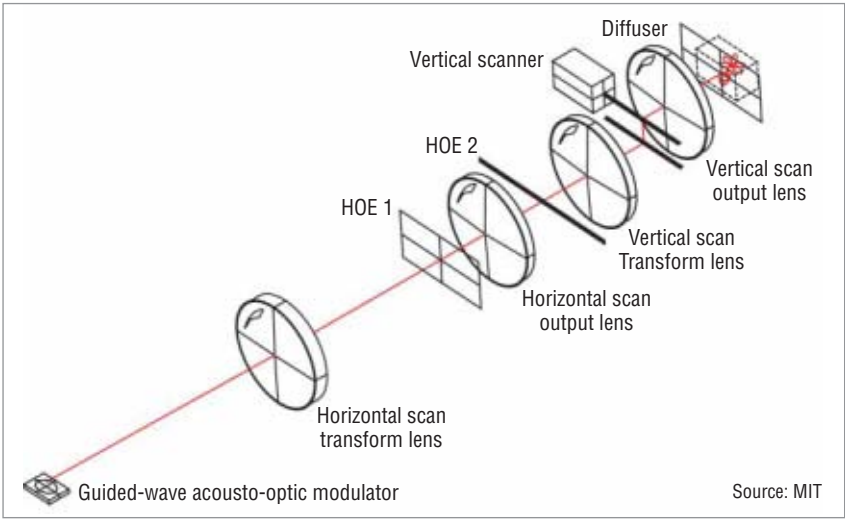
The researchers are working on a fourth-generation holographic display that shows larger, full-color images. However, Bove noted, "We won't start building it until we've learned some lessons from the Mark III."

His team is working with their corporate research sponsors to evaluate the technology's commercial potential.

A monochromatic display like the Mark III might be suitable for applications such as radiography, rather than consumer games, said analyst Jennifer Colegrove with iSuppli, an electronics-market research firm.

On the other hand, she said, a full-color version could capture a significant part of the 3D display market, which is expected to grow significantly. ■

—George Lawton
glawton@glawton.com



In the Mark III holographic display, a guided-wave device diffracts and modulates a laser beam to produce an image signal. The signal passes through stationary lenses and holographic optical elements (HOEs) to a moving vertical scanning mirror, which produces a raster that yields a holographic image when viewed through a diffuser.

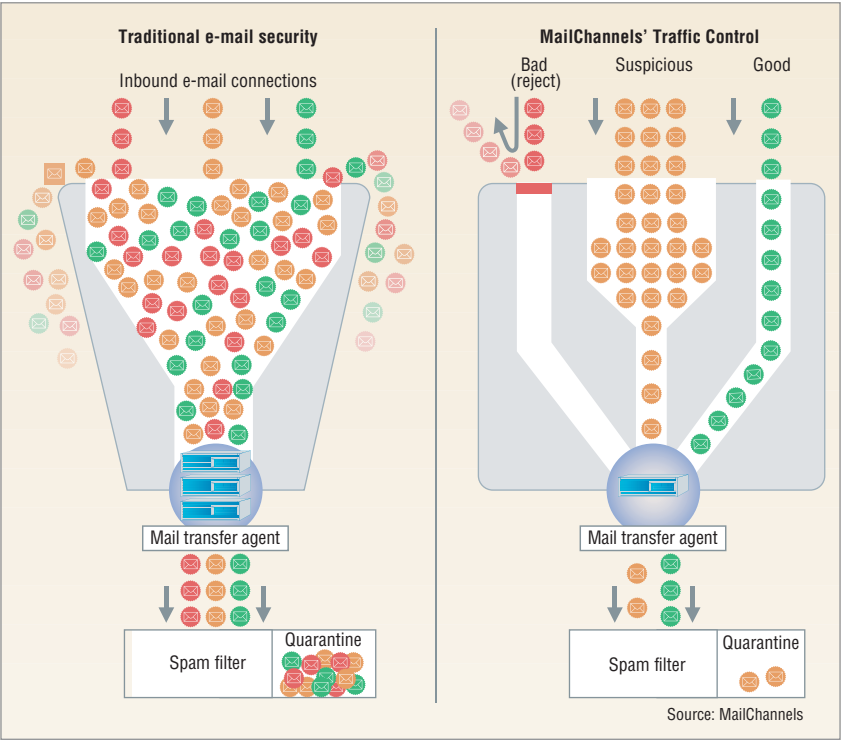
New System Tests Spammers' Patience

A Canadian company has developed a product that delays e-mail communications and thereby causes spammers to stop trying to send junk mail to users. MailChannels' Traffic Control uses this approach to take advantage of spammers' desire to quickly send as many messages as possible, maximizing the chance that some recipients will open them. Typically, when an e-mail server receives a request to accept an incoming message, it quickly responds to the sender, explained MailChannels' CEO and founder Ken Simpson. While this takes place, the Simple Mail Transfer Protocol (SMTP) keeps the connection open between the sender and receiver, thereby tying up the transmitter's resources.

Traffic Control software is usually installed in the e-mail server and receives messages before the server software. Traffic Control initially analyzes traffic to determine legitimate senders based on factors such as their reputation, protocol compliance, host characteristics, and message content. Messages from legitimate senders pass through without delay, and messages deemed to be harmful are blocked. For suspicious messages, Traffic Control causes the e-mail server to conduct the digital handshake a few bits at a time, making the process last up to 10 minutes, depending on user configuration. Legitimate senders' computers generally determine that there is a

problem with the connection and simply try again. "Even after eight minutes [of waiting], 60 percent of legitimate e-mail senders still try to deliver their message," Simpson noted. Spammers, on the other hand, configure the software that sends their messages, usually from hijacked zombie computers, to stop trying when messages don't transmit quickly. According to Simpson, 90 percent of spammers give up trying to send their message after 10 seconds of delays. Their resources are finite, and if they set their systems to keep trying to send messages in the face of delays, it reduces the volume of spam they can send, he explained.

Small pauses in the communications process can quickly add up, create a traffic backlog, and slow receiving e-mail servers to a crawl. The servers generally can't process enough backed-up traffic to recover quickly, Simpson said. In response, he noted, MailChannels has developed SMTP multiplexing, which lets thousands of incoming transmissions multiplex onto far fewer mail-server connections. This lets the user's server handle the traffic backlog in a manageable way, he explained. Traffic Control should be effective because it segregates legitimate and suspicious traffic and then handles the latter before it gets to the mail server and causes problems, said Carl Howe, a principal with Blackfriars Communications, a market research firm. Nonetheless, he said, no single technique will stop a large and complex problem like spam. Instead, he added, users must work with a variety of approaches. ■ —Linda Dailey Paulson ldpaulson@yahoo.com



MailChannels' Traffic Control handles spam by analyzing e-mail to identify legitimate, harmful, and suspicious messages. Traffic Control admits legitimate messages and blocks harmful ones. The product causes the e-mail server to slow the digital handshake with suspicious messages, causing spammers to stop sending them.

Group Works on Open Source Interoperability

A new consortium is trying to make open source software from different vendors work together so that the products can better compete against the suites of integrated and interoperable proprietary applications that many large software companies offer.

The newly formed Open Solutions Alliance (www.opensolutionsalliance.org) plans to work toward enabling open source application interoperability, certifying quality approaches to integration, and promoting cooperation among developers.

There are few interoperability standards for open source business applications, and they aren't applicable to building a suite of interoperable programs, noted OSA board member Barry Klawans, chief tech-

nology officer at JasperSoft, an open source business-intelligence vendor.

Companies thus have had to spend considerable time and money trying to make open source applications work together, which has hindered the software's adoption.

In response, the OSA will document guidelines and best practices for building and deploying interoperable applications. The effort is designed not to push members' products but to provide a resource for companies, Klawans said. The alliance also wants to offer information on open source license-management issues, help arrange collaborative projects among vendors, and provide technical support to sellers and users.

As part of this effort, the OSA wants to ensure that its approaches

will make open source software work together smoothly even when running on proprietary platforms, in which many companies have invested heavily.

The alliance has already produced a roadmap to show the problems it will deal with, as well as a timeline for developing standards, best practices, and prototypes of interoperable open source software.

To enable interoperability, the OSA is considering recommending both APIs and the use of common elements within applications that would enable the programs to work together.

The OSA has announced that it has begun work on its first major interoperability prototype: the Common Customer View. The CCV

Company Cleans up after Online Messes

A startup company has begun offering a service that tries to remove information online that could hurt clients' social reputations or even damage their chances of getting a job.

ReputationDefender (www.reputationdefender.com) tracks online information relating to customers—such as photos, blog postings, and social-networking activities—for \$10 per month. For \$29.95 per incident, the company asks Web sites and data-hosting services to remove potentially damaging materials. When unsuccessful, the company refunds its fee, said CEO Michael Fertik.

When removal isn't practical, the company can make unwelcome content show up less prominently in search engines so that it is less likely to be seen, he noted.

ReputationDefender offers three services: My Reputation for adults; My Child for parents who want to protect their children's online reputations; and My Privacy, currently in alpha testing, for removing private data—such as Social Security and driver's license numbers, home addresses, and phone numbers—from Web sites.

Using site-scraping robots and human research, ReputationDefender searches the Internet, particularly

social-networking sites, for potentially harmful material.

"Fully half of our clients use our service just to search for information. They want to keep up to date on what's being said about them online," noted Fertik. "The requests [to eliminate information] are from clients who are looking out for their professional lives, their romantic lives, and their personal reputation and sense of self."

"If someone is calling you a nasty name or publishing a photo that doesn't need to be on the Internet, we don't think you need to suffer from that kind of indignity," he said. "We also believe it is the right of individuals to know what others are saying about them and of private individuals to protect themselves from unintentional, inappropriate, or illegal intrusions into their privacy."

However, there are certain steps the company refuses to take. For example, it will not try to remove news articles or government records. And, Fertik explained, "We will not hack sites, use viruses, carry out [denial-of-service] attacks, or do anything 'black hat.' We're open and transparent with our techniques. And our approach is very civil and very polite." ■

—George Lawton

NEWS BRIEFS

will provide a common way for users to work with data and will also let them perform tasks such as logging on to an entire suite of applications at one time.

The CCV will also address issues such as a common look and feel for open source application interfaces and real-time synchronization among programs.

The prototype will use Talend's open source data-integration software and expertise from Unisys and SpikeSource, an open source application-management vendor. In addition to these companies, the OSA

includes vendors such as Centric CRM and Enterprise DB.

Several large open source vendors—MySQL, Novell, Linux vendor Red Hat, and SugarCRM—have not joined the OSA. "This will hurt the alliance," said Perry Donham, director of enterprise-integration research with the Aberdeen Group, a market research firm.

Moreover, Donham noted, the OSA must compete with the Interop Vendor Alliance (<http://interopvendoralliance.org>), to which

numerous open source and proprietary software firms, as well as hardware companies, belong.

To be most useful, he said, the OSA should focus on open source applications' interoperability with proprietary applications as well as with other open source software. ■

—Linda Dailey Paulson

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

Want to win a free trip to SC07?

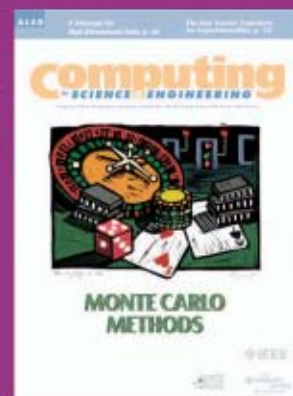
Francis Sullivan (IDA Center for Computing Sciences) has a challenge for you...

What would YOU compute?

"A lot of my working life has been spent arguing for more compute power and explaining the wonderful things that will happen when a more powerful machine is delivered. So far, nobody has called my bluff by saying, 'Okay, you can have all the computing you want. Now, what are you going to do with it?'"

Let's fantasize. Suppose you had available, at your desk, a petaflop machine. Assume that power and cooling were somehow taken care of and that you could get the full power of the machine by writing programs in a flexible and expressive language, no heroics necessary."

Email your answer in a 1,000-word essay to Jenny Stout (jstout@computer.org), with the subject line "petaflop challenge" by 10 August 2007. The winner will receive a year's subscription to *Computing in Science & Engineering* magazine and will see their entry published in the magazine's November/December 2007 issue. SC07 will also cover the winner's travel costs to attend its conference in Reno, Nevada, USA. SC07 is the international conference for high-performance computing, networking, storage and analysis. Visit <http://sc07.supercomputing.org> for conference details.



The Case for Flexible NIST Security Standards



Feisal Keblawi
Independent consultant

Dick Sullivan
Independent consultant

A public dialogue on the role and reach of NIST as a rule maker and as a standards writer for federal cybersecurity is essential to understanding the limits of security rule making in the present federal environment. NIST should delay converting its guidelines into rules until advances in the technology permit such conversion.

Recently, the US National Institute of Standards and Technology (NIST) began issuing a new kind of information system security (ISS) standards. Responding to the 2002 Federal Information Security Management Act (FISMA), these mandatory standards regulate ISS processes in federal civilian agencies and require standardized security controls in all related federal information systems.

Experience shows that federal standards aligned with established commercial practices generally succeed. However, unique government-only standards, such as the Government Open Systems Interconnection Profile (GOSIP), have achieved poor results. On the one hand, NIST has contributed to raising the quality of federal information security by promoting operational norms and by helping agencies to find model security processes. On the other hand, by using its rule-making authority to inappropriately impose standards, NIST—under the pressures associated with implementing FISMA—has the potential to create conflict and unduly raise costs without equivalent payoffs.

We seek to initiate a dialogue on the role and reach of NIST as a rule maker and as a standards writer for federal cybersecurity that will result in a new understanding about the limits of security rule making in the present federal environment.

NIST INVOLVEMENT

Beginning with its founding in 1901 as the National Bureau of Standards, NIST has played a key role in US commerce through promotion of various national standards.

Americans benefit in numerous ways through NIST's role in standards, which range from automotive safety and metrics for international clothing to information technology and security. NIST delivers its services both directly and through partnerships with various standards organizations, such as the American National Standards Institute and the International Organization for Standardization (ISO).

In the 1970s, NIST began issuing standards and guidelines for federal ISS. Generally, NIST products have helped to improve overall security quality. However, reading and absorbing the growing volume of standards and guidelines is a significant challenge. Presently, NIST maintains about 85 ISS guidelines and 15 standards on its Web pages (<http://src.nist.gov>). Some of these are voluminous.

The Computer Security Act of 1987 formalized NIST's role in setting computer security standards. FISMA replaced the Computer Security Act, and it authorizes the Secretary of Commerce to prescribe and mandate minimum federal information security standards. NIST guides this activity in the secretary's behalf (<http://src.nist.gov/policies/FISMA-final.pdf>).

In February 2004, NIST issued *Federal Information Processing Standard Publication 199, Standards for Security Categorization of Federal Information and Information Systems* (FIPS PUB 199).¹ This standard requires that all federal civilian agencies categorize information and information systems according to its newly established security impact rules. Further, in March 2006, NIST finalized FIPS PUB 200, *Minimum Security Requirements for Federal Information and Information Systems*.² This second standard mandates minimum security controls for federal information systems according to the impacts established per FIPS PUB 199.

With these two new standards, NIST will cause the federal civilian government to launch far-reaching, complex, and expensive changes. Because the new security controls involve operational, managerial, and technical mechanisms, they could result in new requirements for staff in addition to the need for new technology.

Moreover, we are uncertain that commonly used commercial products found in most agencies contain the features needed to comply with the new requirements. The new standards go beyond “stretch goals” and could simply increase criticism of federal agencies without commensurate security benefits. According to Steve Kerr, chief learning officer of the General Electric Company, a stretch goal is “an extremely ambitious goal, which gets people to perform in ways they never imagined possible.” (*Stretch Goals: The Dark Side of Asking for Miracles*; <http://faculty.css.edu/dswenson/web/335article/stretch.htm>). Kerr made the point that assigning stretch goals without also providing the resources needed to achieve them is counterproductive and could damage people and organizations.

Conceptually, the new standards seem straightforward, but this perspective ignores the mammoth size and complexity of the federal civilian government’s information and systems. Each affected federal organization, both large and small, has a unique mission that deals with complex information and systems—which have never been categorized under the NIST scheme.

At a minimum, the new NIST standards have the appearance of imposing significant new workloads to determine agreed-to security impacts. Additionally, most existing federal agencies do not meet the minimum security requirements of FIPS PUB 200. Federal managers will need to begin planning, budgeting, and staffing to bring systems into FIPS PUB 200 compliance.

Federal auditors, such as those from the Government Accountability Office (GAO), will likely apply the new standards in a pass-fail mode, and the results will become the basis for criticizing ongoing federal operations at a time when full compliance might be impossi-

ble and when more limited agency-determined controls might be sufficient. Recently, chief security officers in several federal agencies have suggested that system-by-system auditing, used in required FISMA reporting—and closely tied to the new NIST standards—might not correctly represent the federal cybersecurity posture (www.gcn.com/print/25_7/40249-1.html).

Had NIST issued guidelines for the conduct of security processes instead of issuing standards as in FIPS 199 and FIPS 200, there would be fewer issues to address, and this article would not have been written. We suggest that a reevaluation of the use of cyber rule-making might be desirable.

The new standards could simply increase criticism of federal agencies without commensurate security benefits.

CURRENT FEDERAL TECHNOLOGY ENVIRONMENT

The current federal environment provides a context for assessing application of the new NIST standards.

Congress has long expressed frustration with the high costs, inefficiencies, and ineffectiveness of federal information systems. Laws, such as the Clinger-Cohen Act of 1996, reflect a desire to improve management of these systems (www.ed.gov/policy/gen/leg/cca.html). However, because formidable human, technical, and environmental factors often conspire to block congressional goals, legislation has not produced the hoped-for results.

Numerous studies reveal recurring patterns of management and engineering shortfalls, but these cases might simply reflect human inability to estimate the challenges facing large federal information technology projects. Evidence suggests that some megaprojects might exceed human capabilities to complete them.³ Congressional interference in making changes in information system plans and budgets toward its own political ends, including the recent problem of growing earmarks, exacerbates this situation.⁴

Powerful interest groups from academia, industry, states, local governments, and the accounting community bring external, and often conflicting, pressures into the federal information systems venue. Education-to-experience imbalances might exist, where personnel have either long job experience or necessary technical education but not both.

Thus, credible federal mandates must strike a balance between ideal and practical standards, including setting realistic expectations for compliance in the huge base of installed federal systems. Additionally, we must remember that compliance will be put in jeopardy if the standards are perceived to be unreasonable or not viable.

INFORMATION SECURITY PRACTICE

Current ISS practice is as much an art as science, as reflected in the title of Matt Bishop’s popular textbook,

Computer Security: Art and Science (Addison-Wesley, 2003). Further, the nature of cyber threats is poorly understood. On the one hand, reputable security practitioners call for high levels of controls, making a case that cyberthreats could disable critical national infrastructures. On the other hand, different experts suggest that physical assaults can produce greater harm with a higher likelihood of materializing.

Placing cyberthreats in the context of the national security portfolio and establishing related priorities is a challenge. For example, in the three years between the President's Commission on Critical Infrastructure Protection and 11 September 2001, municipal water suppliers placed a high emphasis on protecting supervisory control and data acquisition systems. Since 9/11, however, experts have deemphasized cybersecurity and focused more on physical security to protect against hazardous chemical and biological contamination.⁵ To date, no cyber doomsday scenarios have surfaced.

Historically, ISS has been treated as a specialty engineering discipline, such as human factors and safety, and it is often isolated from mainstream business and technology decisions. Thus, while the security community was expressing reservations about wireless technology, federal executives were already leveraging wireless agility to communicate in an anytime, anyplace fashion. When security professionals raise theoretical instead of practical concerns, they might be seen as more of a hindrance than a help with achieving business objectives, and this could undermine credibility of advice from the security community.

Much good work has been accomplished to study and refine an engineering framework for ISS, but in the end this work currently relies heavily on hypothesis and logic. Thus, we lack systematic research and data that show a clear and measurable relationship between application of the new NIST security standards and a reduction in losses from cyberattacks. We certainly lack a means to precisely quantify the required number and strength of control mechanisms.

Despite its high technology setting, cybersecurity is still a fundamentally human activity that relies on human judgment to provide the flexibility needed to defend against likely attacks. This flexibility must encompass the huge range of federal systems. Some are small, simple, and isolated; others are large, complex, and interconnected. Some are new acquisitions, and others are scheduled for near-term replacement.

NIST's guidelines strike a balance between general rules of thumb for all agencies and the local knowledge and expertise of on-the-ground federal officials. However, fixed, inflexible process standards cannot easily accommodate all of these situations.

In summary, the infancy of ISS and its high reliance on human activity coupled with the rapid changes in technology make it essential that security doctrine remains flexible.

NATURE OF SUCCESSFUL TECHNOLOGY STANDARDS

In *Contemporary Marketing* (Harcourt Brace, 1992), Louis E. Boone and David L. Kurtz indicated that products go through four stages: introduction, growth, maturity, and decline. In early stages, suppliers compete on innovation, and product features vary significantly among manufacturers. In later stages, however, product features coalesce, and suppliers compete more on cost, reliability, performance, and service aspects. This coalescing allows for the development of standards that the free market voluntarily promotes. Thus, technical features of first-generation PCs varied

substantially among different manufacturers. Currently, however, differences among PC products are much less conspicuous, having achieved design consensus with a de facto market standard.

Whether flexible or rigid, standards must be appropriate for the activities being regulated, and they must be mindful of market drivers and required precision. The precision and specificity in standards vary considerably according to their goals and purposes. For example, some technical standards, such as communications protocols, must be very precise and rigid because of a need for interoperability among many vendors' products. In contrast, ISO 9001, widely used for quality management, is a voluntary international standard in which companies chose to participate because of perceived market advantage.⁶ Additionally, ISO 9001 is a flexible standard owing to the wide diversity of businesses seeking certification. Among other benefits, ISO 9001 could help companies gain new customers by establishing a common basis for trusting company products.

Among nine definitions, *Merriam-Webster's Collegiate Dictionary* describes a standard as "something established by authority, custom, or general consent as a model or example" and as "something set up and established by authority as a rule for the measure of quantity, weight, extent, value, or quality." Thus, viable standards need several properties.

First, standards require reliable metrics to enable tracking of compliance. Second, they must be introduced at a specific point in the product life cycle when customers seek standard products and manufacturers are no longer competing on features. Third, there must be a compelling market benefit supporting use of a standard. Finally, standards must be appropriate for the application being standardized.

Whether flexible or rigid, standards must be appropriate for the activities being regulated.

In computer and networking technology, it is easy to see how these properties apply, and there are rich examples of how market-driven standards succeeded when idealized and mandated standards failed. For example, federal agencies have abandoned many unique standards such as the military standards (MIL-STDs) because they lacked flexibility and led to expensive and complex acquisitions. In 1994, Secretary of Defense William Perry essentially prohibited the use of detailed MIL-STDs.⁷ These standards sometimes blocked use of highly desirable but nonconforming technologies and commercial-off-the-shelf products. Instead, COTS products and their indigenous standards are preferred because they are cheaper to acquire and operate, and they avoid the need to maintain warehouses full of unique spare parts. The Defense System Software Development Standard (DOD-STD-2167A) provides another example in which the government shifted away from mandating a unique federal standard in favor of industry standards.

In general, systems development has increasingly emphasized flexibility and integration of COTS products when organizations must make tradeoffs between available products and agency requirements. Experience shows that standards mandated in inflexible policy that does not account for real-world considerations will ultimately fail. Many will recall the “C2 by 92” security mandate in which policy makers directed federal agencies to achieve the C2 evaluation level of the Trusted Computer Security Evaluation Criteria (a Common Criteria predecessor) by 1992. Most observers would likely agree that affected agencies never came close to meeting this objective.⁸

As another example, consider the huge success of the Internet protocols contrasted with the GOSIP standard, which NIST published in 1990 with directions for government agencies to adopt it.⁹ By then, however, the Internet Protocol family was already leading the commercial networking market, even though it lacked many of the idyllic features in GOSIP. Many will remember substantial federal investments in GOSIP, which have subsequently been converted to Internet protocols at significant conversion costs to taxpayers to achieve the substantial benefits of using lower-priced commercial products. In some sense, this experience demonstrates the benefits of selecting market-driven standards as they arise instead of imposing government-only standards, which might never find a natural acceptance in the evolving technology framework.

Arguably, cybersecurity is still in a growth phase where manufacturers compete on features. Additionally, we lack reliable metrics for evaluating the contribution or implementation of process standards. Benefits might

accrue from using security standards, but they might not presently be compelling enough to dominate other considerations.

Thus, the cybersecurity market might lack the maturity needed to embrace new mandatory federal standards. Consequently, developing guidance for ISS processes and best practices might be more effective than mandating standards. In addition to exploiting an agency’s expertise in addressing its individual security situations, such processes will also take advantage of any organic security that was developed to meet the market demand for standardization of security features.

Experience shows that standards mandated in inflexible policy that does not account for real-world considerations will ultimately fail.

MISMATCH BETWEEN EXPECTATIONS AND REALITY

Federal information security has received considerable public attention over the past 25 years as computers and networks have taken an increasing role in society and as related security risks have come into clearer focus. Various stakeholder groups, both inside and outside government, take alternative positions on appropriate federal security and advocate their particular views to Congress and the administration, including NIST. A few individuals persist with “sky is falling” scenarios and related calls for immediately fixing all security problems. However, experience shows that perfect security cannot be achieved—at least with acceptable political and economic costs. Sorting out the public costs and benefits among the different alternatives affecting federal security programs is no easy challenge. Thus, NIST does not operate in a political vacuum.

During the Clinton administration, the President’s Commission on Critical Infrastructure Protection began an exploration of national risks from cyberattacks. The PCCIP report affirmed the need for a new public policy focus on securing our critical national infrastructure (www.emergency.com/pcciprpt.htm). Presidential Decision Directive Number 63 set in motion various federal initiatives to begin addressing the PCCIP findings.¹⁰ Following this early work and especially because of 9/11, the Bush administration has continued to focus on critical infrastructure and cybersecurity, and it has launched and funded several key initiatives. One of these, the February 2003 *National Strategy to Secure Cyberspace* identifies priority areas for attention and greater protection (www.whitehouse.gov/pcipb). Accordingly, there is a strong consensus on the need for reducing potential cyber-risks both inside and outside government, but the overall approach is still evolving.

In contrast to these goals, government agencies have been roundly criticized for poor cybersecurity, and this likely reflects deeper problems relating to the realism of expectations and measures of success. According to a report from the House Government Reform Committee

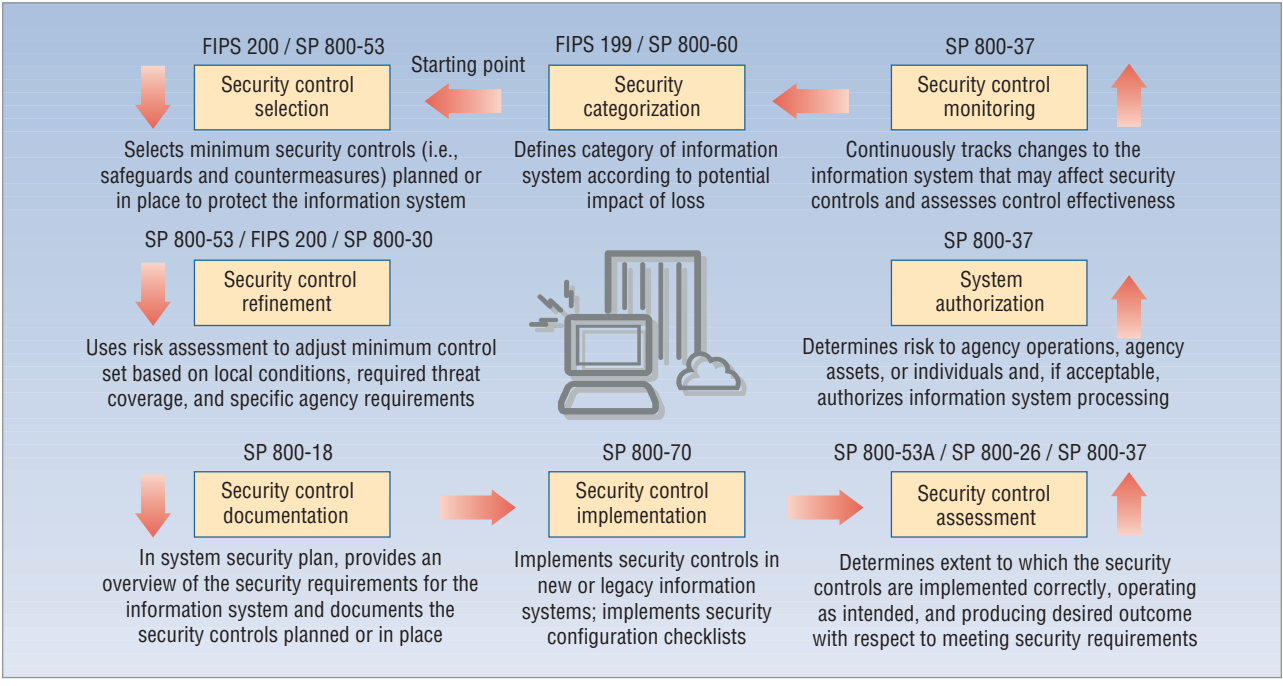


Figure 1. Managing enterprise risk. The process begins with security categorization and proceeds to control selection before conducting risk assessments.

citing FISMA-required status reporting for 2005, most federal civilian agencies are rated marginal to failing on computer security.¹¹ A recent issue of *Government Computer News* cites Bob Lentz, assistant secretary of defense (www.gcn.com/print/25_7/40277-1.html), and Bruce Brody, former chief security officer at the Departments of Veterans Affairs and Energy (www.gcn.com/print/25_7/40249-1.html), regarding problems with FISMA. Both officials suggest that the FISMA metrics—closely linked to the NIST standards—might not correctly represent the federal cybersecurity posture. Specifically, the present system-by-system approach could divert attention from corporate security infrastructures, which are needed to implement effective agency security programs.

In any event, there has been an overabundance of criticism without balancing analysis about underlying causes of the poor grades. One likely underlying cause is that the standards being applied are too ambitious, too premature, or invalid.

CONCERNS ARISING FROM NEW STANDARDS

Specific concerns arise from the new NIST standards.

Regulating agency processes

We take issue with the use of NIST standards to regulate agency processes instead of technology. That is, the NIST documents mandate how an agency will go about selecting and establishing requirements for specific types of security controls. This differs from previous NIST standards, which established technology regulations such as those for cryptographic algorithms,

time standards, and so on. This also varies from past successful NIST guidance on best security practices, such as performing risk assessments.

The new standards appear to conflict with agency responsibilities for making decisions about information security. Agencies must balance ISS controls with other priorities and programs, which also cost money.

Had NIST published guidelines instead of standards, there would be fewer problems because the federal agencies would accept these guidelines as security planning frameworks.

Weakening risk-management policy

In taking a so-called high-watermark stance that requires all possible controls, the new standards appear to undermine agency responsibilities for cost-conscious security controls. The high-watermark concept represents a risk-avoidance philosophy that argues for fixing all possible security problems. Rather than being based on risk avoidance, NIST standards should be heavily rooted in risk management, which addresses high priority concerns first. Risk management is the cornerstone of federal security policy, as Office of Management and Budget (OMB) Circular A-130 Appendix III explains (www.whitehouse.gov/OMB/circulars/a130/a130appendix_iii.html).

Figure 1 shows a NIST chart for applying the new standards.¹² The process begins with impact definition—shown as security categorization—and proceeds to control selection before actually conducting risk assessments in step three. This approach disassociates impacts from

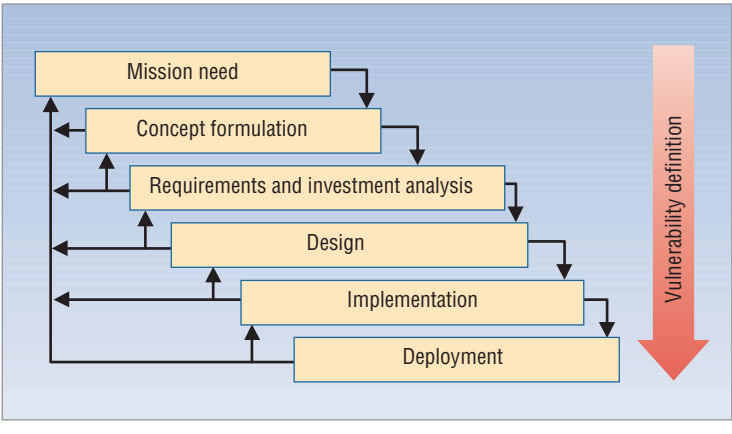


Figure 2. System life cycle with feedback loops. Distinct phases involve coordination and stakeholder buy-in. Several years might separate the concept formulation phase from the implementation and deployment buy-in phases.

potential attacks as is done in traditional risk assessments. That is, according to FIPS 199, impact must be assigned without regard to feasible attacks based on specific threats and vulnerabilities. Thus, NIST equates impact from a cyberattack with the mission value of information and information systems.

We maintain, however, that mission impacts from security breaches cannot be assessed until specific threats and vulnerabilities are known. According to NIST Special Publication 800-30, a risk constitutes a threat operating on a vulnerability, and impacts map to these threat-vulnerability (T-V) pairs. Thus, impact is not some generalization that arises independent of specific threats and vulnerabilities.

As NIST SP 800-30 states, risk management considers both the likelihood and impact of a cyberattack.¹³ The new standards appear to ignore likelihood (or probability) and base control selection exclusively upon impact—albeit without the valid use of T-V pairs.

At least a part of the new emphasis on impacts seems to arise from an implied NIST assumption that impact can be quantified where likelihood cannot. However, discussions with security analysts suggest that impact is often a function of an analyst’s point of view and knowledge of the system and its information. Different analysts are likely to have diverging opinions on the impacts of cyberattacks, so assessment of impact might be as subjective as assessment of likelihood.

The FIPS 199 guidance on impacts is ambiguous. Using quantifying language such as minor, significant, and major has the appearance of quantification, but these terms are just as subjective as likelihood because different analysts might apply these terms differently. Further, we contend that linking impact to the presence of valid T-V pairs actually reduces the subjectivity of impact evaluation.

Finally, in focusing on individual systems and information instead of agency services, the new standards do

not adequately consider that agencies often manage risk by providing service in multiple ways. Taken in isolation, a single system’s compromise might lead to a higher assessment of impact than would otherwise be the case when all of the backup capabilities supporting service delivery are considered.

Ignoring life-cycle phases

As Figure 1 shows, NIST contends that the starting point for enterprise protection is the security categorization of impacts. While impacts can be estimated for an existing system, the NIST approach is far from adequate for a system that will be further defined at various life-cycle phases—where many related feedback loops exist.

As Figure 2 illustrates, system acquisition has distinct and strenuous phases involving considerable coordination and stakeholder buy-in. Several years might separate the concept formulation phase from the implementation and deployment phases. Stakeholders can identify mission value during concept formulation, but system risks only clarify through identification of specific threats, vulnerabilities, likelihoods, and impacts. Thus, mission value and security impact cannot be equated as FIPS 199 does.

Rigid accounting standard

Federal inspector generals and the GAO will likely apply the new NIST standards as rigid and inflexible accounting standards on a system-by-system basis. Rigid application of these standards will likely result in unwarranted criticism of federal agencies, and it will ultimately and unduly increase costs for federal information systems and unjustifiably undermine public perceptions of government.

In the private sector, many companies have raised concerns about similar results from the Sarbanes-Oxley Act of 2002.¹⁴ While this law does not address information security directly, accounting firms and standards organizations have interpreted its call for financial accountability as requiring greatly strengthened information security controls.

Some observers believe that the new accounting standards have not materially improved corporate security, but they have unquestionably enriched accounting firms and vendors of security products. If ISS were a solved problem, detailed standards rather than flexible guidelines might be appropriate, but ISS concepts remain in flux. Clearly, there is agreement about general security processes that must be in place in federal agencies—for example, personnel security, physical security, risk management, separation of duties, and others. However, applying the new NIST standards will require resource levels well beyond what current processes and practices

require, and this could divert attention from other higher priority needs.

In some cases, ISS auditors appear to simply apply the accounting standards without analyzing and making judgments about appropriate use, especially regarding technical controls such as encryption. A more practical approach might be determination of what security processes are actually practiced along with the identification of weak processes—not specification of actual controls. Examples include answering questions such as the following: Do agency managers have a process for learning about security concerns? Are managers tracking and prioritizing significant concerns on a routine basis? Are managers taking steps to address concerns? It turns out that the answers to these questions are material to security.

No implementation plan

Both FIPS 199 and FIPS 200 are effective immediately. However, neither one addresses transition issues such as cost, schedule, size of project, and legacy systems. Indeed, although federal rule-making generally requires impact assessments prior to implementation, NIST has not published any estimates of the cost of implementing the new standards or a roadmap for achieving them.

In the technology area alone, it is not clear that available commercial products can satisfy all of the stated requirements, especially at the high baseline for security controls. Consequently, agencies could face conflicts between meeting the standards and selecting technology that satisfies agency needs.

Because the new standards are effective immediately, they do not give agencies any assistance with legacy systems. Further, it makes little business sense to apply expensive retrofits to systems nearing the end of their service life, but the standards appear to have no flexibility in this respect.

In exercising its new FISMA powers, NIST appears to be setting stretch goals that are beyond practical attainment. The consequences of this approach will likely be decades of failing grades on federal information security. NIST seems to be repeating mistakes from past efforts at imposing inflexible, government-only standards such as GOSIP at a time when most of the country is not ready for security standards. Further, the new standards might contribute to a rosy picture suggesting that all problems can be solved through more rules and regulations.

NIST should reconsider the use of FIPS 199 and FIPS 200. It might be advisable to revise the standards to be more closely aligned with SP 800-30, which has considerable support in the community. Instead of mandating standards, NIST could use the more flexible approach of providing guidance to improve security

processes without mandating them. If there is a need for top-down regulation of agency processes, as opposed to technology, the OMB is in a better position to mandate change while considering the costs associated with new regulation and also ensuring funding and other resources for approved initiatives.

NIST should exercise restraint in mandating federal security standards by applying a cost and feasibility test. We recommend that NIST produce an impact assessment and use it as a moderating influence prior to requesting compliance with any new rule. ■

References

1. US Government, "FIPS PUB 199, Standards for Security Categorization of Federal Information and Information Systems," Feb. 2004; <http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf>.
2. US Government, "FIPS PUB 200, Minimum Security Requirements for Federal Information and Information Systems," Mar. 2006; <http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf>.
3. C. Jones, "Project Management Tools and Software Failures and Successes," *CrossTalk: The J. Defense Software Eng.*, July 1998; www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1998/07/tools.asp.
4. D. McCullagh, "Pork Barrel Technology Projects on the Rise," *CNET News.Com*, 30 Mar. 2006; <http://news.com.com/2009-1028-6050719.html>.
5. US Government, "Drinking Water: Experts' Views on How Federal Funding Can Best Be Spent to Improve Security," Government Accountability Office, Sept. 2004; www.gao.gov/new.items/d041098t.pdf.
6. R. Tricker and B. Sherring-Lucas, *ISO 9001:2000 in Brief*, 2nd ed., Elsevier Butterworth-Heinemann, 2005.
7. Office of the Assistant Secretary of the Navy (Research, Development and Acquisition), "Lessons Learned"; <http://acquisition.navy.mil/aosfiles/tools/specright/ts3.htm>.
8. US Government, "A Summary Guide: Public Law, Executive Orders, and Policy Documents: NTISSP No 200, National Policy on Controlled Access Protection," 15 July 1987, Dept. of Treasury, 1991; <http://csrc.nist.gov/fasp/FASPDocs/program-mgmt/legal-requirements.doc>.
9. "Government Open Systems Interconnection Profile," 27 Jan. 2006; http://en.wikipedia.org/wiki/Government_Open_Systems_Interconnection_Profile.
10. National Security Council, "Presidential Decision Directive/NSC-63: Critical Infrastructure Protection," 22 May 1998; www.fas.org/irp/offdocs/pdd/pdd63.htm.
11. B. Krebs, "Agencies' Computer Security Sharply Criticized," *The Washington Post*, 16 Mar. 2006, p. D5.
12. R. Ross, "Building More Secure Information Systems," Nat'l Institute of Standards and Technology, 2001; www.ehcca.com/presentations/HIPAA10/ross.ppt.
13. NIST, "Risk Management Guide for Information Technology Systems," NIST SP-800-30, July 2002.

14. Bureau of National Affairs, "Law Profs Decry SOX as 'Debacle,' Express Hope for 'Re-Examination,'" *Corporate Law and Business*, vol. 38, no. 12, 20 Mar. 2006; <http://corplawcenter.bna.com/pic2/clb.nsf/id/BNAP-6MY23B?OpenDocument>.

Feisal Keblawi, formerly the information system security manager for developmental systems with the US Federal Aviation Administration, is retired from the FAA and is now an independent consultant. His work includes helping develop an information system security architecture for use as a framework for implementing security in the National Airspace System. Keblawi received a PhD in electrical

engineering from North Carolina State University, and is a Certified Information System Security Professional. Contact him at fkeblawi@gmail.com.

Dick Sullivan has assisted the FAA with the National Airspace System Architecture and related information security, both with TRW and SAIC and recently as an independent consultant. His research interests include public policy, transportation, information security, software engineering, and project management. Sullivan received an MS in software engineering from George Mason University. He is member of the IEEE Computer Society and the Air Traffic Control Association and is a Certified Information System Security Professional. Contact him at sullivan@verizon.net.



The advertisement features a warm-toned background with a grid pattern. At the top left is the IEEE logo. The word "Computer" is written in large, bold, red letters. Below it, the phrase "Welcomes Your Contribution" is written in a similar style. On the left side, a vertical yellow bar contains the text "Computer magazine looks ahead to future technologies" in white. At the bottom left, the IEEE Computer Society logo is displayed. On the right side, there is a list of three bullet points describing the magazine's content and editorial process. At the bottom right, there is a call to action to submit a manuscript for peer review, followed by a URL.

IEEE

Innovative Technology for Computer Professionals

Computer

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

Best Practices for Automated Traceability

Automated traceability applies information-retrieval techniques to generate candidate links, sharply reducing the effort of manual approaches to build and maintain a requirements trace matrix as well as providing after-the-fact traceability in legacy documents. The authors describe nine best practices for implementing effective automated traceability.



Jane Cleland-Huang, Raffaella Settini, and Eli Romanova
DePaul University

Brian Berenbach
Siemens Corporate Research

Stephen Clark
iRise

Requirements traceability, defined as the ability to “follow the life of a requirement in both a forward and backward direction,”¹ provides critical support for software engineers as they develop and maintain software systems. Traceability helps determine that researchers have refined requirements into lower-level design components, built them into the executable system, and tested them effectively. It further helps analysts understand the implications of a proposed change and ensures that no extraneous code exists.

Numerous standards include traceability as a recommended or legally required activity. For example, *IEEE Std. 830-1998* states that a software requirements specification must be traceable. The standard defines an SRS as traceable “if the origin of each of its requirements is clear and if it facilitates the referencing of each requirement in future development or enhancement documentation.” Backward traceability is required from requirements to “previous stages of development,” while forward traceability proceeds from requirements to “all documents spawned by the SRS.”²

Further, organizations building safety-critical systems are often legally required to demonstrate that all parts of the code trace back to valid requirements. Laws such as the US Sarbanes-Oxley Act of 2002 require organizations to implement change-management processes with explicit traceability coverage for any parts of a software product that potentially impact the balance sheet.

Unfortunately, many organizations fail to implement effective traceability practices either due to difficulties in creating, assessing, using, and maintaining traceability links or because they succumb to the misconception that traceability practices return little value for the effort involved.³

Traditionally, traceability links are physically stored in spreadsheets, text files, databases, or requirements management (RM) tools such as Telelogic’s DOORS or IBM’s Rational RequisitePro, and such links tend to deteriorate during a project as time-pressured team members fail to update them.¹ Offshoring and outsourcing exacerbate this problem by creating temporal and physical distance between subject-matter experts and developers.

Because manual traces are often created in an ad hoc fashion according to the developers’ whim, they tend to be inconsistent and often incomplete. Current RM tools provide limited support for link creation and maintenance, offering features

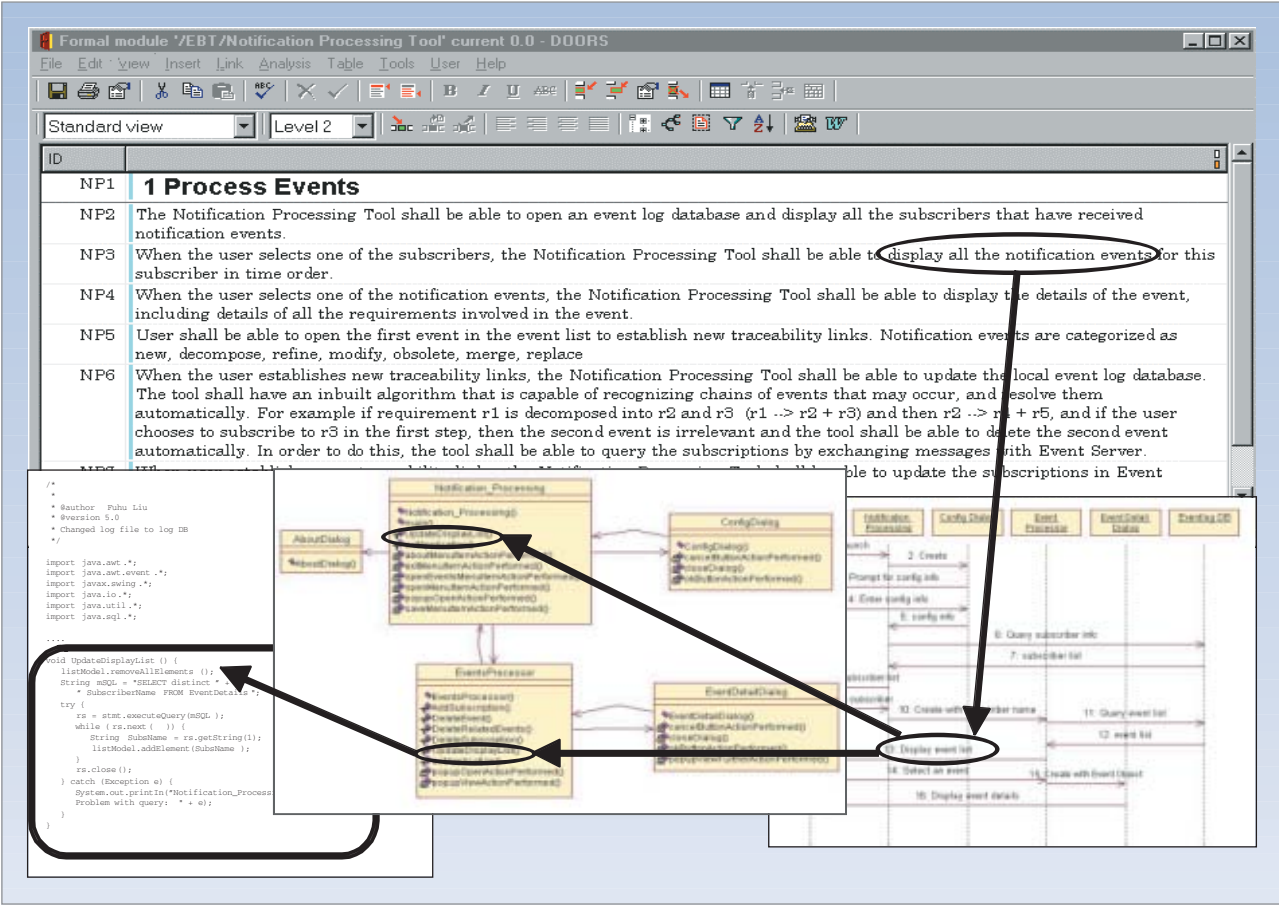


Figure 1. Automated traceability solutions retrieve links based on the similarity of terms across documents.

such as drag-and-drop techniques for creating links, simple visualization for depicting traces, and flagging of “suspect” links—those for which one of the associated artifacts has been modified, suggesting that the links might have become outdated. Unfortunately, this type of support does not sufficiently ease the burden of managing traceability links, and practitioners often find that increasingly larger percentages of links become suspect.

Further, there is almost universal failure to establish traceability between requirements and supplemental documents such as stakeholders’ rationales, vision documents, or other free-form textual documents. For many organizations seeking to attain a corporate-wide standard such as Capability Maturity Model Integration (CMMI) Level 3, achieving traceability in projects for which it was not initially a systematic part of the development process can seem like an arduous and often unachievable task.

Automated traceability methods aggressively tackle these problems by decreasing the effort needed to construct and maintain a set of traceability links and by providing traceability across a much broader set of documents.

THE AUTOMATED SOLUTION

Automated traceability relies on various information-retrieval algorithms⁴⁻⁸ that use techniques such as the *vec-*

*tor-space model*⁴ or the *probabilistic network model*^{6,7} to compute the likelihood of a link based on the occurrence and distribution of terms, as Figure 1 shows. During the preparing phase, VSM- and PN-based algorithms remove stop words such as “this” and “the” representing extremely common terms from the artifact text and stem remaining terms to their root forms so that similar words such as “registers” and “registered” are considered equivalent. For code and Unified Modeling Language (UML) documents, the algorithms separate method names that follow standard conventions of uppercase and lowercase usage and appear in forms such as “getLastName()” into individual words such as “get last name.” They also remove program-related keywords such as “while” and “for.”

VSM- and PN-based algorithms calculate a similarity score based upon the frequency and distribution of terms in the artifacts. Both types of algorithms belong to the *tf-idf* (term frequency-inverse document frequency) family of information-retrieval methods, which consider rarer terms to be stronger indicators of a potential link than more common ones.

Probabilistic network model

To better understand how these algorithms work, consider the PN-based approach. In this model, the basic

probability of a link between a query q and a traceable artifact a_j is defined as

$$pr(a_j|q) = \left[\sum_{i=1}^k pr(a_j|t_i)pr(q,t_i) \right] / pr(q).$$

The first two parts of the formula, $pr(a_j|t_i)$ and $pr(q,t_i)$, represent the dispersion of the term t_i within the artifact a_j and query q , respectively. These are estimated by computing the normalized frequency of terms. For example, $pr(a_j|t_i)$ is calculated by considering the frequency with which t_i occurs in the artifact, normalized over the total number of words in the artifact. This is represented as

$$pr(a_j|t_i) = \frac{\text{freq}(a_j,t_i)}{\sum_k \text{freq}(a_j,t_k)}.$$

Similarly, $pr(q,t_i)$ is calculated by considering the frequency with which term t_i occurs in query q , normalized over the total number of potential queries in which t_i occurs.

In the third part of the formula, $pr(q)$ is computed as

$$pr(q) = \sum_i pr(q,t_i).$$

using simple marginalization techniques and represents the relevance of the term t_i to describe the query concept q . The resulting probability is inversely proportional to the number of artifacts containing the index term, reflecting the assumption that rarer index terms are more relevant than common ones in detecting potential links.^{4,5}

PN-based algorithms translate raw probability scores into confidence levels that are more intuitive to a human analyst and depict the likelihood of each link being correct.⁹ Once the algorithm calculates the confidence scores, it returns a set of candidate links to the analyst for assessment. Our experience indicates that with adequate supporting documentation, an analyst can use this information to quickly make a decision about the correctness of a link.⁹

A trace query

Figure 2 shows the results returned by our Poirot: TraceMaker tool for the query “All

trucks shall display a map of the de-icing route.” Of the top 10 results displayed on the screen, five were actually correct; the analyst accepted these and rejected the other five. Several of the incorrect links were retrieved because the term “map” was used broadly across a number of requirements. A filtering feature (not shown) can be used by an analyst to help remove some of these extraneous links.

Experimental results

Researchers have conducted extensive experiments to assess the effectiveness of automated traceability methods. Results are typically evaluated using two metrics.¹⁰ The *recall* metric measures the extent to which all of the desired links are retrieved:

$$\frac{\text{Correct links} \cap \text{Retrieved links}}{\text{Correct links}}$$

The *precision* metric measures the percentage of retrieved links that are relevant:

$$\frac{\text{Correct links} \cap \text{Retrieved links}}{\text{Retrieved links}}$$

There is typically a clear tradeoff between recall and precision, so that as one increases the other decreases.

A third metric of particular importance in automated traceability measures the retrieval algorithm’s ability to place more good links at the top of the can-



Figure 2. Results from a trace query in Poirot: TraceMaker. Precision in the top 10 results was 50 percent.

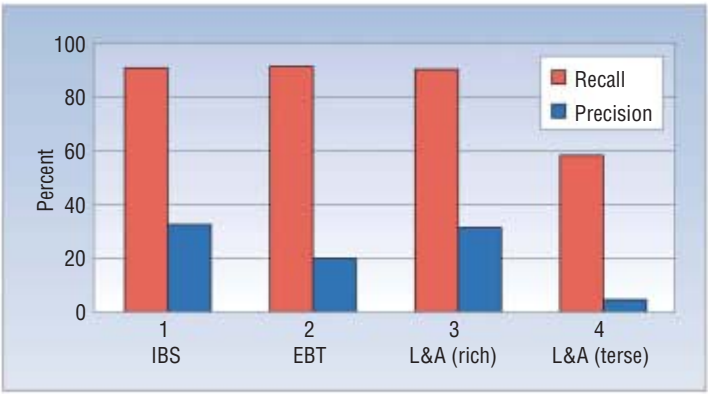


Figure 3. Trace retrieval results for several data sets. Data sets 1-3 all returned acceptable precision results ranging from 19 to 32 percent when recall was fixed close to 90 percent, while data set 4 returned an abysmal precision rate of 4 percent at recall levels of 58 percent.

didate list and to place bad links at the bottom. This can be calculated either by considering precision at various levels of recall or by utilizing the *lag* metric, defined as the average number of false links that occur in an ordered list of candidate links above each good link.⁷

Figure 3 shows experimental results for several data sets using the PN method. Because industrial applications of automated traceability are only considered successful when they achieve high recall levels, precision results are reported at a fixed recall level close to 90 percent.

The first data set represents the requirements and UML classes of a system for managing de-icing of roads in a public works department. The Ice-Breaker System (IBS) includes 180 functional requirements and 75 primary classes.⁶ The second data set represents an event-based traceability (EBT) system that includes 54 requirements and 60 classes.⁵ The third data set, provided by Siemens Logistics and Automation, represents a system for automating the layout of product lines in a factory. The artifacts used in the L&A experiment included business use cases (BUCs) and system use cases (SUCs), both represented as relatively terse requirements in RequisitePro. A subset of SUCs, labeled as “rich” in Figure 3, had additional associated information modeled in UML activity diagrams.

Data sets 1-3 all returned acceptable precision results ranging from 19 to 32 percent when recall was fixed close to 90 percent. In contrast, data set 4 performed poorly, returning an abysmal precision rate of 4 percent at recall levels of 58 percent.

These results suggest that not every data set will perform well. In fact, we have found that although automated traceability can provide a viable and economically beneficial solution, returning a good set of candidate traces requires properly structuring and specifying the artifacts.

BEST PRACTICES

Best practices for automated traceability fall into three categories. The first category describes best practices for establishing a traceability environment, the second describes the structuring and content of the artifacts, and the third describes a process for introducing automated traceability into an organization.

Traceability environment

The first three best practices relate to establishing the traceability environment.

Trace for a purpose. During the software development life cycle, project stakeholders create numerous work products that introduce the potential for a vast number of links. Although automated methods support after-the-fact traceability of legacy documents, stakeholders can perform day-to-day trace tasks more seamlessly if they determine in advance which types of artifacts they will trace and which types of artifacts they will trace to. For each identified traceability path, stakeholders must identify the artifact type at the origin and destination of each link, establish where each artifact is physically located, and determine in what format or third-party case tool it is stored.³

As an example, consider the traceability framework established for the L&A project’s requirements. This project captured several different types of requirements during the project’s life cycle from several unique role-oriented viewpoints. These included business goals, stakeholder requests, minimal marketable features (MMFs) that defined units of business value, MMF groups, BUCs, BUCs grouped by life-cycle stages, SUCs, and concrete system capabilities (CSCs), which were further refined into concrete system components.

A guiding principle during the requirements process was that it would express domain-specific requirements in the vocabulary of that domain’s participants. As a result, stakeholders were interested in different subsets of artifacts and had different traceability needs. For example, business end users would need to answer questions such as “Do all MMFs trace to one or more business use cases?” or “Are all system requirements derived from one or more MMFs, and through which BUCs do they trace?” In contrast, as developers would need to know how accurately and completely their system requirements were aligned with business priorities, they would need support for queries such as “Do all CSCs trace to one or more SUCs?” Figure 4 shows the artifacts, traceability paths, and rationales for each link.

Making up-front decisions and explicitly modeling the traceability needs of the project stakeholders provide the necessary physical infrastructure to support automated traceability.

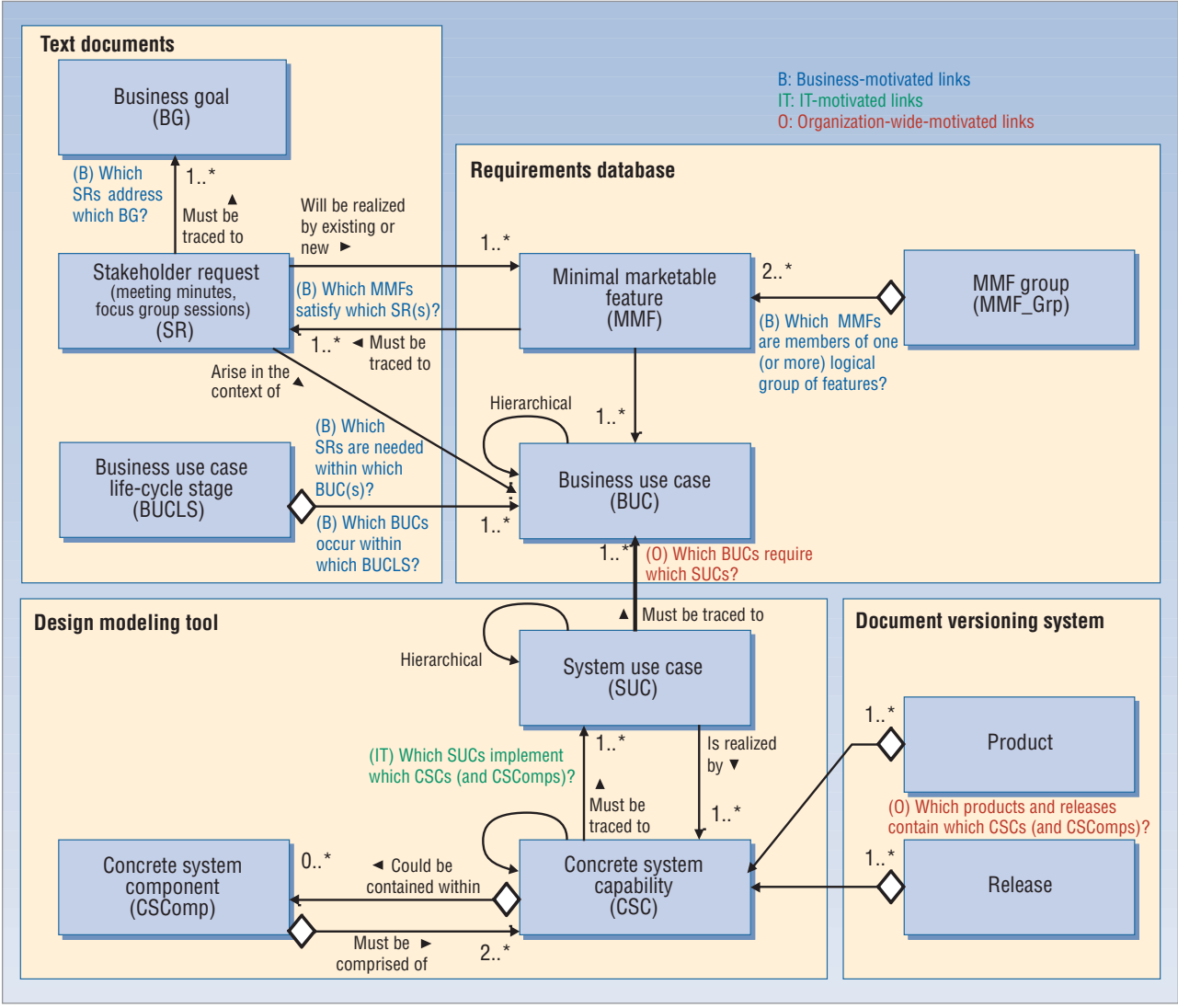


Figure 4. Requirements framework developed for Siemens L&A. Making up-front decisions and explicitly modeling the traceability needs of project stakeholders provide the necessary physical infrastructure to support automated traceability for business-, IT-, and organization-wide-motivated links.

Define a suitable trace granularity. Project stakeholders also must decide on the appropriate level of trace granularity for each artifact type. For example, when tracing to UML class diagrams, they could generate a trace at the package, class, or method level.

Alexander Egyed and colleagues evaluated the economic value of tracing at lower levels of granularity measured by the effort needed to create the links versus value returned through tracing at various levels of precision.¹¹ Even ignoring the costs of maintaining links, they found that the benefits of improving the granularity of trace links beyond a certain level were very limited. In fact, their case study showed a tenfold increase in granularity returned only a twofold improvement in precision (correct links/all retrieved links).

Granularity must be carefully determined to effectively support stakeholders in their traceability tasks, while

minimizing the effort involved to analyze and utilize the set of returned links. This can be especially problematic in large, weakly structured documents that might not contain clearly defined components at the desired granularity level. To mitigate this problem, automated traceability tools can cluster sentences into meaningful semantically related groups and then generate traces to those groups.

Support in-place traceability. In a software project, developers generally use a wide variety of third-party case tools to manage artifacts, and traceability should be provided to these artifacts as they reside within their native environments. We refer to this as *in-place* traceability.

Poirot delivers in-place traceability through an enterprise-level architecture, shown in Figure 5, that uses customized adapters to parse and reconstitute data from within third-party case tools. Each adapter uses the

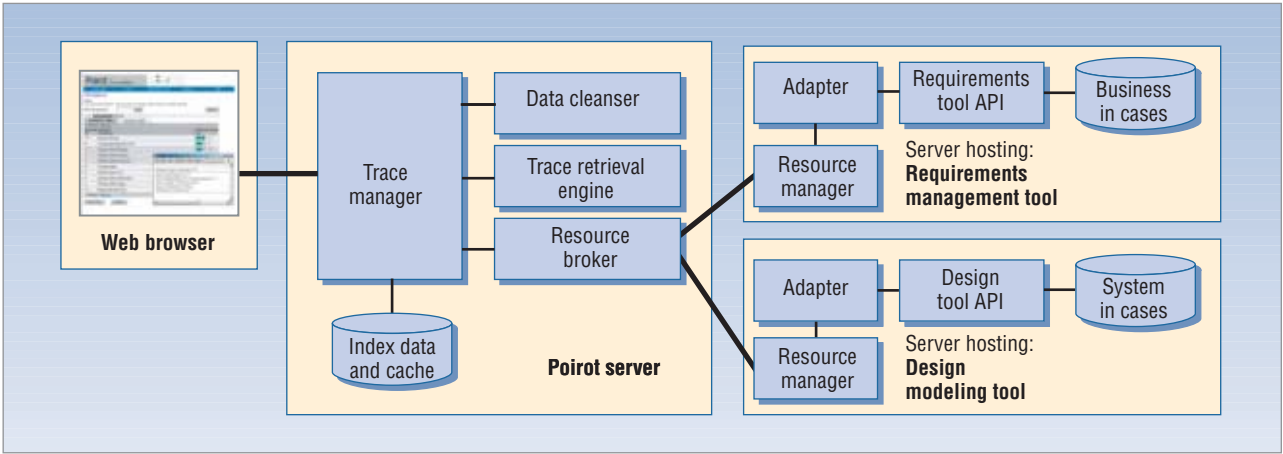


Figure 5. Poirt delivers in-place traceability through an enterprise-level architecture that uses customized adapters to parse and reconstitute data from within third-party case tools.

third-party case tool’s API to retrieve the traceable data and transform it into a standard format. The data then passes to the Poirt server, which removes stop words, stems terms, counts term frequencies, and carries out other related activities. Customized adapters are currently available for a limited number of model types within various third-party case tools.

To facilitate reuse, an adapter must also allow the user to define filter conditions to retrieve targeted artifacts and sift out unwanted ones. For example, the SUCs in the L&A project represent a subset of activity diagrams embedded deep within an IBM Rational Rose hierarchy. A filter based on model types, pathname, partial filenames, and keywords differentiates SUCS from other activity diagrams.

Given the proprietary nature of many third-party case tools and the complexity that exists in the organization of artifacts within real-world data sets, it is essential to use an in-place traceability tool that is capable of establishing advanced filters and understands the nuances of specific third-party case tools. More general solutions such as enterprise-level search engines generally fail in this respect.

Creating traceable artifacts

In addition to the traceability environment, requirements that are well written and organized in meaningful ways also tend to return better traceability results than those that are haphazardly created. Several requirements best practices can significantly improve automated traceability results.

Use a well-defined project glossary. A well-constructed project glossary, defined during initial discovery meetings with stakeholders and used consistently throughout product development, will generally increase consistency in term usage and subsequently improve traceability.

In one of our experiments, we enhanced the basic probabilistic formula to weight terms found in the glos-

sary more highly than other types of terms.¹² In the IBS project, for which developers used glossary terms throughout the design process, this enhancement factor led to precision improvements from 20 to 25 percent at recall values of approximately 95 percent, and from 55 to 74 percent in the top 5 percent of the links—that is, the change pushed good links to the top of the candidate list. In contrast, glossaries created for other projects at the end of the development phase led to no improvement in precision. In related work, Huffman Hayes and colleagues also showed that using a project glossary could improve the quality of retrieved traces.⁷

Write quality requirements. Requirements should exhibit generally accepted qualities of a good specification such as being correct, unambiguous, complete, consistent, prioritized, verifiable, understandable, identifiable, and so on. The qualities of completeness and conciseness are specifically important for automated traceability. In fact, this could be considered a best practice for requirements in general, but it is typically only consistently attainable in an organization that provides requirements specification training.

Construct a meaningful hierarchy. Including a strong hierarchy of information, such as headings within a requirements document or meaningful package names, can enable a trace retrieval tool to strengthen the semantics of individual requirements and help construct more accurate links.

Poirt’s probabilistic formula incorporates hierarchical information as follows:

$$pr(a_j|q)=\left(\sum_{g\in pa_D(a_j)}\sum_i pr(a_j,g|t_i)pr(q,t_i)\right)\bigg/pr(q),$$

where $pa_D(a_j)$ represents the set of parents of a_j , that is, higher-level artifacts representing titles, subtitles, pack-

age names, and so on. The probability $pr(a_j, g|t_i)$ for any parent g of a_j is computed as

$$pr(a_j, g|t_i) \propto \frac{\text{freq}(a_j, t_i)}{\sum_k \text{freq}(a_j, t_k)} + \beta_D \frac{\text{freq}(g, t_i)}{\sum_k \text{freq}(g, t_k)}.$$

where the probability $pr(q, t_i)$ is defined as

$$pr(q, t_i) \propto \left[\text{freq}(q, t_i) + \beta_Q \sum_{h \in \text{an}_D(q)} pr(q|h) \text{freq}(h, t_i) \right] / n_i.$$

The probability term $pr(q|h)$ is computed as $pr(q|h) = 1/(M_{[q,h]} + 1)$, with $M_{[q,h]}$ being the distance of the ancestor from the query. The term $pr(q|h)$ can be regarded as a measure of the extent to which an ancestor document contributes information about a query q . A closer ancestor, such as a parent, is assumed to provide stronger information about q than more distant artifacts in the hierarchical topology.⁷

The optimal weights β_D and β_Q shown in these formulas were experimentally discovered as being 0.5 and 1, respectively. Experimental results showed that in certain data sets containing strong hierarchical information such as the IBS system, precision of trace results at a fixed recall value of 90 percent improved from approximately 26 to 31 percent when the retrieval process incorporated hierarchical data.

Bridge the intradomain semantic gap. As a single project typically includes various groups including customers, software developers, and systems engineers, stakeholders must often generate traces across domains containing artifacts developed using very different terminology to describe the same concepts. The semantic gap can be resolved by defining intradomain synonyms and utilizing a tool that can support domain-specific synonym matching. It is also essential to avoid reusing the same term to describe different concepts in different domains, as this will ultimately result in the generation of unnecessary links and decrease traceability result precision.

Create rich content. Elizabeth Hull, Ken Jackson, and Jeremy Dick¹³ described the value of incorporating rationales and domain knowledge into the traceability infrastructure, and this approach can be useful for supporting automated traceability.

For example, the infrastructure shown in Figure 6a includes no rationale information; the only common terms shared between user requirement UR21 and the three system requirements (SR15, SR32, and SR53) are the stop words “the,” “shall,” and “to,” and the word “vehicle,” which is a relatively common term in this particular application and would likely return numerous links at fairly low levels of confidence. In contrast, the satisfaction argument included in Figure 6b introduces the additional shared terms “weight,” “clearance,” and “power,” which are sufficient to support automated trace generation from the user requirement to each of the targeted system requirements.

The Siemens L&A data set for terse and rich SUCs illustrates the value of enhancing terse or hard-to-trace requirements with supporting rationale and domain information. This data set contained a total of 263 SUCs, including 221 terse ones containing an average of 5.3 words per requirement and 42 rich SUCs represented as activity diagrams. The highest achievable recall for the terse requirements was 58 percent with 4 percent precision, while the rich requirements achieved 90 percent recall with 31 percent precision. In the IBS, LC, and EBT data sets, which had average requirement word counts of 10.3, 17.0, and 13.3, respectively, requirements were traced successfully without the need for supporting data.

Introducing automated trace processes

Introducing new methods into an organization usually requires a planned process improvement initiative.

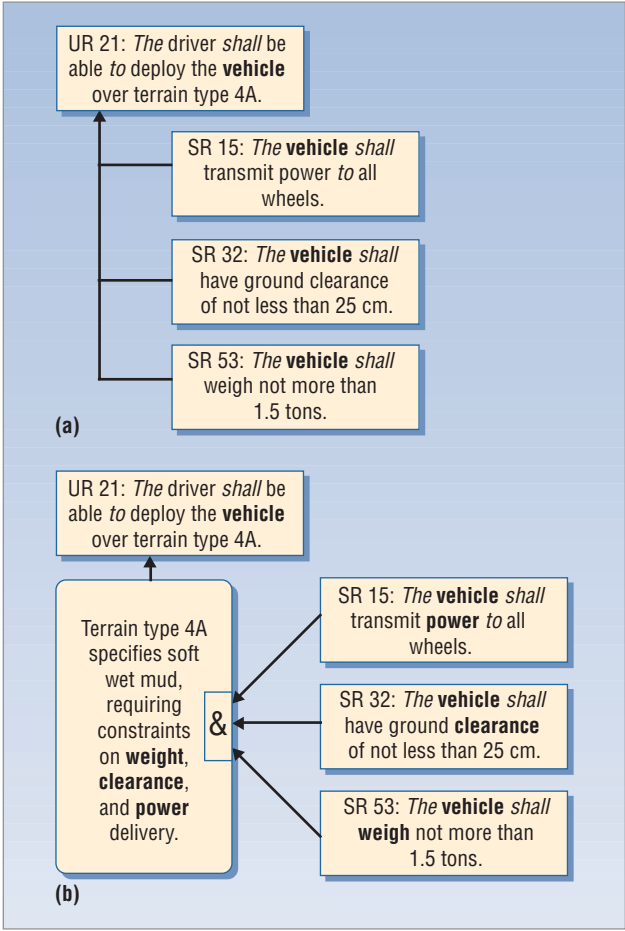


Figure 6. Rich trace support for automated traceability, as illustrated by Hull, Jackson, and Dick.¹³ (a) The only common terms shared between the user requirement and system requirements are meaningless stop words (italicized) and the word “vehicle”; automated traceability is not supported. (b) The introduction of additional shared terms (boldface) facilitates successful automated traceability between the system requirements and the user requirement. Figure adapted with permission.

This leads to the final best practice: *Use a process improvement plan.*

Three basic scenarios describe organizations that wish to adopt automated traceability methods.

The first scenario represents an organization with no traceability process currently in place. This organization desires to implement cost-effective traceability methods to more efficiently manage change and possibly meet more formal process-improvement initiatives. It might have large amounts of legacy data and no effective means for tracing it.

The second scenario represents an organization that has introduced traceability into a limited subset of artifacts such as high-level or critical requirements, code, and test cases but seeks to broaden its scope to include supplemental stakeholder documents without significantly increasing the effort involved.

The third scenario represents an organization that has traceability practices in place but is looking to replace them with more cost-effective and scalable methods.

Organizations can initiate a move to automated traceability by applying the best practices described here and implementing a pilot study designed to evaluate the traceability performance of a given data set. Experiments should be conducted to compare the results obtained from automatically created traces with manually created ones. Prior studies suggest that 20-35 percent precision should be achievable at recall levels of 90 percent. Where necessary, training and process improvement initiatives can improve results.

We derived these nine best practices from our experiences in applying automated traceability against several data sets. Researchers and practitioners interested in traceability continue to investigate automated methods to improve the performance of trace results and to provide increasingly sophisticated tool support.

The Center of Excellence for Software Traceability (www.traceabilitycenter.org), which is currently funded by the National Science Foundation, Siemens Corporate Research, and NASA, has been established to advance traceability research and to transfer traceability solutions to industry. Further information about automated traceability tools and other traceability support is available through the COEST Web site. ■

Acknowledgments

The work was partially funded by NSF grant CCR-0306303. Siemens Logistics and Automation in Grand Rapids, Mich., provided the Siemens data set. We also acknowledge the contributions of numerous DePaul University students who helped develop Poirot, especially Jun Lin, Rafal Habrat, Scott Bradley, Joseph Amaya,

Grace Bedford, Oussama BenKhadra, Mary Brophy, Chuan Duan, Chan Chou Lin, Andre Robertson, and Xuchang Zou.

References

1. O.C.Z. Gotel and A.C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem," *Proc. 1st IEEE Int'l Conf. Requirements Eng.*, IEEE CS Press, 1994, pp. 94-101.
2. IEEE, *IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications*, 1998; http://standards.ieee.org/reading/ieee/std_public/description/se/830-1998_desc.html.
3. B. Ramesh and M. Jarke, "Toward Reference Models for Requirements Traceability," *IEEE Trans. Software Eng.*, vol. 27, no. 1, 2001, pp. 58-92.
4. G. Antonio et al., "Recovering Traceability Links between Code and Documentation," *IEEE Trans. Software Eng.*, vol. 28, no. 10, 2002, pp. 970-983.
5. J.H. Hayes, A. Dekhtyar, and S.K. Sundaram, "Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods," *IEEE Trans. Software Eng.*, vol. 32, no. 1, 2006, pp. 4-19.
6. J. Cleland-Huang et al., "Goal-Centric Traceability for Managing Non-Functional Requirements," *Proc. 27th Int'l Conf. Software Eng.*, ACM Press, 2005, pp. 362-371.
7. J. Cleland-Huang et al., "Utilizing Supporting Evidence to Improve Dynamic Requirements Traceability," *Proc. 13th IEEE Int'l Conf. Requirements Eng.*, IEEE CS Press, 2005, pp. 135-144.
8. A. Marcus, J.I. Maletic, and A. Sergeyev, "Recovery of Traceability Links between Software Documentation and Source Code," *Int'l J. Software Eng. and Knowledge Eng.*, vol. 15, no. 4, 2005, pp. 811-836.
9. X. Zou et al., "Supporting Trace Evaluation with Confidence Scores," *Proc. 2005 Workshop Requirements Eng. Decision Support*, IEEE CS Press, 2005, pp. 1-7.
10. G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management*, vol. 24, no. 5, 1988, pp. 513-523.
11. A. Egyed et al., "A Value-Based Approach for Understanding Cost-Benefit Trade-Offs During Automated Software Traceability," *Proc. 3rd Int'l Workshop Traceability in Emerging Forms of Software Eng.*, ACM Press, 2005, pp. 2-7.
12. X. Zou, R. Settini, and J. Cleland-Huang, "Phrasing in Dynamic Requirements Trace Retrieval," *Proc. 30th Ann. Int'l Computer Software and Applications Conf.*, IEEE CS Press, 2006, pp. 265-272.
13. E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*, 2nd ed., Springer, 2004.

Jane Cleland-Huang is an assistant professor in the School of Computer Science, Telecommunications, and Information Systems at DePaul University's Center for Requirements Engineering as well as president-elect of the Center of Excellence for Software Traceability. Her research inter-

ests include requirements engineering, traceability, and software architectural design. Cleland-Huang received a PhD in computer science from the University of Illinois at Chicago. She is a member of the IEEE Computer Society and IEEE Women in Engineering. Contact her at jhuang@cs.depaul.edu.

Brian Berenbach is technical program manager for requirements engineering at Siemens Corporate Research, Princeton, New Jersey, with responsibilities for research, consulting, and corporate training. He has worked in the field of requirements engineering for more than 15 years and has been involved in requirements definition for a broad range of systems including medical, baggage-handling, mail-sorting, automated-warehouse, and embedded automotive systems. Berenbach received an MSc in physical chemistry from Emory University. He is a member of the IEEE Computer Society. Contact him at brian.berenbach@siemens.com.

Stephen Clark is an enterprise solution manager at iRise, an application definition software and services company based in El Segundo, California. His research interests include requirements traceability, product requirements

engineering methodologies, and organizational transformation through technology adoption. Clark is a member of the International Institute of Business Analysts. Contact him at StephenKClark@yahoo.com.

Raffaella Settimi is an assistant professor in the School of Computer Science, Telecommunications, and Information Systems at DePaul University. Her research interests include information-retrieval methods, Bayesian learning, statistical methods for knowledge discovering under uncertainty, and statistical genetics. Settimi received a PhD in statistics from the University of Perugia, Italy. She is a member of the American Statistical Association and the Royal Statistical Society. Contact her at rsettimi@cs.depaul.edu.

Eli Romanova is a master's student in the School of Computer Science, Telecommunications, and Information Systems at DePaul University's Center for Applied Requirements Engineering, and is currently interning in the Ultrasound Division at Siemens Medical Solutions, Mountain View, California. Her research interests include requirements engineering and human-computer interaction. Romanova received a BA in computer science from Hunter College. Contact her at eromanova@gmail.com.

Who sets computer industry standards?

802.11

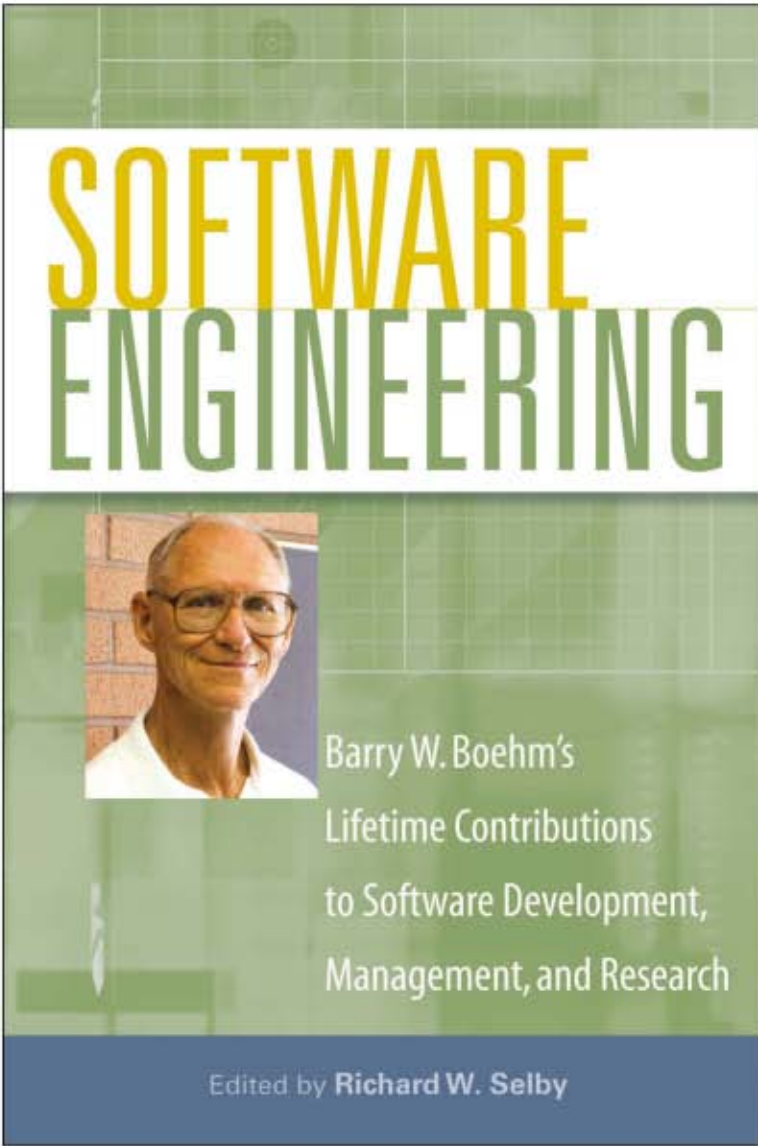
firewire

gigabit Ethernet

Together with the IEEE Computer Society, **you do.**

Join a standards working group at www.computer.org/standards/

FEATURED TITLE FROM WILEY AND CS PRESS



Software Engineering:
Barry W. Boehm's
Lifetime Contributions to
Software Development,
Management, and
Research

edited by
Richard W. Selby

978-0-470-14873-0
June 2007 • 832 pages
Hardcover • \$79.95
A Wiley-IEEE CS Press Publication

To Order:
North America
1-877-762-2974
Rest of the World
+ 44 (0) 1243 843294

This is the most authoritative archive of Barry Boehm's contributions to software engineering. Featuring 42 reprinted articles, along with an introduction and chapter summaries to provide context, it serves as a "how-to" reference manual for software engineering best practices. It provides convenient access to Boehm's landmark work on product development and management processes. The book concludes with an insightful look to the future by Dr. Boehm.



20% Promotion Code
CSCH7

COVER FEATURE

An Open Source Environment for Cell Broadband Engine System Software

Michael Gschwind, IBM T.J. Watson Research Center

David Erb, Sid Manning, and Mark Nutter, IBM Austin

The Cell Broadband Engine provides the first implementation of a chip multiprocessor with a significant number of general-purpose programmable cores targeting a broad set of workloads. Open source software played a critical role in the development of the Cell software stack.

Computer architects rarely introduce new architectures because incumbent architectures offer significant advantages due to tool maturity, programmer familiarity, and software availability. New architectures are usually a response to tectonic shifts in technology and market conditions. Thus, the original System/360 architecture was the first architecture to respond to mass production of systems. RISC systems corresponded to the introduction of VLSI manufacturing and the advent of single-chip microprocessors.

As the era of pure CMOS frequency scaling ends, architects must again respond to massive technological changes by more efficiently exploiting density scaling. The Cell Broadband Engine (Cell BE) answers these challenges by providing the first implementation of a chip multiprocessor with a significant number of general-purpose programmable cores targeting a broad set of workloads, including intensive multimedia and scientific processing.

Jointly developed beginning in 2000 by IBM, Sony, and Toshiba (STI) for the PlayStation 3 as well as other data-processing-intensive environments, Cell's design goal was to improve performance an order of magnitude over that of desktop systems shipping in 2005.¹⁻³ To meet that goal, designers had to optimize performance against area, power, volume, and cost in a manner not possible with legacy architectures. Thus, the design strategy was to exploit application parallelism through numerous cores that support established application models, thereby ensuring good programmability as well as programmer efficiency.⁴

The resulting Cell design is a heterogeneous, multicore chip capable of massive floating-point processing optimized for computation-intensive workloads and rich broadband media applications. As the "Cell BE Architecture Overview" sidebar describes, the design consists of one 64-bit Power processor element (PPE), eight accelerator processors called Synergistic Processor Elements (SPEs), a high-speed memory controller, a high-bandwidth element interconnect bus, and high-speed memory and I/O interfaces, all integrated on-chip.

SOFTWARE CHALLENGES

When we first outlined the Cell system's basic notions, we immediately realized that this revolutionary microprocessor design could substantially enhance application performance, but the task at hand was massive. Developing a new architecture has a set of risks that microprocessor design teams rarely face. Failure to verify that a new architecture responds to the needs that led to its conception, or to provide a satisfactory software stack to early adopters, usually will result in the failure of an architecture launch and its eventual demise.

In addition to the traditional challenge of defining a new microarchitecture, the design team faced the challenge of ensuring that the architecture can efficiently operate across a wide range of applications. Given the many innovations in Cell, it was important to provide early proof-of-concept to test and refine concepts that form the basis of the Cell BE Architecture (CBEA) as it is known today and its first implementation, the Cell Broadband Engine.

Cell BE Architecture Overview

We created the Cell Broadband Engine Architecture (CBEA) to address the needs of applications as they embrace chip multiprocessing. Rather than merely replicating a core multiple times on a chip, the Cell's heterogeneous architecture offers a mix of execution elements optimized for a spectrum of functions. Applications execute on this system, rather than a collection of individual cores, by partitioning the application and executing each component on the most appropriate execution element. While supporting different execution elements, the architecture also ensures efficient data sharing by providing a common system view of addressing, data types, and system functions across the heterogeneous execution elements. Based on this common system view, a Cell BE application process can consist of threads (lightweight processes) on both types of processor elements.

As Figure A shows, the Cell Broadband Engine, the first implementation of the CBEA,¹ includes a Power Architecture processor and eight attached processor elements. An internal high-performance element interconnect bus integrates the processor elements.

With a clock speed of 3.2 GHz, the Cell processor has a theoretical peak performance of 204.8 Gflop/s (single precision) and 14.6 Gflop/s (double precision). The

element interconnect bus supports a peak bandwidth of 204.8 Gbytes/s for intrachip data transfers, the memory interface controller provides a peak bandwidth of 25.6 Gbytes/s to main memory, and the I/O controller provides peak bandwidth of 25 Gbytes/s inbound and 35 Gbytes/s outbound.

Power Processor Element

The Power processor element (PPE) consists of a 64-bit, multithreaded Power Architecture processor with two concurrent hardware threads. The PPE supports the Power Architecture vector multimedia extensions to accelerate multimedia applications using SIMD execution units. The processor has a memory subsystem with separate first-level 32-Kbyte instruction and data caches, and a 512-Kbyte unified second-level cache. By using a Power Architecture processor as the base building block of the CBEA, we leveraged our decade-long experience with this mature and tuned architecture, as well as a stable software environment.

Synergistic Processor Element

The eight on-chip synergistic processor elements (SPEs) provide a significant portion of compute power in a Cell system.² An SPE consists of a new processor—the synergistic processor unit (SPU)—designed

to accelerate a wide range of workloads by providing an efficient data-parallel architecture and the synergistic memory flow controller (MFC), providing coherent data transfers to and from system memory.

The SPU cannot access main memory directly; the SPU obtains instructions and data from its 256-Kbyte local store and it must issue DMA commands to the MFC to bring data into the local store or write results back to main memory. In parallel to MFC data transfers, the SPU processes data stored in its private local store.

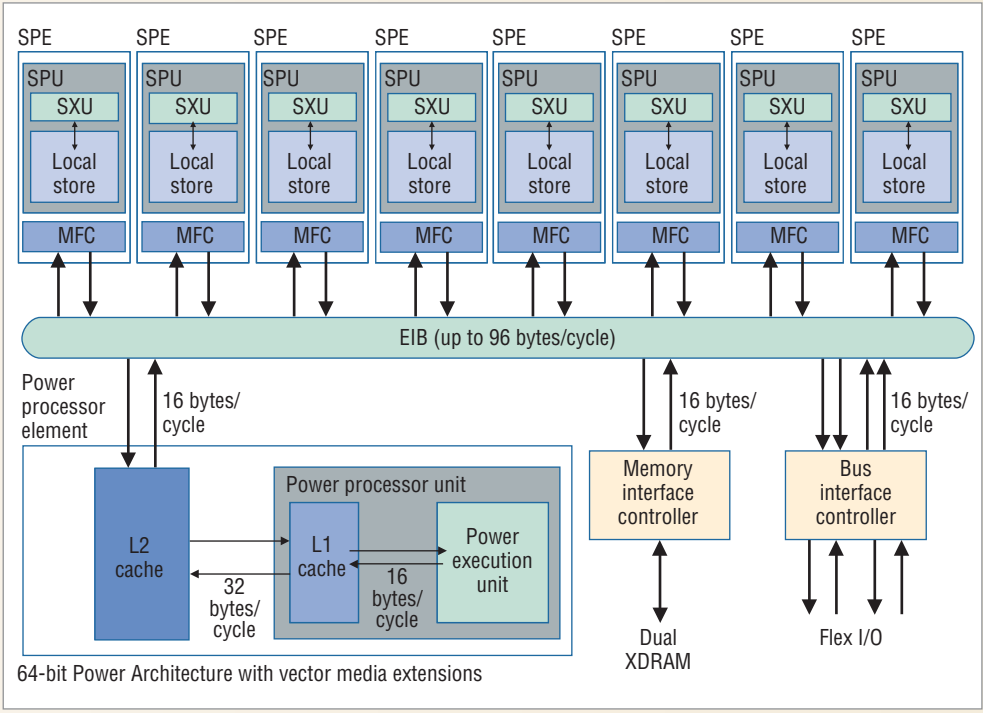


Figure A. Cell Broadband Engine system diagram. The system includes a Power Architecture processor and eight attached processor elements; an internal high-performance element interconnect bus integrates the processor elements.

The local store provides each SPU with private data access capability, guaranteed data availability, and deterministic access latency. The local store architecture offers logic simplicity, as cache-hit and coherence logic do not affect the critical memory access operations during load and store operations, allowing faster and more compact implementations. All data accesses with load and store operations refer directly to physical locations within an SPE's local store without further address translation.

Memory Flow Controller

To access global data shared between threads executing on the PPE and other SPEs, each SPE includes an MFC, which performs data transfers between SPU-local storage and system memory. The MFC provides the SPEs with access to system memory by supporting high-performance direct memory access (DMA) data transfer between the system memory and the local store. Data transfers can range in size from a single byte to 16-Kbyte blocks.

The MFC transfers copy between local store and system memory. An MFC transfer request specifies the local store location as the physical address in the local store. It specifies the system memory address as a Power Architecture virtual address, which the MFC's memory management logic translates to a physical address based on system-wide page tables that the Power Architecture specification provides.

Using the same virtual addresses to specify system memory locations independent of processor element type enables seamless data sharing between threads executing on both the PPE and SPE. An application executing on Cell can pass a PPE-generated pointer to code executing on the SPE and use it to specify the source or target in an MFC transfer request. Using full memory translation also ensures data protection between processes, as a thread can only access the system memory mapped into the associated process's virtual memory space.

Finally, using virtual addressing makes traditional operating system services such as demand paging available to SPE threads. When an SPE thread references paged-out memory via its associated MFC, the MFC's memory management unit generates a page-fault exception and delivers it to the PPE. The PPE then services the page fault on behalf of the SPE. When the page fault service has completed, the PPE restarts the MFC transfer that caused the page fault.

Memory Management

Multiple SPEs can share an address space with PPE threads in a Cell BE application, but at the same time other SPEs can reference different virtual memory spaces associated with respective applications executing concurrently in the system. To support this, each MFC includes a memory management unit (MMU) to provide address translation of system addresses in transfer requests. The MFC participates in the memory coherence protocols to ensure page table coherence.

Because each SPE contains an independent MMU, an SPE can execute independently from the PPE. However, the SPE is optimized for user-level data processing. Only the PPE performs privileged operations such as handling page faults, changing memory translation, and so forth, providing a centralized system control function. The Cell BE supports this by forwarding all exception-type events to the PPE via the on-chip interrupt controller.

Each MFC can be programmed to perform memory transfers either from the local SPU by placing commands in a 16-deep command queue using so-called SPU channel instructions or from remote nodes via memory-mapped I/O (MMIO). In addition to DMA transfers, the MFCs can also participate in the Power Architecture load-and-reserve and store-conditional lock synchronization and execute memory-synchronizing operations. Finally, the MFC supports list commands corresponding to an "MFC program" specifying a sequence of transfer requests.

Element Interconnect Bus

The element interconnect bus (EIB) provides high-bandwidth communication with a peak bandwidth of 204.8 Gbytes/s for intrachip data transfers among the PPE, the SPEs, and the memory and I/O interface controllers. The EIB has separate communication paths for commands (requests to transfer data to or from another element on the bus) and data. The EIB command path consists of a star-network to perform coherence actions. The EIB data network consists of four data rings—two rings running clockwise, two rings running counterclockwise.³

References

1. J. Kahle et al., "Introduction to the Cell Multiprocessor," *IBM J. Research and Development*, Sept. 2005, pp. 589-604.
2. M. Gschwind et al., "A Novel SIMD Architecture for the Cell Heterogeneous Chip Multiprocessor," *Hot Chips 17*, Aug. 2005.
3. S. Clark et al., "Cell Broadband Engine Interconnect and Memory Interface," *Hot Chips 17*, Aug. 2005.

DEVELOPING AN OPEN SOURCE STRATEGY

To succeed, modern technology solutions require rapid deployment in the marketplace. To address this challenge, the design team turned to open source software to accelerate the development of an ecosystem for the Cell architecture. Open source software allowed us to rapidly deploy an environment to be used both for architecture exploration and as an early adopter platform for the development of architecture verification suites, libraries, middleware, and sample applications.

The Cell open source software strategy had four phases:

- initial proof-of-concept focused on validating the design goals, compilation concepts, and programming paradigms developed in conjunction with the architecture definition;
- formative software phase supporting early adopter code for libraries, middleware, and applications;
- programming model innovation phase using a richer set of primitives, tools, and environments to explore the most efficient software development paradigms for the new platform; and
- transition to a full-fledged Cell ecosystem available to a steadily growing community of Cell developers via software development kit distributions. The Cell SDK is publicly available on IBM alphaWorks at www.alphaworks.ibm.com/tech/cellsw.

Open source also was used to provide an environment in which to deploy proprietary tools targeted at specific high-leverage points in the Cell BE software stack, such as autoparallelizing compilers based on the IBM proprietary XL C.⁴ While XL C provides a significant value proposition beyond open source tool suites, it integrates with open source assemblers, linkers, debuggers, and libraries in a seamless mixed environment.

Adopting open source allowed us to reduce the development cycle by leveraging a wide developer base with open source tool skills, leveraging tools designed for portability across platforms and providing early prototyping ability. During the exploratory phase, development occurred independent of the open source community at large, and we were able to make decisions based solely on the technology needs of the emerging architecture. Later, public distributions reflected changes made as part of the open source community adoption process and involved compromises to accommodate the cross-platform nature of the open source projects.

Open source tools were deployed in a proprietary execution environment, based on execution-driven simula-

tors for the SPU and a Cell BE full-system simulator based on Mambo.⁵

ANATOMY OF A CELL APPLICATION

A Cell application executes in a heterogeneous architecture consisting of PPE and SPE cores, respectively implementing the Power Architecture and Synergistic Processor Architecture. To match this mix of processor elements, a Cell application consists of two classes of instruction streams corresponding to the different architectures.

In the current software architecture model, each Cell application consists of a process that can have associated

PPE and SPE threads that are dispatched to the corresponding processors. When an application starts, the operating system initiates a single PPE thread, and control resides in the PPE. The PPE thread can then create further application threads executing on both the PPE and SPEs, supported by a thread management library based on the pthreads model.

SPE thread management includes additional functions, such as moving

a Cell application's SPE component into an SPE's local store, transferring application data to and from the local store, and initiating execution of a transferred executable at a specified start address as part of thread creation.

Once an application has initiated the SPE threads, execution can proceed independently and in parallel on PPE and SPE cores. While the PPE accesses memory directly using load and store instructions, application components executing on the SPE use the MFC to perform data transfers to the SPE local store before accessing application data with load and store instructions. The MFC is accessible from the PPE via a memory-mapped I/O interface and from the SPU via a channel interface.

The CBEA allows a variety of programming models, including an accelerator model based on a remote procedure call, function pipelines, and autonomous SPE execution. The simplest use of the SPE is the accelerator model where the PPE transfers the working set as part of the invocation and offloads a compute-intensive function onto one or more SPEs. Developers can also compose function pipelines where each SPE performs a set of functions on a data stream and then copies its output to the next pipeline stage implemented on another SPE via the MFC. Autonomous SPE execution occurs when the application starts an SPE thread, and the thread uses its MFC to independently transfer its input data set to the local storage and copy result data to the system memory.

In these programming models, the PPE typically uses its cache-based memory hierarchy to execute several control

The Cell team turned to open source software to accelerate the development of an ecosystem for the Cell architecture.

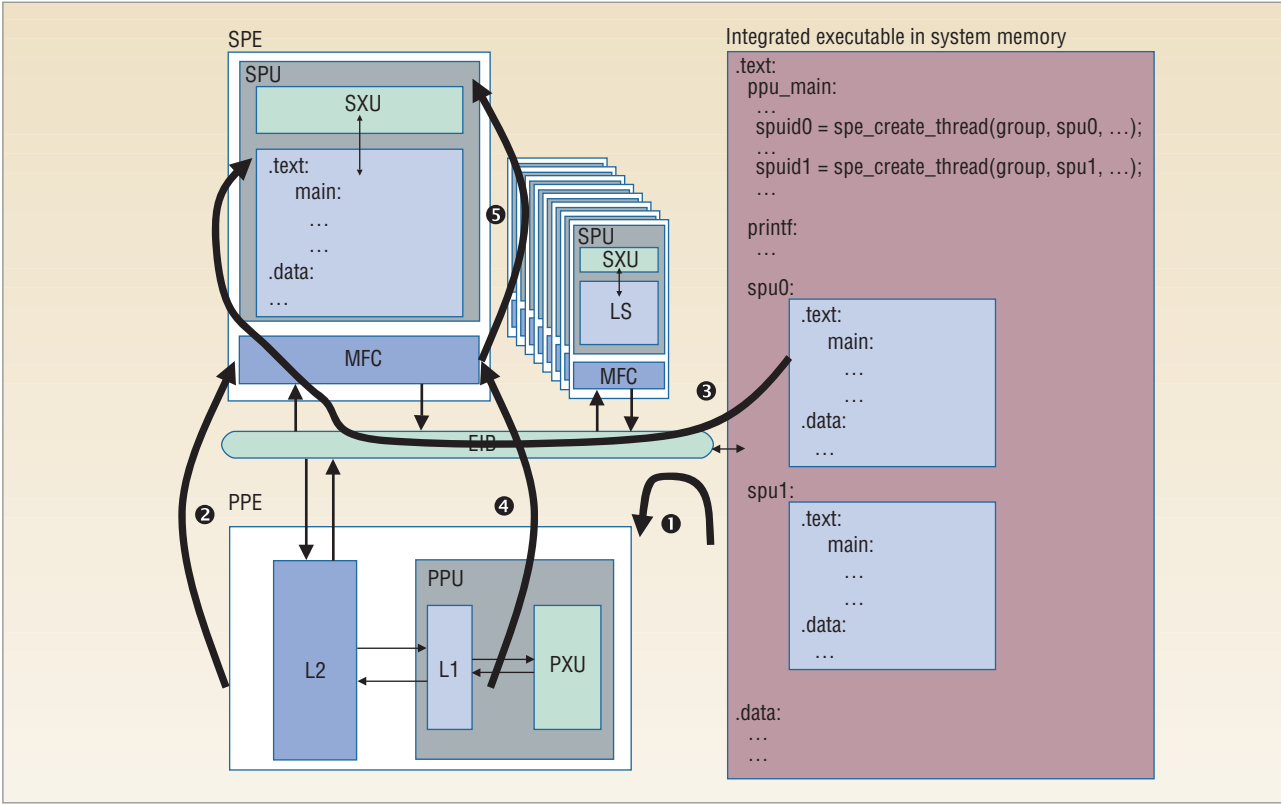


Figure 1. Execution start of an integrated Cell Broadband Engine application. (1) Power Architecture image loads and executes; (2) PPE thread initiates MFC transfer; (3) MFC data transfer occurs; (4) PPE instructs MFC to initiate SPU execution at specified address; and (5) MFC starts SPU execution.

functions, such as workload dispatch to multiple SPE data-processing threads, load balancing and partitioning functions, and a range of control-dominated application code.

Data-intensive processing

The SPE programming model is particularly optimized for the processing of data-processing-intensive applications, where the application transfers a block of data to the SPE local store and the SPU operates upon it. Computation results are stored back to the local store and eventually transferred back to system memory or directly to an I/O device by the MFC.

This processing model using SPEs to perform data-intensive regular operations is particularly well suited for media processing and numerically intensive data processing.⁶ Both the SPE and PPE offer data-parallel SIMD compute capabilities to further increase the processing performance of data-processing-intensive applications. While these facilities increase the data processing throughput potential of each processor element, the key is exploiting the 10 execution thread contexts on each Cell BE chip (two PPE threads and eight SPEs).

Data multibuffering

To hide the memory access latency to the slow external memory, data transfers are best performed using data

multibuffering (double buffering or even triple buffering). With double buffering, software pipelining is performed at the memory transfer level: The SPU operates on one data set in one data buffer, while the MFC transfers the next data set into another data buffer. Data multibuffering maps onto and exploits the compute-transfer parallelism in each SPE with its independent SPU execution and MFC data transfer threads.⁷

Application loading

Figure 1 illustrates application execution on the heterogeneous cores in the Cell BE. Initially, the image resides in external storage. The executable is stored in an object file format such as extensible linking format (ELF), consisting of text (read-only) and data (read/write) sections. In addition to instructions and read-only data, the text section also contains copies of one or more SPE execution images specifying the operation of one or more SPE threads.

To start the application, the operating system loads the Power Architecture object file, and (1) execution of the Power Architecture program thread begins. The application then initiates execution of application threads on the SPEs. To accomplish this, the application PPE must first transfer a thread execution image to an SPE's local store. (2) The PPE initiates a transfer of a

thread execution image by programming the MFC to perform a system memory-to-local storage block transfer, which is queued in the MFC command queues. (3) The MFC schedules the MFC request and performs a coherent data transfer.

The PPE can repeat these steps to transfer multiple additional memory-image segments containing either SPE application code, SPE libraries shared between threads, or SPE application data. When it has transferred the image, (4) the PPE issues an MFC request to start SPU execution. (5) The SPU starts execution at a specified address.

In addition to integrated executables consisting of PPE and SPE threads, Cell also can execute traditional unmodified Power Architecture executables for compatibility with industry-standard Power Architecture processors, as well as a new class of Synergistic Processor executables called *spulets*. A spulet is a Synergistic Processor Element-only program executing in a protected virtualized environment provided by the Power Architecture protection and translation model.

COMPILING FOR A Pervasively DATA-PARALLEL ARCHITECTURE

The first tool to provide any proof-of-concept prototyping capability for Cell systems, in particular the novel SPU architecture, was an execution-driven ISA simulator based on a preliminary architecture specification proposal. To simplify the development and prototyping flow, this simulator read assembly source code, and early library deployment occurred by loading multiple assembly source files.

The GNU C compiler (GCC) provided the first testing ground for the open source strategy and offered an early confirmation and proof-of-concept of many ideas introduced in the Cell BE. Before the final proposal was complete, we started development of a compiler based on GCC to demonstrate and explore the concepts introduced in the SPU—in particular, its SIMD-based architecture and the scalar layering used to implement a pervasively data-parallel computing architecture. This configuration also provided the first programming environment for library development and the first media-processing and encryption/decryption kernels that validated the newly defined architecture’s performance on these critical functions.

To implement a compiler showing the feasibility of concepts the SPU architecture introduced, we leveraged the entire GCC front end, including the Power Architecture SIMD extension interface, and rewrote a back end from the ground up to support this new computing concept. This allowed us to quickly support the entire semantics of the C language, its GCC exten-

sions, and the SIMD vector-programming intrinsic interface.

Scalar layering

One major concept of the Synergistic Processor Architecture that we needed to validate was scalar layering. Unlike prior architectures, the SPU architecture does not provide separate resources to support execution of scalar computations; instead, the compiler generates code sequences to compute scalar results with the SIMD data paths. We refer to an architecture using SIMD execution resources for scalar operations as a pervasively data-parallel computer architecture.

In the SPU architecture, all instructions take their operands from a unified 128-bit-wide vector register file with 128 architected registers. Compilers and programmers can use these instructions either to implement data-parallel SIMD operations or to produce scalar results by performing a wide result and using only the result returned in a single slot. To support scalar layering, instructions that use a single scalar input also read their operand from a 128-bit register and use the value from the “preferred slot,” the vector register’s first 32-bit vector element slot. This includes memory operations, which expect the memory address in the preferred slot, and branch instructions that can access a condition value or target address in the preferred slot.

In the SPU, all memory accesses operate on aligned quadwords, which must reside at addresses that are multiples of 16 bytes. To facilitate reading and writing of data values shorter than a quadword, the architecture supports efficient extract and merge operations, and memory accesses to retrieve an aligned quadword ignore the low-order four bits. Using a quadword-based memory interface simplifies the data-alignment logic and reduces operation latency. If the program is to perform access to a data value smaller than a quadword, the low-order bits indicate the data location within the quadword. The compiler expands such functionality and generates code to extract and format data explicitly using the simple SIMD RISC primitives that the architecture provides.

Although this alignment sequence requires several instructions, it reduces the overall data-flow latency because properly aligned scalar and vector data do not require alignment in most cases. For misaligned vector data, the compiler can optimize data-access patterns across loop iterations to generate more efficient alignment sequences. This new architectural concept eliminates the separate scalar execution units typically found in processors to support execution of scalar operations. Scalar layering reduces SPE area and design complexity

Using a quadword-based memory interface simplifies the data alignment logic and reduces operation latency.

and increases the number of SPEs that can be placed on a same-sized chip, which improves overall system performance.

Compiler prototype

By leveraging the GCC infrastructure, we could concentrate on developing compiler support for the novel SIMD RISC architecture features rather than undertaking the lengthy and costly process of developing an entire compiler from scratch. Using this compiler, we demonstrated the feasibility of generating appropriate sequences to implement data alignment in software instead of in hardware and demonstrated that hardware complexity reduction and efficient instruction scheduling result in an overall faster implementation.

The GCC also served as a vehicle to prototype an application binary interface (ABI) by experimenting with calling conventions and stack frame layouts and prototyping a first set of support libraries. The SPU ABI adopts the preferred-slot concept for passing scalar variables as function arguments and results and for allocating scalar variables in globally allocated registers as the default location for scalar data within a register file. Advanced compilers with intraprocedural optimization capabilities can optimize placement of scalar data in any slot.

To provide a consistent language interface for programmers between the PPE and SPE code, we adopted the same language interface to vector data types for the SPE as was already provided for the PPE. Similar to the Power Architecture vector specification, the SPU programming model also uses polymorphic intrinsics where the data type specifies the intrinsic operation—much as the operator “+” specifies either integer or floating-point operation based on its operands’ data type.

Seeding Cell application development

The development of the GCC-based SPU compiler proved the viability of the SPU architecture concept. Library and application developers adopted the compiler soon after it could compile the first programs and before full functionality became available. This had the desired effects of seeding a high-level-language (HLL)-based library and kernel development effort (which evolved into the SDK distribution), as well as giving valuable feedback from application developers to the Cell software and architecture teams.

By providing an early high-level development environment, the open source strategy also addressed a form of Clayton Christensen’s innovator’s dilemma⁸ by preventing the emergence of a tuned assembly code base. Invariably, such an assembly code base would have outperformed any nascent, unoptimized HLL

codes, drawing attention and efforts from the development of the HLL code, slowing or even completely forestalling development of the HLL library code. Using HLLs ultimately provides advantages in terms of programmer productivity and ease of adoption of new algorithms and data structures; thus, it delivers significant returns in performance or functionality.

Cell GCC became available in 2001, and we used it for all code development for the first two years until the XLC compiler became available. GCC-based compilers continue to be an important part of the Cell BE software ecosystem.

The first programming support specific to the Cell BE targeted the SPU to explore the new architecture.

HETEROGENEOUS ARCHITECTURE TOOLS

Supporting software development in a heterogeneous architecture represents a set of challenges surpassing traditional application build environments. Integrating tools across different architectures is key to allowing programmers to focus on

application development and ensuring their productivity. To address this need for a cohesive application development and build environment, we used a multipronged approach, reflecting the options available for different tools.

The initial tool environment started out hand in hand with the architecture definition work. A small team concentrated on developing key functionality and exploring the new architecture. The first programming support specific to the Cell BE targeted the SPU to explore the new architecture. Software and hardware development occurred in parallel, and we developed the SPU specification, compiler, and simulator infrastructure in parallel as we explored different design choices.

As the architecture evolved and the developers wrote longer programs, they needed a more robust development environment. We accomplished this by porting the GNU binutils to the SPU, providing a robust assembler, linkage, and binary manipulation utilities.

At the same time, integration between PPE and SPE to support advanced application development became more pressing. Ideally, this environment would provide a single, common interface for PPE and SPE program build with the ability to specify the target processor element on the command line. In a next step, a compiler would then automatically build Cell applications, partition the program into functions to be executed on the PPE and SPEs, respectively, and insert thread synchronization and data transfer as necessary for the correct execution of the program.

Integrated compilation

We defined the compiler to share a common vector programming model and support migration of applica-

tion source code between the different processor element types. Based on the common type system to represent vector data, we provided low-level intrinsics to access the specific architecture features of the two processor elements.

To compile an application for a Cell BE processor, portions of the program must be compiled specifically for each processor type. To accomplish this, compilers are provided for both processor element targets with separate executables for PPE and SPE, which are built from common source code. This makes traditional compiler optimizations and newly developed SIMD vectorization support available for both processor elements. To provide a common compilation interface for PPE and SPE, the compiler driver can invoke the proper executable for each target type based on a specified target architecture.

Building integrated executables

The GNU binutils provide a highly portable binary utilities tool chain with architecture versioning support. Thus, we chose to provide assembler and linker support for both PPE and SPE targets with a single binary. The linker generates object files in ELF format for both PPE and SPE. Finally, as Figure 1 shows, we developed an embedder program to build an integrated executable by including SPE executables in PPE executables, such that a thread executing on a PPE can initiate a thread executing the code the SPE binary specifies.

The embedder reads one or more fully compiled and linked SPE ELF binaries and embeds the SPE program in the integrated Cell executable in ELF format. The resulting PPE executable contains the PPE code, multiple embedded SPE executables, and management functions for transferring the SPE code to an SPE.

To embed an SPE executable in a PPE program, the embedder reads the fully linked SPE executable, extracts the memory image (both instruction and data), and generates C code containing data arrays corresponding to the memory image (data and text segment). It then invokes the PPE compiler to generate an object file with the data array holding the executable, which can be linked to PPE object files to give a single Power Architecture executable containing SPU object modules.

USING LINUX IN HETEROGENEOUS ARCHITECTURES

The Linux operating system played a central role in the STI development process. We based the initial port to the Cell BE on the Linux 2.4 kernel’s 64-bit Power Architecture distribution and bootstrapped it on the Mambo full system simulator long before the design was

finished. A key advantage of this approach was that it allowed exploration of heterogeneous execution models and evaluation of software support for proposed architecture functionality.

Porting Linux to the Cell BE involved addressing two important challenges. From a programming model perspective, we had to explore programming paradigms to enable applications to efficiently use the SPEs; from an operating system design perspective, the engineering challenge revolved around the dramatic break with the kernel’s expectations—namely, that each processor would be handling its own memory-mapping needs. While centralizing system management functions (such as virtual memory management) is one of the enablers of Cell’s efficiency, special consideration must be given to this aspect in porting legacy operating systems.

We experimented with several generations of SPE enablement in Linux to derive the most efficient and programmer-friendly model. From a programmability perspective, a key challenge was making SPEs easily accessible without imposing numerous constraints that would complicate application development. As we addressed these issues, we provided several experimental prototypes to early adopters to gather feedback. Based on real-world programming requirements and feedback from those developers, we evolved a generic and flexible SPE thread model. We based this model on the familiar pthreads concepts using the Linux 2.6.3 kernel source base and providing a heterogeneous lightweight thread model where a system call could spawn an SPU process, as Figure 2 shows.

Fault handling

From an operating system design perspective, a key challenge was to handle exceptions delivered on behalf of SPEs. This was a novel architectural mechanism, which had not been planned for in the internal Linux architecture. This model broke with traditional operating system kernels in one significant way: In normal symmetric multiprocessor system kernels, exceptions are associated with the currently scheduled process and can deliver only a single exception to the operating system at a time. In contrast, a Cell system could simultaneously deliver eight SPE exceptions to a single PPE, which also must handle its own PPE-related exceptions.

To address page-fault handling, we adopted an innovative deferred SPE exception approach in which the exception handler collects and preserves the relevant SPE fault information. A new deferred SPE page-fault handler then uses this information, executing in a kernel thread and implementing a Power Architecture-compliant page-fault handling routine—acquiring

We experimented with several generations of SPE enablement in Linux to derive the most efficient and programmer-friendly model.

spinlocks, sleeping, and so forth, as needed. Because the kernel thread executes the page-fault code at noninterrupt priority, it can spin on locks or sleep while waiting on a page transfer from external storage without causing deadlocks that might be introduced if multiple page-fault handlers were active simultaneously.

Thread management

To support a flexible SPE programming environment and provide a familiar programming abstraction, we created an SPE thread management API similar to the Posix pthreads library. This API supports both the creation and termination of SPE tasks and atomic update primitives for ensuring mutual exclusion. The API can access SPEs using a virtualized model wherein the OS dynamically assigns SPE threads to the first available SPE. This API completely virtualizes SPEs and the number of SPEs provided in a specific CBEA implementation or hypervisor-created partition. Optionally, applications can use a program-specified affinity mask to assign SPE threads to specific SPEs.

Interelement thread communication and synchronization architecture features (mailboxes, signal delivery, and so on) can be accessed either through a set of system calls or by allowing the user application to map an SPE’s memory-mapped control block into its application space. In the CBEA, the SPE control block actually consists of three separate control blocks corresponding to functions to be accessed by a user space application, an operating system, and a hypervisor. Using the user-accessible function control block, an application can perform direct MMIO operations between processor elements to communicate between SPEs and remote elements (either SPEs or the PPE) and avoid the overhead associated with system calls.

When the application requests creation of a thread, the SPE thread library requests the OS to allocate an SPE and creates SPE threads from SPE ELF object format files wrapped into an integrated Cell executable. To offload a portion of thread initialization onto the SPE, the PPE can use a “miniloader” executing on the SPE to perform SPE program loading. The miniloader, a 256-bit SPE program, downloads the application ELF segments from the host thread’s effective address space to the SPU local store. Using an SPE-side miniloader is advantageous because it offloads the PPE from having to pace program loads and it can use the SPE miniloader to preinitialize registers with application/OS parameter values.

This is attractive because multiple SPEs can load threads simultaneously, and SPEs have deeper fetch queues to hold multiple block transfer requests associ-

ated with loading a thread. In addition, communication within a processor element’s scope—that is, between the SPU and its associated MFC—is more efficient than interprocessor element communication between the MFC in an SPE and the PPE using MMIO.

Debugging integrated executables

The Cell BE requires an advanced debugging environment to allow developers to track applications executing on up to nine cores in a heterogeneous environment. Application developers working on a Cell BE application need to be able to follow the flow of control from one processor element to another processor element, from the PPE to a task spawned on the SPE, or from one SPE to the next.

The Cell debugging environment is built on the GNU debugger (GDB) and is the Cell debugging solution for both the GCC open source compiler and the IBM proprietary XL C compiler. The Cell debugging environment, however, goes far beyond a simple port of the

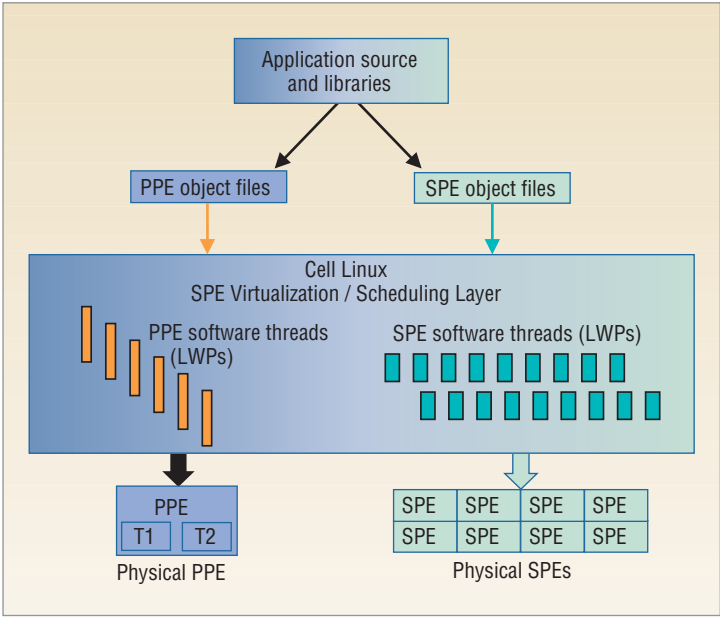


Figure 2. Application development and execution for a heterogeneous chip multiprocessor such as the Cell BE. An application program and libraries are partitioned into a set of functions executing on the PPE and SPE and compiled into object files for the PPE and SPE, respectively. The object files are then linked into an integrated executable (shown in Figure 1). The PPE object files contain code for several PPE software threads, and the SPE files contain code for several SPE software threads. When the application executes on a Cell-aware operating system (such as Cell Linux), it creates software threads using the thread library and the operating system services providing software threads (“lightweight processes” or LWPs) for the PPE and SPE. The operating system then maps the software threads to the available hardware threads in a Cell system. In the first implementation, each Cell BE chip offers two PPE hardware threads using hardware multithreading in the PPE core and eight single-threaded SPEs.

GDB debugging tool. To take advantage of the Cell BE's unique characteristics, the environment exploits additional system services to offer application debugging in a heterogeneous multicore architecture. When a Cell BE application spawns an SPE thread, GDB will follow that newly created SPE thread with the ability to properly interpret executables for the SPU architecture.

As both PPE and SPE debuggers are based on the common GDB source, PPE and SPE debuggers offer a consistent user interface. Initially, starting a thread instantiated a new processor-element-specific instance of the debugger; more recent versions support PPE and SPE debugging with a single heterogeneous debugger. Unless the developer selects an assembly language view of the program, the source-level debugger makes Cell's heterogeneous architecture completely transparent, allowing the developer to concentrate on the application behavior without regard to underlying instruction set architecture.

The Cell multicore debugging environment is based on several components:

- a GUI tracking multiple threads on the PPE and SPEs (an alternative text-based debugging environment is also available);
- GDB as the debugger engine, allowing developers to follow the execution of code across the PPE and SPEs, set breakpoints, and display data values stored in registers and memory; and
- debugging support in the system software stack that allows GDB sessions to gain control of a thread when it is initiated as well as interfaces to implement state inspection and modification.

The heterogeneous debugger architecture depends on support in the ABI—specifically, the thread creation interfaces provided in `libspe.a`, the SPU support library. Thus, all applications built with the standard Cell BE libraries automatically benefit from transparent heterogeneous debug support. To accomplish this, `libspe.a` and the dynamic library loader (`ld.so`) include support (during SPE thread creation) to allow `ppu-gdb` to obtain control at predictable points and retrieve information necessary to debug code in a newly created SPE thread. We have also included support for the debug environment in the SPU linker (`spu-ld`) by generating context information. This allows the debugger to find the symbol tables and other debugging information for each SPE thread when an application developer initiates an `spu-gdb` session.

The architecture, operating systems, and Cell system ABIs tightly integrate heterogeneous debug support. As an example, programmers can set arbitrary breakpoints in an SPU program at the source level. The GDB then translates this breakpoint into a location in the SPU local store and inserts an SPU “stopd” instruction. When the

SPU attempts to execute this instruction, the SPE delivers an interrupt to the PPE. In response to this interrupt, the kernel will perform a context save of the SPU thread state and send a SIGSTOP signal to the tracing process, allowing the debugger to take control when the application reaches a breakpoint.

The SPU GDB supports access to both the program state of user programs in the SPU and access to SPE state to provide a comprehensive view of application execution in a Cell system. In addition to SPU application state, this includes other SPE state corresponding to program-initiated operations such as mailbox communications, DMA transfers maintained in the MFC, and so forth.

We used open source software across the entire system stack to explore novel architecture concepts and their software enablement. We architected the software stack to present a high-level language programming environment abstracting specific architecture choices. The software environment allows application developers to focus on exploiting application parallelism to deliver the superior Cell performance as actual application performance. Using open source software has allowed accelerating architecture validation and debugging in a full-fledged software environment. In addition to being highly useful during the later stages of architecture definition and refinement, this approach also has provided an environment for early Cell adopters.

We have benefited—in real-world applications and in real time—from the feedback of Cell adopters in exploring programming abstractions for an integrated heterogeneous environment as pioneered by the Cell Broadband Engine Architecture. Many of the tools that formed the basis of the Cell BE infrastructure are still in use today, while others have served as a testbed and will coexist with commercial tools in a rich Cell software ecosystem. Adopting an open software strategy has allowed us to accelerate the market deployment of a new architecture offering innovations to improve efficiency and performance across the entire architecture stack by prototyping innovative software solutions while building on a familiar environment.

Finally, the Cell BE software environment allows application programmers to deliver high performance by focusing on applications, not the architecture or an unfamiliar tools environment. The true success of the Cell software environment is to allow the development of new, previously unseen applications for the Cell BE. ■

Acknowledgments

The authors thank Jim Kahle, Mike Day, and Ted Maeurer for their leadership and support. They also

thank Peter Hofstee, Ted Maeurer, Dan Prener, Valentina Salapura, and John-David Wellman for their many insightful comments and suggestions during the preparation of this article.

References

1. J. Kahle et al., "Introduction to the Cell Multiprocessor," *IBM J. Research and Development*, Sept. 2005, pp. 589-604.
2. M. Gschwind et al., "Synergistic Processing in Cell's Multi-core Architecture," *IEEE Micro*, Mar./Apr. 2006, pp. 10-24.
3. M. Kistler et al., "Cell Multiprocessor Communication Network: Built for Speed," *IEEE Micro*, May/June 2006, pp. 10-23.
4. A. Eichenberger et al., "Optimizing Compiler for the Cell Processor," *Proc. 14th Int'l Conf. Parallel Architectures and Compilation Techniques (PACT 2005)*, IEEE CS Press, 2005, pp. 161-172.
5. P. Bohrer et al., "Mambo—A Full System Simulator for the PowerPC Architecture," *ACM Sigmetrics Performance Evaluation Rev.*, Mar. 2004, pp. 8-12.
6. S. Williams et al., "The Potential of the Cell Processor for Scientific Computing," *Proc. ACM Computing Frontiers 2006*, ACM Press, May 2006, pp. 9-20.
7. M. Gschwind, "Chip Multiprocessing and the Cell Broadband Engine," *Proc. ACM Computing Frontiers 2006*, ACM Press, May 2006, pp. 1-8.
8. C. Christensen, *The Innovator's Dilemma*, McGraw-Hill, 1997.

Michael Gschwind is an architect and design lead for a future server system at IBM T.J. Watson Research Center, where he helped develop the Cell Broadband Engine Archi-

tecture concept and was a lead architect in defining the Synergistic Processor Architecture. He developed the first Cell BE compiler and helped initiate and contributed to the creation of the Cell software environment. His research interests include power-efficient high-performance computer architecture and compilation techniques. Gschwind received a PhD in computer science from Technische Universität Wien. He is an IBM Master Inventor and a senior member of the IEEE. Contact him at mkg@us.ibm.com.

David Erb is a developer currently working on Cell Ecosystem Development at IBM Austin. His research interests include performance tuning for multicore and SIMD systems. Erb received an MS in electrical engineering from the University of Texas at Austin. Contact him at djerb@us.ibm.com.

Sid Manning is an advisory software engineer working on Cell solution architectures and development at IBM Austin. His current interests are development tools for multicore architectures. He received a BS in computer science from the Rochester Institute of Technology. Contact him at sid@us.ibm.com.

Mark Nutter is a senior software engineer and IBM Master Inventor working for IBM's Systems & Technology Group. His responsibilities include adapting algorithms, programming models, and OS software to multicore processors such as the Cell Broadband Engine. His research interests include the development of an interactive ray tracer for the Cell BE and large model visualization. Nutter received a BS in computer science from the University of New Mexico. Contact him at mnutter@us.ibm.com.

IEEE Software Engineering Standards Support for the CMMI Project Planning Process Area

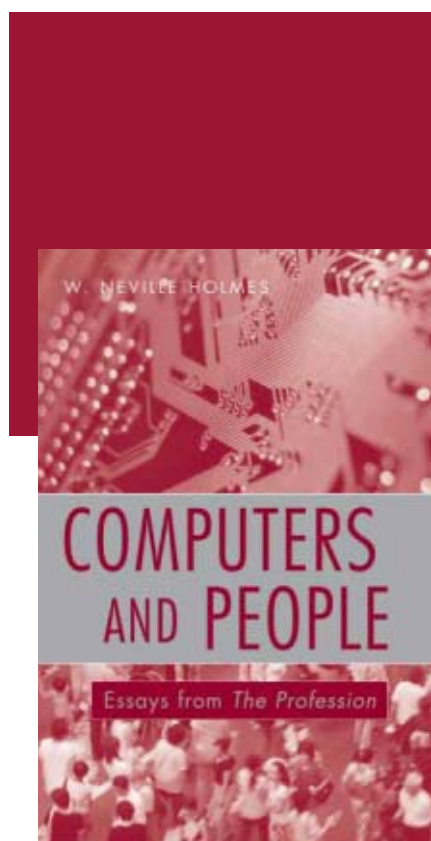
By Susan K. Land
Northrup Grumman

Software process definition, documentation, and improvement are integral parts of a software engineering organization. This ReadyNote gives engineers practical support for such work by analyzing the specific documentation requirements that support the CMMI Project Planning process area. \$19

www.computer.org/ReadyNotes

IEEE ReadyNotes





ISBN 0-470-00859-8

ISBN 978-0-470-00859-1

October 2006

\$47.50



On the Web

www.wiley.com/ieeecs

COMPUTERS AND PEOPLE

Essays from The Profession

W. NEVILLE HOLMES

In an intriguing collection of essays and overviews, W. Neville Holmes comments on the social and professional impact of computers. Each chapter covers a particular topic and contains a new, introductory essay, followed by selected essays from his column published in *Computer*, the IEEE Computer Society's flagship magazine. In a style meant to entertain and inform, these essays prod computing professionals to understand the broader context of their work.

This book strives to promote an understanding of how much more digital technology could benefit society and what type of professionals will help it fully realize that potential.

Computers and People has six main chapters:

- The Basis of Computing considers what technology is, and why digital technology is particularly significant in today's world.
- Computers So Far details how computers have been used and misused, questioning many common beliefs and practices.
- Computers and Education argues that the greatest potential benefit of digital technology is in education, and examines reasons why this potential is ignored in so many areas.
- Computing and Professions takes a close look at the nature and role of the computing profession, but brings out issues that are important to all professions.
- The Potential of Computing suggests several ways in which digital technology might be used to greatly benefit different areas of society.
- Facing the Future looks at the major problems facing the world and at how digital technology and computing professionals could help mitigate these problems.



IEEE
computer
society



Order Today

CALL North America: 1-877-762-2974
All others: +44 (0) 1243 779 777

FAX US: 1-800-597-3299
All others: +44 (0) 1243 843 296

MAIL John Wiley & Sons Inc.
Customer Care-Wiley
10475 Crosspoint Blvd.
Indianapolis, IN 46256

E-MAIL US: custserv@wiley.com
All others: cs-books@wiley.co.uk

COVER FEATURE

Isolation in Commodity Multicore Processors

Nidhi Aggarwal, University of Wisconsin-Madison

Parthasarathy Ranganathan and Norman P. Jouppi, Hewlett-Packard Laboratories

James E. Smith, University of Wisconsin-Madison

Resource sharing in modern chip multiprocessors (multicores) provides many cost and performance benefits. However, component sharing also creates drawbacks for fault, performance, and security isolation. Thus, integration of components on a multicore chip should also be accompanied by features that help isolate effects of faults, destructive performance interference, and security breaches.

Technology scaling and power trends have led to the widespread emergence of chip multiprocessors (CMPs) as the predominant hardware paradigm.¹ Multiple cores are being integrated on a single chip and made available for general-purpose computing. Intel and AMD manufacture dual-core processors and, more recently, quad-core processors. From a system viewpoint, CMPs provide higher levels of integration, typically including multiple processing cores, caches, memory controllers, and even some I/O processing—all in a single socket. The Sun Niagara processor, for example, includes eight cores, a shared second-level cache, and integrated memory controllers and I/O interfaces. The IBM Power5 dual-core processor has an on-chip memory controller.

Trends toward multiple cores will likely continue. Indeed, at a recent Intel Developer Forum, the company announced an aggressive roadmap of multicore processors with on-chip integration, including an 80-core prototype chip. AMD and other processor vendors have similar roadmaps. Further research in the academic community focuses on processors with a much larger number of cores,² as well as interesting variations in the design of multicore chips to include asymmetric and conjoined multicore processors.³

This scaling to include more cores allows for greater computational capability and system integration at the

chip level. In turn, this enables cost benefits from reduced component count. Additionally, enhanced resource sharing leads to better performance. On-chip components can now be easily shared to improve resource utilization, such as core sharing via hyper-threading, shared caches, and I/O interfaces. However, the same features of multicore processors that offer benefits can also present drawbacks. In particular, the increased levels of consolidation and integration lead to important isolation concerns—for performance, security, and fault tolerance.

Fault tolerance is an area of major concern. This is a particularly important issue given that recent studies have shown dramatic increases in the number of hardware errors when scaling technology to smaller feature sizes.⁴ Developers have encountered two main kinds of errors. First, defects in the silicon cause permanent or intermittent hardware faults, resulting in wear out over time and leading to *hard errors*. Second, electrical noise or external radiation can cause transient faults when, for example, alpha radiation from impurities or gamma radiation from outside changes random bits, leading to *soft errors*.

With CMPs, the fault-tolerance problem is compounded because a fault in any single component can lead to the failure of the entire chip. The *failure in time* (FIT) of cores, caches, memory, or I/O components combines to provide a high FIT for the CMP. Future CMP

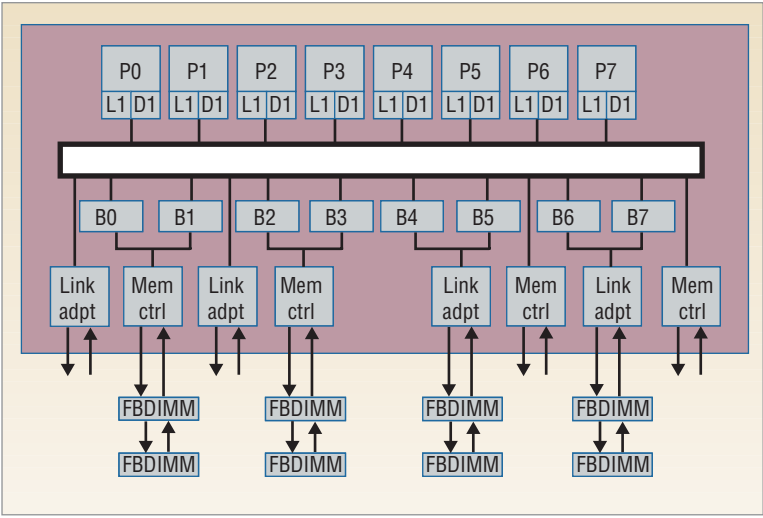


Figure 1. Conventional chip multiprocessor architecture. This CMP architecture has eight cores, P0 ... P7, each with private L1 caches; an eight-way banked, shared L2 cache, B0 ... B7; four memory controllers; and coherent links to other sockets or I/O hubs; FBD, IMM = fully buffered dual in-line memory module, Link adpt = link adapter, and Mem ctrl = memory controller.

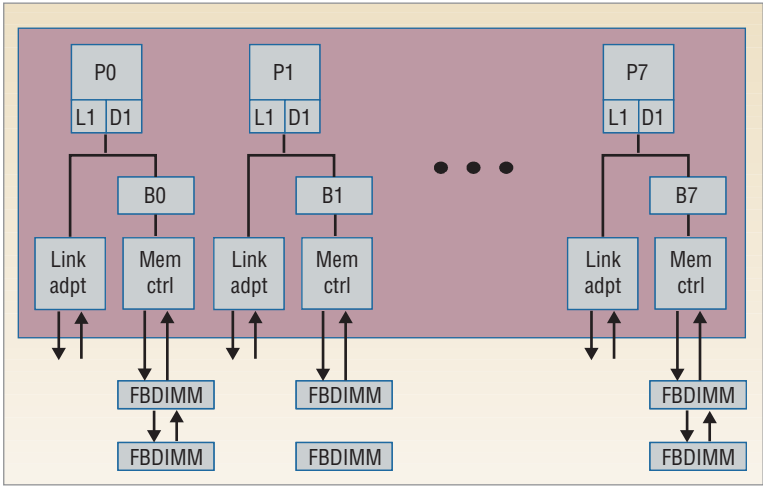


Figure 2. Static isolation. Independent computers are fabricated on the same die, and each computer has its own memory controller and I/O connections.

designs must offer the capability to isolate the faulty components and map them out so that the chip can be used with the remaining fault-free components.⁴

Figure 1 shows a conventional CMP architecture with eight cores, P0 ... P7, each with private L1 caches, an eight-way banked, shared L2 cache, four memory controllers, and coherent links—such as Hypertransport—to other CMP sockets or I/O hubs. In this architecture, a bidirectional ring connects the processors and cache banks, but other configurations with more complex 2D arrangements, such as meshes and interleaved layouts, are possible.

As the number of cores in a CMP increases geometrically with lithographic scaling, a failure in one part of

the conventional organization affects increasingly larger amounts of computational capability. For example, if the system shares all L2 cache banks and employs low-order address interleaving among the banks, a transient fault in the cache controller state machine can lead to an erroneous coherence state. Using error detecting codes on a coherence bit does not help in this case because the fault lies in the cache controller logic, before the coherence bits are set. Such a fault affects an entire chip's availability. Similarly, a fault in a memory controller, or anywhere in the ring interconnect, affects all the cores.

As the “Examining Current Commodity CMPs for Fault Isolation” sidebar describes, in the past when individual processors, memory controllers, and cache memory SRAMs provided the basic system building blocks, system designers could achieve good fault isolation by combining these chip-level components into redundant configurations at the board level. When necessary, designers can incorporate small amounts of application-specific integrated circuit (ASIC) glue logic. For example, the HP NonStop Advanced Architecture implements process pairs and fault-containment boundaries at the socket level.

With multicore approaches, however, socket-level isolation is no longer an attractive solution, and neither is using off-chip glue logic, especially for small systems. With growing numbers of cores at the socket level, implementing redundant configurations using different parts of a single multicore processor has become increasingly desirable. However, the lack of fault isolation in current multicore processors makes this impossible.

CHALLENGES

Static isolation, in which independent computers are fabricated on the same die, is a very straightforward approach to providing isolation in multicore processors. As Figure 2 shows, each computer has its own memory controller and I/O connections. This architecture has several disadvantages, however. The static partitioning of cache resources—which inhibit any interprocessor sharing—significantly reduces overall system performance and has not been used in proposed CMP designs. Similarly, static partitioning of chip interfaces and pins uses off-chip bandwidth inefficiently, making such a design unattractive for high-volume applications in which performance rather than high availability is the objective. Therefore, this offers a poor design choice for

balancing the tradeoffs between isolation and the benefits from shared resources.

The challenge therefore is to design techniques for configuring “off-the-shelf” CMPs with relatively little added on-chip hardware and complexity into high-availability, redundant systems. This can enable configuring

the levels of sharing dynamically, allowing isolation to be selectively turned on when needed.

CONFIGURABLE ISOLATION

We propose CMP implementations with *configurable isolation*—a set of techniques for dynamically config-

Examining Current Commodity CMPs for Fault Isolation

We analyzed the reliability and availability features of five commodity multicore architectures from key vendors: IBM’s Power5,^{1,2} AMD’s Opteron,³ Sun’s Niagara,⁴ and Intel’s Xeon⁵ and Montecito.⁶ Figures A shows the five commodity CMP architectures.

AMD’s Opteron 64-bit microprocessor has an on-chip memory controller and three HyperTransport links. The links connect an Opteron to other Opteron processors without additional chips. The Opteron has error-correcting codes (ECC) and protects large storage arrays like caches and memory. Hardware scrubbers are implemented for the L1 data cache, L2 cache tags, and DRAM, which supports chip kill ECC.

Sun Niagara is a CMP of multithreaded cores that supports 32 threads, with four threads per core. All the cores share a single floating-point unit. The memory system consists of an on-chip crossbar, L2 cache, and memory controllers. Each L2 bank connects to one memory controller. Niagara also supports ECC, chip kill, and memory scrubbing to protect against errors in the storage arrays. In addition, the chip has extensive support for per-thread trap detection and notification.

In the Northbridge, Intel Xeon-based 64-bit multiprocessors have multiple cores sharing a single external memory controller. The Xeon also supports ECC, parity, and scrubbing to protect the storage arrays.

The IBM Power5 is a dual-core, two-way SMT processor with an on-chip memory controller. Power5-based multiprocessors have extensive error checking and also include reliability features such as support for CPU sparing, chip kill, ECC, and parity for the memory hierarchy.

Intel Montecito is an Itanium-based dual-core and dual-threaded processor. Montecito provides parity protection for register files in addition to the ECC, scrubbing, and parity protection in the memory hierarchy. It also supports steering logic to isolate hard errors in the L3 cache lines.

Key Components

We divided each CMP into different components and then characterized whether they satisfied key requirements for fault tolerance: fault isolation, fault detection, and online repair. These three requirements are typically satisfied by employing redundancy.

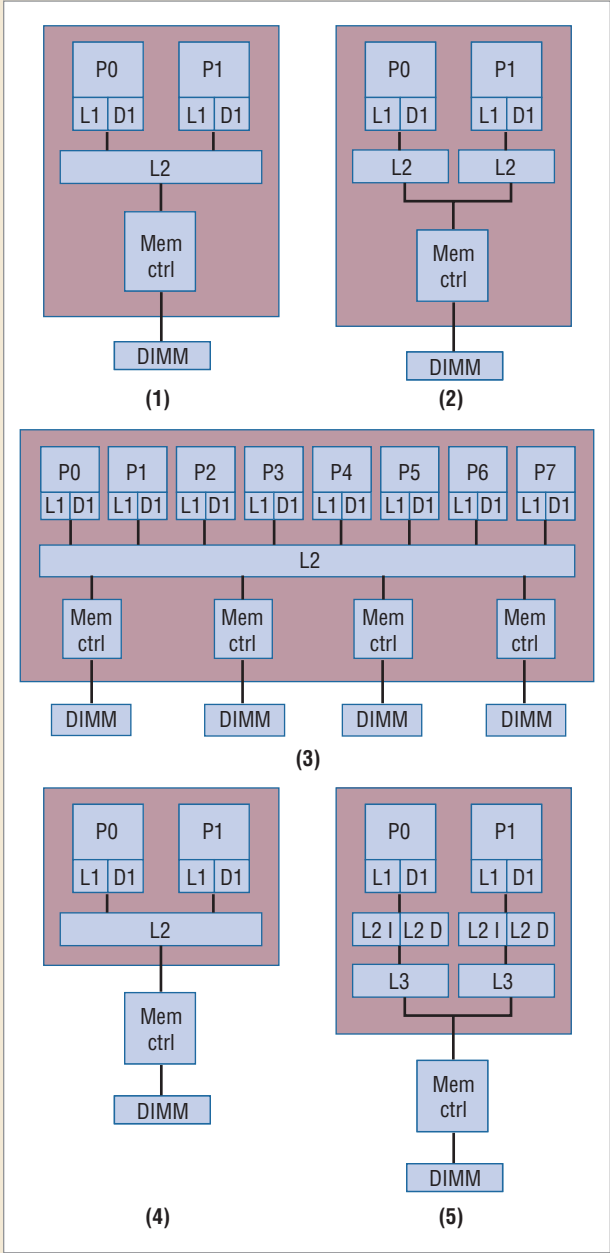


Figure A. High-level architectural block diagrams of the various commodity processors studied. Diagrams 1-5 correspondingly approximate the IBM Power5, AMD Opteron, Sun Niagara, Intel Xeon, and Intel Montecito.

Continued on the next page

Table A. IBM zSeries.				
Component	Redundancy	Fault isolation	Fault detection	Online repair
Core	8 spare processors (in 4 books)	Processors checkstops on failure	Mirrored pipeline, ECC, and parity with retry in register files	Dynamic core sparing, checkpoint at each instruction boundary, concurrent book add, checkpoint transplant to spare processor, separate register file for checkpoint
Cache	Active redundant L2 cache, redundant L2 rings	Special uncorrectable error codes	L1 - parity protected, L2 - ECC protected, XOR checking of control signals from L2 chips	Capability to add new cache at L2 ring interface, retry
Memory	Spare DRAM chips, redundant main storage controller, redundant store protect keys	Isolation of erroneous DRAM chip and store key	TMR for store keys, ECC, memory scrubbing, extra ECC code space, special uncorrectable error (UE) codes to indicate error source	Concurrent book add, chip kill, ECC
I/O	Redundant memory bus adapters (per book), I/O resources shared across all partitions	Single memory bus adapter clockstep design	Parity protection, command reject request, forced hang, special UE tag for known uncorrectable data	Operation retries, concurrent book add
System data and control buses	Redundant buses	Independent buses, immediate checkstop on control bus UE, regeneration of ECC across interfaces	Parity protected tag bit for uncorrectable data	Call for repair

Cores

Inside the core, currently transient fault detection is mainly restricted to the register file via parity or ECC. Montecito provides an exception with its built-in lockstep support and internal soft-error-checking capabilities. Opteron, Xeon, and Niagara have no fault isolation, so an error originating in any core can propagate to all other cores through the shared system components. Power5 and Montecito provide some degree of isolation for cores in different logical or electrical partitions, respectively.

In summary, all the commodity CMP architectures are vulnerable to soft errors, except Montecito in its lockstep configuration.

Caches

Most architectures are resilient to errors in the cache array and provide ECC or parity checking at all cache levels. However, Opteron and Xeon cannot tolerate errors that are not correctable by ECC alone, such as multibit errors. Niagara, Power5, and Montecito have more redundancy and fault isolation and can tolerate important classes of multibit errors. These CMPs usually share at least one level of the cache hierarchy, either across cores or contexts. However, none of the commodity CMPs can tolerate errors in the associated cache control circuitry.

Memory

Memory is perhaps the most fault-tolerant resource in commodity CMP systems. All the conditions for fault tolerance are satisfied in the memory arrays. This also reflects that historically memory is a system’s most error-prone component.

All the architectures have sophisticated techniques like chip kill, background scrubbing, and DIMM-sparing to tolerate failures. However, there is no tolerance to failures in memory access control circuitry. A failure in any memory controller or anywhere in the interconnect would affect all the cores. For example, in a design like the Xeon, an error in one memory controller in the shared Northbridge memory controller hub can affect multiple cores. On the other hand, in Opteron the failure of an on-chip memory controller can potentially be isolated to the cores in that chip.

Summary

Overall, we find that existing transient fault detection is limited to storage arrays such as register files, cache, and memory arrays. The lack of system-level fault isolation poses the biggest problem. Shared components do not have adequate fault isolation because a fault in one shared component can affect all cores on the chip. This

Table B. HP NonStop.				
Component	Redundancy	Fault isolation	Fault detection	Online repair
Core	Dual or triple modular redundancy	Isolated to a CPU	Compare results of I/O	Reintegration of new processing element, dedicated reintegration link
Cache	Dual or triple modular redundancy	Isolated to cache	Compare results of I/O	Replace processor slice
Memory	Dual or triple modular redundancy	Isolated to memory, no shared memory	Compare results of I/O, symmetric handling of interrupts for memory coherence across replicas	Replace processor slice
I/O	Dual redundant SAN	Independent fabrics	Self-checked circuits	Online replacement of logical synchronization units
System data and control buses	Redundant buses	Independent buses	CRC checksums	Replace processor slice

is true even if the system is running programs in a dual-modular redundant (DMR) or triple-modular redundant (TMR) configuration.

Comparing High-End, High-Availability Systems

We also examined two state-of-the-art systems, the IBM zSeries, shown in Table A, and the HP NonStop, shown in Table B. Enterprise-class applications that demand continuous availability use both of these systems.

The NonStop systems are DMR or TMR fault-tolerant servers built from standard HP four-way SMP Itanium server processor modules, memory boards, and power infrastructure. The processors communicate with each other and with shared I/O adapters through the ServerNet system area network (SAN). Each row of processors in a dual-mode or triple-mode redundant configuration forms one logical processor, which is made up of processor elements, one from each of the slices. The logical processor is the self-checked member of the cluster. Each processor element is a microprocessor running its own instruction stream and has a portion of the slice memory dedicated to its use. There are no synchronized clocks among the slices. The system compares all outputs from the servers at the I/O operation level (both IPC and device I/O) for 100 percent detection of faults. The voters themselves are self-checked.

The IBM zSeries servers incorporate extensive reliability, availability, and serviceability features to prevent both hard and soft errors. The zSeries pipeline is duplicated and each instruction checked before committing its results to an architected state. The servers have extensive redundancy in all components, including processors, L2 rings, L2

cache, and memory bus adapters. Most of the redundant components can be deployed dynamically, with no downtime, using techniques like Concurrent Book Add and Dynamic CPU Sparing. Wendy Bartlett and Lisa Spainhower provide an excellent discussion of the evolution of the NonStop and zSeries systems.⁷

References

1. D.C. Bossen et al., "Fault-Tolerant Design of the IBM Pseries 690 System Using Power4 Processor Technology," *IBM J. Research and Development*, vol. 46, no. 1, 2002, pp. 77-86.
2. "IBM Power5 Processor-Based Servers: A Highly Available Design for Business-Critical Applications," IBM white paper; www.ibm.com/systems/p/hardware/whitepapers/power5_ras.html.
3. C.N. Keltcher et al., "The AMD Opteron Processor for Multiprocessor Servers," *IEEE Micro*, vol. 23, no. 2, 2003, pp. 66-76.
4. P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded Sparc Processor," *IEEE Micro*, vol. 25, no. 2, 2005, pp. 21-29.
5. "Reliability, Availability, and Serviceability for the Always-on Enterprise," Intel white paper; www.intel.com/business/bss/products/server/ras.pdf.
6. C. McNairy and R. Bhatia, "Montecito: A Dual-Core, Dual-Thread Itanium Processor," *IEEE Micro*, vol. 25, no. 2, 2005, pp. 10-20.
7. W. Bartlett and L. Spainhower, "Commercial Fault Tolerance: A Tale of Two Systems," *IEEE Trans. Dependable and Secure Computing*, vol. 1, no. 1, 2004, pp. 87-96.

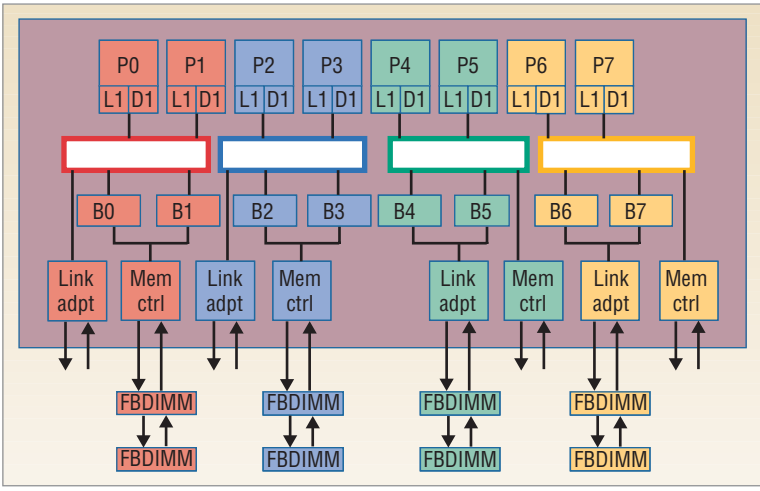


Figure 3. Configurable isolation. The introduction of low-cost configurable isolation at the interconnect, caches, and memory-controller levels provides a set of techniques for configuring the system with different isolation levels by controlling resource sharing.

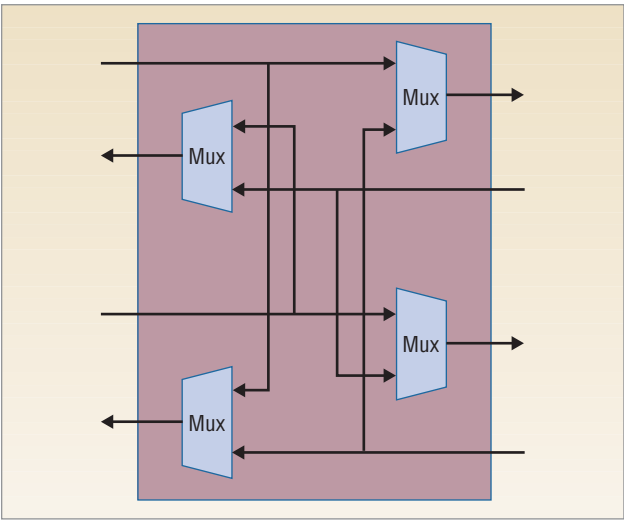


Figure 4. Ring configuration units. Physically, the ring forms the chip's central spine, so the cross-links should be less than a millimeter long. Their activation requires inserting a multiplexer at the input of a ring interface incoming data port.

uring the system with different isolation levels by controlling resource sharing. Figure 3 shows one such system.

The key difference between the system in Figure 3 and the one in Figure 1—the baseline architecture—is the introduction of low-cost configurable isolation at the interconnect, caches, and memory controller levels. For example, the ring interconnect in Figure 1 has been cut apart and reconfigured to create multiple logically independent rings using configuration crosslinks similar to the ring configuration units (RCU) shown in Figure 4. Physically, the ring is expected to form the chip's central spine, so the crosslinks should

be less than a millimeter long, and their activation requires inserting a multiplexer at the input of a ring interface incoming data port. The crosslinks and input multiplexers introduce a small additional fixed cost in terms of area and power, which does not significantly increase the design's cost for system configurations in which higher availability is not an objective.

As shown, an RCU can be implemented using multiplexers. Under software configuration control, the multiplexers can pass signals through to create a larger ring, or they can divide the larger ring into separate segments. The crosslinks are also expected to be shorter than the ring segments between cores, so the crossconnects should operate at least as fast as core-to-core or bank-to-bank ring segments.

Because the cross-links and input multiplexers are shared and can form a single point of failure, they must be implemented using self-checked logic if the design requires stringent fault tolerance. At the cache level, providing configurable isolation requires small changes to the ring and bank addressing. When the system software partitions the intercore interconnect, fewer address bits are required for interleaving among L2 cache banks within a single domain. Therefore, the L2 cache size available in a domain is inversely proportional to the number of domains.

Providing reconfiguration capabilities for cache banks and memory controllers requires the addition of two mode bits and extra tag bits. The first mode bit and one extra tag bit enable caching lines from the bank connected to the same memory controller. Another mode and two tag bits can enable caching lines from banks connected to a different memory controller. Overall, the number of extra bits required in a bank to enable caching of lines from any other bank in the system is \log_2 (number of banks).

This architecture offers the advantage that the system can be partitioned into separate *domains* on the fly, starting, for example, with Figure 1, and using configurable isolation to separate faulty domains from working domains. If there is a core fault, system software can isolate it within its domain and continue functioning with cores in the remaining, working domains. Further, the proposed architecture can continue functioning in the event of faults in cache banks, memory controllers, and the interconnection network. For example, if a fault occurs in a cache bank—say, B0 in Figure 1—then all the lines in that bank can be cached in the bank that shares the memory controller with the faulty bank—in this case B1. Should a memory controller fault occur, lines cached by both the B4 and B5 banks, connected with the memory controller,

can be cached by two other banks connected to a fault-free memory controller: B6 and B7. Similarly, link adapter and interconnect failures can be tolerated by isolating the faulty components and reconfiguring the system to use the remaining fault-free components.

The architecture in Figure 3 offers another advantage. Because system software can now divide the multicore processor into separate isolated domains, the separate domains can execute redundant copies of the same software to check for soft or transient errors. For example, Figure 5 shows how the system can be configured into two domains. The system employs resources from two domains to run dual-modular redundant (DMR) process pairs, with computations in the one domain (red in the figure) replicated in the second (green) domain when higher availability is required. In this design, self-checked voters compare the output of the redundant execution to detect errors.

For highest availability, voters can be implemented in I/O hubs connected to adapters from the redundant domains, similar to the hardware voters in the Nonstop Advanced Architecture.⁵ For lower-cost, lower-availability solutions, hypervisors that communicate between the redundant domains through I/O can implement the voter.⁶ Similarly, we could start with Figure 3 and use three isolated domains to enable a triple modular redundant (TMR) configuration. Further, the number of domains need not be static if the RCUs are self-checked, and they can be changed as system needs evolve.

BENEFITS

Configurable isolation in a CMP lets reconfiguration map out the faulty component and provides graceful performance degradation. We evaluated the impact of hard faults and subsequent reconfiguration on the system’s computing capacity over its lifetime by comparing three architectures:

- *Shared.* A completely shared system similar to proposed CMPs, as shown in Figure 1.
- *Static isolation.* A completely private system with full isolation, as shown in Figure 2.
- *Configurable isolation.* Our proposed architecture, with reconfiguration and configurable isolation, as shown in Figure 3.

Because the configurable-isolation architecture does not contain any modification to the cores, the size of the working set and its effect on cache behavior is the most important workload characteristic. Using SPEC benchmarks, we constructed three workloads with large,

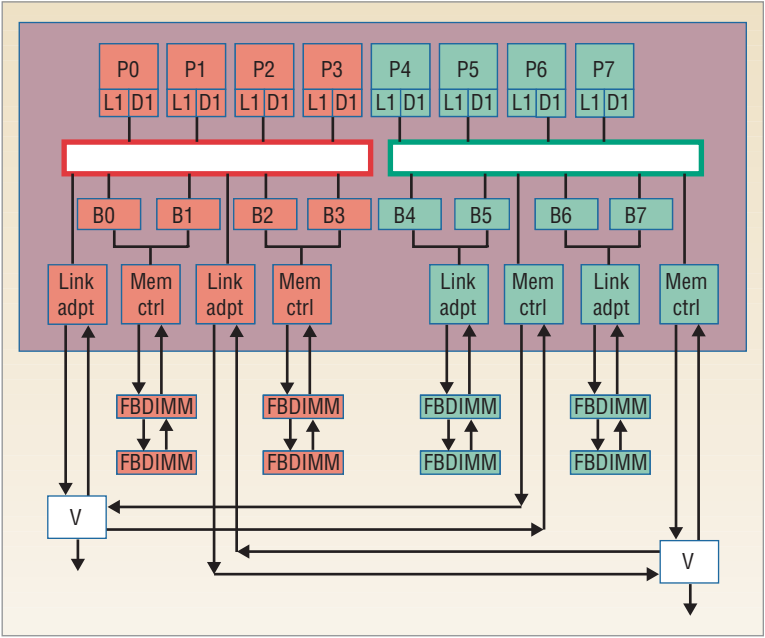


Figure 5. Dual fault domains. The system can be configured into two domains to run a dual-modular redundant process pair, with computations in the one domain (red) replicated in a second (green) domain when the system requires higher availability.

mixed, and small memory footprints. Over the course of a simulation run, as cores become unusable due to hard faults, benchmarks drop from the workloads, reflecting the loss of computing capability.

The fault model is based on state-of-the-art technology and derived from detailed and confidential micro-processor vendor models. We used HP-internal fault-analysis experiments to calibrate the fault model. The fault data includes FIT rates and distributions for hard and soft errors per component. We modeled five different regions that represent the granularity of reconfigurations: core and L1 cache, L2 circuitry, L2 banks, memory controller circuitry, and link controller.

On the shared system, any hard fault leads to system failure. This means that after a failure, such a system’s throughput drops to zero for all workloads. On the statically isolated system, any single fault leads only to the loss of throughput from the benchmark mapped to that private system. For example, even a fault in the bank associated with a core leads to that core being unusable. On a configurable isolated system, a fault—in a memory controller, for example—leads to loss of performance from the banks connected to the memory controller, but not the loss of a workload. Only when a core fails does a benchmark drop from the workload. Thus, the entire system becomes unusable in the configurable isolated architecture only when the last component of any type fails.

To make evaluation feasible, we used a two-phase methodology to simulate the performance of different processor configurations for various fault-arrival

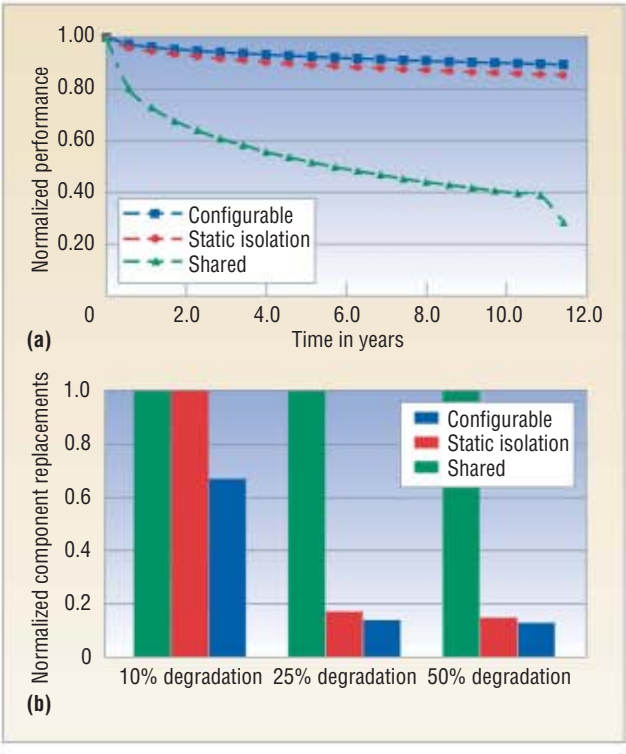


Figure 6. Evaluating the benefits of reconfiguration. Normalized performance from Monte Carlo hard-fault simulation over an 11-year period generated the results shown for three architectures—a baseline conventional system with full sharing; the proposed system, with configurable isolation; and a system with static isolation: (a) Performance over time and (b) normalized component replacements.

scenarios. First, using more than one machine-year, we ran a full-system simulator to exhaustively simulate the possible system configurations and compute the throughput of all configurations, subject to specific policies. Second, we performed a Monte Carlo simulation using a detailed component-level fault model. By running the Monte Carlo simulation for 10,000 runs, we simulated fault injection in a total of 10,000 systems, with each run comprising 100,000 simulated hours—approximately 11 years, as Figure 6 shows.

All simulations were done using a full system x86/x86-64 simulator based on AMD SimNow, which can boot an unmodified Windows or Linux OS and execute complex application programs. We used a timing model with a memory hierarchy similar to that supported by an AMD Opteron 280 processor, except with smaller L2 cache sizes to match the workloads’ working set.¹

As Figure 6a shows, performance degrades more gracefully with respect to hard faults in a system with configurable isolation. The average performance of the configurable isolation architecture degrades by less than 10 percent over 10 years. In contrast, the fully shared configuration degrades by almost 60 percent over the same period.

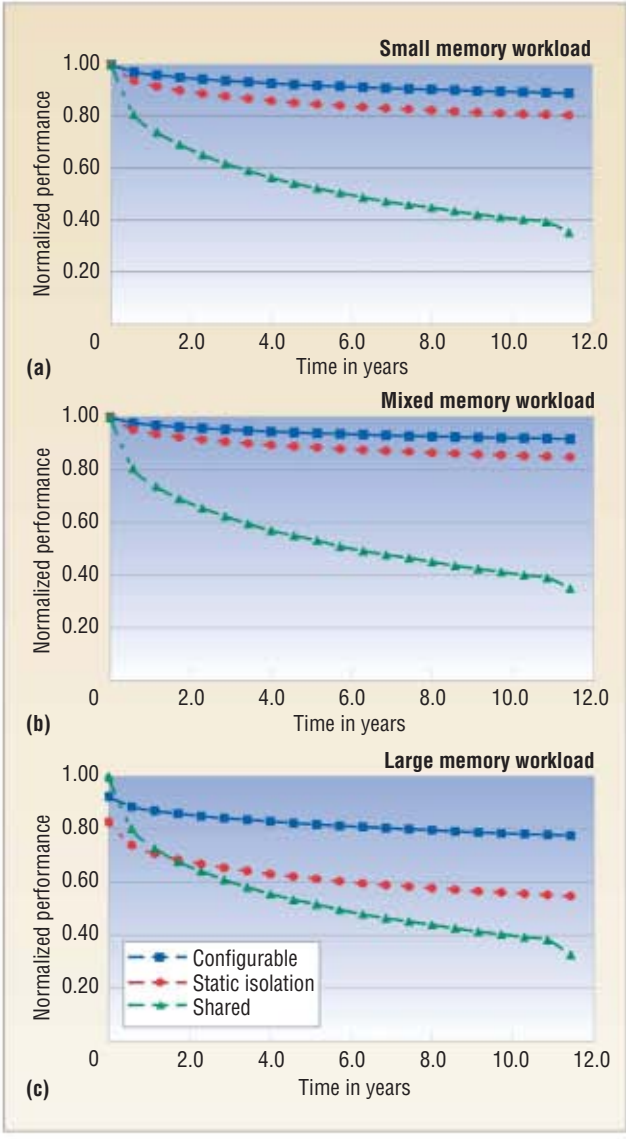


Figure 7. Three dual-modular redundant (DMR) configuration architectures: shared, statically isolated, and configurable isolation. The benefits of configurable isolation for providing graceful performance degradation in the event of hard faults (for DMR systems with transient fault protection) are shown across three memory workloads: (a) small, (b), mixed, and (c) large.

Figure 6b provides an alternate view of configurable isolation’s benefits, showing the number of component replacements for each of the three approaches. We assume that the system continues to stay operational until the performance dips below a certain threshold, after which the entire multicore component must be replaced and the performance reinitialized to that of the no-fault configuration. The simulation then continues for the remainder of the 100,000 hours with the new system.

We consider three cases in which the performance threshold is set to 90, 75, and 50 percent of initial performance. The total number of replacements across

Other Benefits from Isolation

In addition to fault isolation that enables more graceful degradation of performance in the presence of faults, isolation offers other benefits.

Power Reprovisioning

Isolation can lead to optimizations that are otherwise impossible. For example, with suitable fault-isolation support, the power budget could be dynamically reprovisioned by reassigning the power allotments of faulty components to the remaining fault-free components. Figure B shows the results assuming a future fault-model when a failed core's power budget can be reallocated dynamically to increase the clock frequency of the remaining cores, leading to improved system performance.

Trust and Performance

Other kinds of isolation include trust and performance. Consider, for example, scenarios in which workloads of different priorities compete for shared resources or have destructive interference—such as a background virus scanner running in parallel with an interactive user application. Support for isolation can ensure that the system partitions resources for the two workloads dynamically to avoid conflict. Similarly, in environments where the same computing platform hosts multiple workloads with different service-level agreements, configurable isolation can be used to partition resources to end users based on priority.

Recent studies describe the need for and benefits of performance isolation in a CMP.^{1,2} Kyle Nesbit and colleagues² propose a virtual private machine system that allocates a set of CMP resources—processors, bandwidth, and memory resources—to individual tasks. Virtual private machines isolate performance for coscheduled tasks in a CMP and ensure that the performance does not vary significantly regardless of the load placed on the system by other tasks.

Even if performance is not an issue, from a security and trust viewpoint, isolation could still prove useful to

10,000 Monte Carlo runs for a statically isolated and configurable isolated system is normalized with respect to the total number of replacements for a fully shared system. In such a system, every fault leads to system replacement because the performance drops to zero. These results show that the architecture with configurable isolation dramatically reduces the need to replace components irrespective of performance thresholds.

Figure 7 presents results for the three architectures when used in a DMR configuration. In this configura-

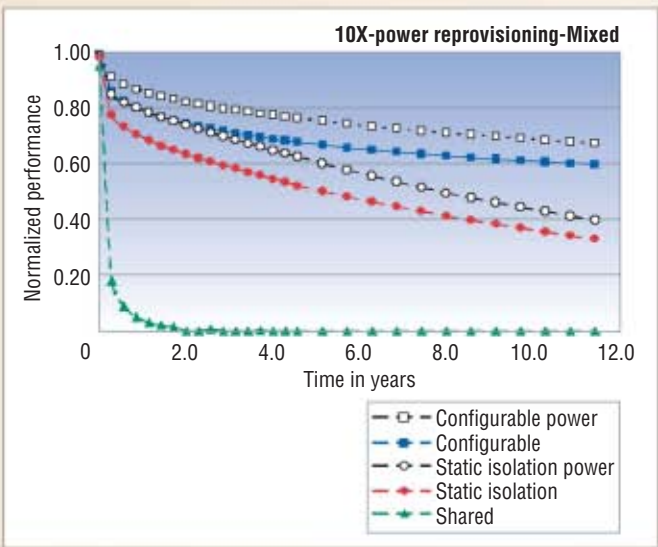


Figure B. Dynamic power reallocation with a future fault-model. System performance can be improved when a failed core's power budget is reallocated dynamically to increase the clock frequency of the remaining cores.

avoid malicious attacks from one user or application affecting other users or applications hosted on the same multicore. Dong Woo and Hsien-Hsin Lee³ suggest active monitoring to identify denial-of-service attacks and point out the challenges in differentiating attack scenarios from normal heavy-usage cases.

References

1. R. Iyer, "CQoS: A Framework for Enabling QoS in Shared Caches of CMP Platforms," *Proc. Int'l Conf. Supercomputing (ICS 04)*, ACM Press, 2004, pp. 257-266.
2. K.J. Nesbit, J. Laudon, and J.E. Smith, "Virtual Private Caches," *Proc. Int'l Symp. Computer Architecture (ISCA 07)*, IEEE CS Press, 2007, in press.
3. D.H. Woo and H.H. Lee, "Analyzing Performance Vulnerability Due to Resource Denial-of-Service Attack on Chip Multiprocessors," *Proc. Workshop Chip Multiprocessor Memory Systems and Interconnects*, 2007.

tion, a core failure would lead to loss of throughput from both copies of the benchmark. Since memory footprint affects performance significantly in this configuration, we present results for the large, mixed, and small memory workloads in Figures 7a, b, and c, respectively.

As expected, the shared system performs worst, with a dramatic degradation in average performance of 30 to 35 percent during the first two years, and degradation close to 50 percent by the end of five years. The statically isolated configuration is more resilient to failures and pro-

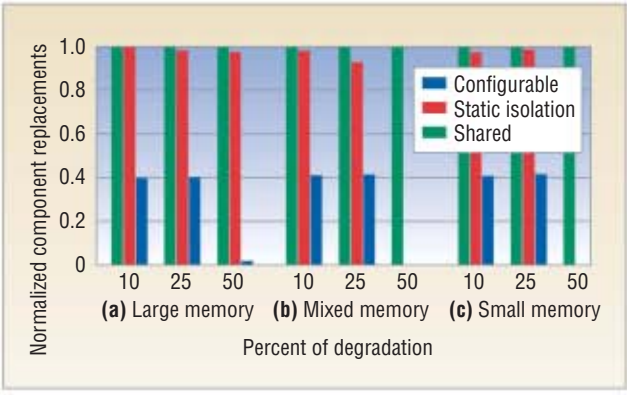


Figure 8. Number of normalized component replacements as a function of performance. When comparing the three architectures, assume components are replaced (a) when performance dips below 90 percent, (b) when performance dips below 75 percent, and (c) when performance dips below 50 percent.

vides more gradual performance degradation. Over five years, the net performance loss is only 10 to 15 percent.

The results for the large memory workload in Figure 7c are particularly interesting. Here, the isolated configurations (statically isolated and configurable isolated), by virtue of having private caches, initially underperform the shared configuration. However, compared to the fully shared system, the statically isolated system becomes performance competitive at around two years, the crossover point in the curves in Figure 7c.

Figure 8 presents the results for the number of component replacements required for the three architectures across all three workloads. The configurable isolation system consistently achieves the best performance across all workloads. With configurable isolation, resources can still be shared within a given fault domain. Additionally, dynamically repartitioning the resources leads to the most graceful degradation across all three workloads.

Several enhancements to the configurable isolation architecture provide additional benefits. For example, we assumed a single process per core. Overloading processes on remaining cores in a given working domain can potentially mitigate some of the performance degradation from losing a core in that domain. Similarly, when remapping fault domains, we assume arbitrary remapping of the fault domains and assignment of processes to cores. More advanced policies, aware of workload requirements and latency effects, could improve performance further. For example, prior work on heterogeneous multicore architectures demonstrates significant benefits from intelligently mapping workloads to available hardware resources.²

Configurable isolation also offers the ability to dynamically reconfigure the system’s availability guarantees. The approach we propose lets the system be configured to a spectrum of choices, from no-fault isolation to mul-

tiples smaller domains. For example, in utility-computing environments, a server can be provisioned as a payroll server with high levels of availability turned on, then it can be redeployed later as a Web server with lower availability levels. Isolation in CMP environments also has benefits beyond availability; the sidebar on “Other Benefits from Isolation” discusses some of these.

Multiple cores will provide unprecedented compute power on a single chip. However, integration of several components on a chip must be accompanied by features that enable isolation from fault effects, destructive performance interference, and security breaches. These features must ideally be low cost in terms of power and area and not impact the performance of the system adversely.

Here, we focus on isolation from faults. Future processors will be increasingly susceptible to hardware errors. The impact of errors on a conventional CMP with extensive sharing will likely be severe because the shared resources lack system-level fault isolation. Much of the recent architecture research in fault-tolerant systems has focused on tolerating errors originating in the core, such as DIVA,⁷ AR-SMT,⁸ chip-level redundantly threaded processor with recovery (CRTR),⁹ dynamic reliability management,¹⁰ total reliability using scalable servers,¹¹ and several others that use the extra cores or contexts available in a CMP. Other system-level recovery solutions for SMPs, such as NonStop⁵ and zSeries,¹² handle errors in the interconnection network and the cache coherence protocol, but they do not deal with the lack of fault isolation in CMPs.

For reliability at the system level, all components of the chip must be protected and faults must be isolated to smaller fault domains than the entire socket. Our design requires minimal hardware changes and retains the commodity economics and performance advantages of current CMPs. Further, we believe that there are exciting research opportunities in the area of enabling low-cost isolation features in CMPs that can enable them to be used as building blocks for high-performance, dependable, and secure systems. ■

References

1. This article is based on an earlier work: N. Aggarwal et al., “Configurable Isolation: Building High Availability Systems with Commodity Multicore Processors,” *Proc. Int’l Symp. Computer Architecture (ISCA 07)*, ACM Press, 2007; <http://doi.acm.org/10.1145/nnnnnnn.nnnnnnn>.
2. D. Patterson, “Recovery-Oriented Computing: A New Research Agenda for a New Century,” keynote address, *Proc. Int’l Symp. High-Performance Computer Architecture (HPCA 02)*, 2002; <http://roc.cs.berkeley.edu/talks/pdf/HPCAkeynote.pdf>.

3. R. Kumar et al., "Heterogeneous Chip Multiprocessors," *Computer*, Nov. 2005, pp. 32-38.
4. S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE Micro*, Nov. 2005, pp. 10-16.
5. D. Bernick et al., "NonStop Advanced Architecture," *Proc. Int'l Conf. Dependable Systems and Networks (DSN)*, IEEE CS Press, 2005, pp. 12-21.
6. T.C. Bressoud and F.B. Schneider, "Hypervisor-Based Fault Tolerance," *ACM Trans. Computer Systems*, Feb. 1996, pp. 80-107.
7. T.M. Austin, "DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design," *Proc. Int'l Symp. Microarchitecture (MICRO)*, IEEE CS Press, 1999, pp. 196-207.
8. E. Rotenberg, "AR-SMT: A Microarchitectural Approach to Fault Tolerance in Microprocessors," *Proc. Int'l Symp. Fault-Tolerant Computing*, IEEE CS Press, 1999, pp. 84-91.
9. M. Goma et al., "Transient-Fault Recovery for Chip Multiprocessors," *Proc. Int'l Symp. Computer Architecture (ISCA)*, IEEE CS Press, 2003, pp. 98-109.
10. J. Srinivasan et al., "The Case for Lifetime Reliability-Aware Microprocessors," *Proc. Int'l Symp. Computer Architecture (ISCA 04)*, IEEE CS Press, 2004, pp. 276-287.
11. J.C. Smolens et al., "Fingerprinting: Bounding Soft-Error Detection Latency and Bandwidth," *Proc. Int'l Conf. Architecture Support for Programming Languages and Operating Systems (ASPLOS)*, ACM Press, 2004, pp. 224-234.
12. M.L. Fair et al., "Reliability, Availability, and Serviceability (RAS) of the IBM eServer z990," *IBM J. Research and Development*, Nov. 2004, pp. 519-534.

Nidhi Aggarwal is a doctoral student in the Computer Sciences Department at the University of Wisconsin-Madison. Her research interests include high-performance and high-availability systems, virtual machines, and memory system design. Aggarwal received an MS from the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison. Contact her at naggarwal@wisc.edu.

Parthasarathy Ranganathan is a principal research scientist at Hewlett-Packard Laboratories. His research interests include low-power design, system architecture, and parallel computing. Ranganathan received a PhD in electrical and computer engineering from Rice University. Contact him at partha.ranganathan@hp.com.

Norman P. Jouppi is a Fellow and Director of the Advanced Architecture Lab at Hewlett-Packard Laboratories. His research interests include highly parallel systems, high-performance networking, and the impact of photonics on computer systems architecture. Jouppi received a PhD in electrical engineering from Stanford University. Contact him at norm.jouppi@hp.com.

James E. Smith is a professor in the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison. His research interests include high-performance processors and systems. Smith received a PhD in computer science from the University of Illinois. Contact him at jes@ece.wisc.edu.

Giving You the Edge

IT Professional magazine gives builders and managers of enterprise systems the "how to" and "what for" articles at your fingertips, so you can delve into and fully understand issues surrounding:

- Enterprise architecture and standards
- Information systems
- Network management
- Programming languages
- Project management
- Training and education
- Web systems
- Wireless applications
- And much, much more ...

IT Professional

www.computer.org/itpro



RESEARCH FEATURE

iMouse: An Integrated Mobile Surveillance and Wireless Sensor System

Yu-Chee Tseng, You-Chiun Wang, Kai-Yang Cheng, and Yao-Yu Hsieh
National Chiao Tung University

Incorporating the environment-sensing capability of wireless sensor networks into video-based surveillance systems can provide advanced services at a lower cost than traditional surveillance systems. The integrated mobile surveillance and wireless sensor system (iMouse) uses static and mobile wireless sensors to detect and then analyze unusual events in the environment.

The remarkable advances of microsensoring micro-electromechanical systems (MEMS) and wireless communication technologies have promoted the development of wireless sensor networks. A WSN consists of many sensor nodes densely deployed in a field, each able to collect environmental information and together able to support multihop ad hoc routing. WSNs provide an inexpensive and convenient way to monitor physical environments. With their environment-sensing capability, WSNs can enrich human life in applications such as healthcare, building monitoring, and home security.

Traditional surveillance systems typically collect a large volume of videos from wallboard cameras, which require huge computation or manpower to analyze. Integrating WSNs' sensing capability into these systems can reduce such overhead while providing more advanced, context-rich services. For example, in a security application, when the system detects an intruder, it can conduct in-depth analyses to identify the possible source. The "Related Work in Wireless Surveillance" sidebar provides additional information about other work in this area.

Our *integrated mobile surveillance and wireless sensor system* (iMouse) consists of numerous static wireless sensors and several more powerful mobile sensors. The benefits of iMouse include the following:

- It provides online real-time monitoring. For example, when the system is capturing events, the static sensors can immediately inform users where the events are occurring, and the mobile sensors can later provide detailed images of these events.
- It's event-driven, in the sense that only when an event occurs is a mobile sensor dispatched to capture images of that event. Thus, iMouse can avoid recording unnecessary images when nothing happens.
- The more expensive mobile sensors are dispatched to the event locations. They don't need to cover the whole sensing field, so only a small number of them are required.
- It's both modular and scalable. Adding more sophisticated devices to the mobile sensors can strengthen their sensing capability without substituting existing static sensors.

Because mobile sensors run on batteries, extending their lifetime is an important issue. We thus propose a dispatch problem that addresses how to schedule mobile sensors to visit emergency sites to conserve their energy as much as possible. We show that if the number of emergency sites is no larger than the number of mobile sensors, we can transform the problem to a maximum matching problem in a bipartite graph; otherwise, we group emergency sites into clusters

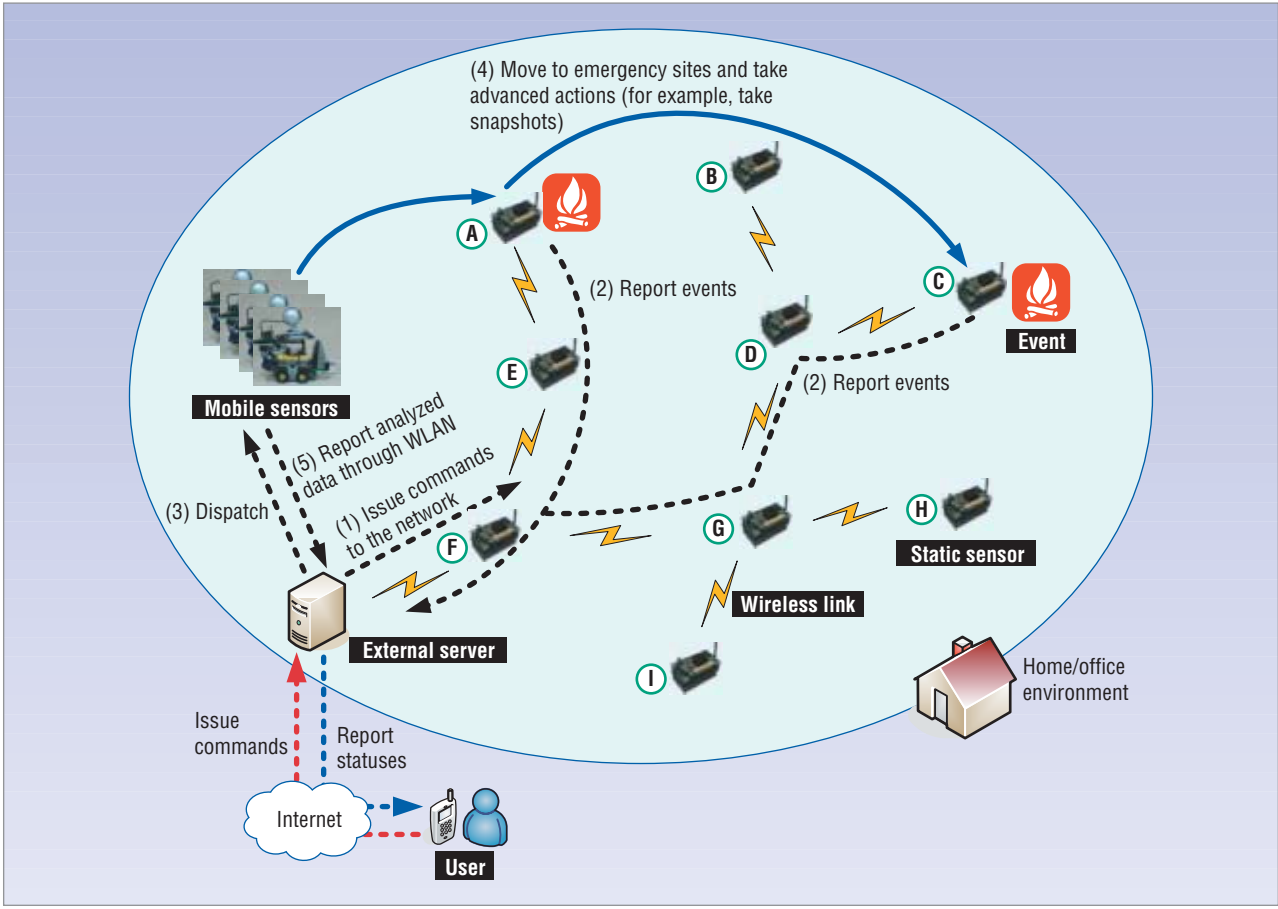


Figure 1. The iMouse system architecture. Three main components make up the iMouse architecture: static sensors, mobile sensors, and an external server. The user issues commands to the network through the server (1). Static sensors monitor the environment and report events (2). When notified of an unusual event, the server notifies the user and dispatches mobile sensors (3), which move to the emergency sites, collect data, (4) and report back to the server (5).

so that one mobile sensor can efficiently visit each cluster.

SYSTEM DESIGN

Figure 1 shows the iMouse architecture, which consists of static and mobile sensors and an external server. The static sensors form a WSN to monitor the environment and notify the server of unusual events. Each static sensor comprises a sensing board and a mote for communication. In our current prototype, the sensing board can collect three types of data: light, sound, and temperature. We assume that the sensors are in known locations, which users can establish through manual setting, GPS, or any localization schemes.¹

An event occurs when the sensory input is higher or lower than a predefined threshold. Sensors can combine inputs to define a new event. For example, a sensor can interpret a combination of light and temperature readings as a potential fire emergency. To detect an explosion, a sensor can use a combination of temperature and sound readings. Or, for home security, it can use an unusual sound or light reading.

To conserve static sensors' energy, event reporting is reactive.

Mobile sensors can move to event locations, exchange messages with other sensors, take snapshots of event scenes, and transmit images to the server. As Figure 2 shows, each mobile sensor is equipped with a Stargate processing board (www.xbow.com/Products/productsdetails.aspx?sid=229), which is connected to the following:

- a Lego car (<http://mindstorms.lego.com/eng/default.asp>), to support mobility;
- a mote, to communicate with the static sensors;
- a webcam, to take snapshots; and
- an IEEE 802.11 WLAN card, to support high-speed, long-distance communications, such as transmitting images.

The Stargate controls the movement of the Lego car and the webcam.

The external server provides an interface through which users can obtain the system status and issue commands. It also maintains the network and interprets the

meanings of events from sensors. On detecting a potential emergency, the server dispatches mobile sensors to visit emergency sites to obtain high-resolution images of the scene. The dispatch algorithm also runs on the server.

System operations and control flows

To illustrate how iMouse works, we use a fire emergency scenario, as Figure 1 shows.

On receiving the server’s command, the static sensors form a treelike network to collect sensing data. Suppose static sensors A and C report unusually high tempera-

tures, which the server suspects to indicate a fire emergency in the sensors’ neighborhoods.

The server notifies the users and dispatches mobile sensors to visit the sites. On visiting A and C, the mobile sensors take snapshots and perform in-depth analyses. For example, the reported images might indicate the fire’s source or identify inflammable material in the vicinity and locate people left in the building.

Each static sensor runs the algorithm in Figure 3. The server periodically floods a tree-maintenance message to maintain the WSN. It also records each static sensor’s location and state, which is initially set to normal. Tree-maintenance messages help the static sensors track their parent nodes. To distinguish new from old messages, tree-maintenance messages are associated with unique sequence numbers. The goal is to form a spanning tree in the WSN.

When a sensor receives an input above a threshold, indicating an event, the sensor reports that event to the server. To avoid sending duplicate messages, each sensor keeps a variable event flag to indicate whether it has already reported that event. When a sensor detects an event and the event flag is false, the sensor reports that event and sets the flag to true. The server collects multiple events and assigns them to mobile sensors in batches. When a mobile sensor visits an event site, it asks the local sensor to clear its event flag.

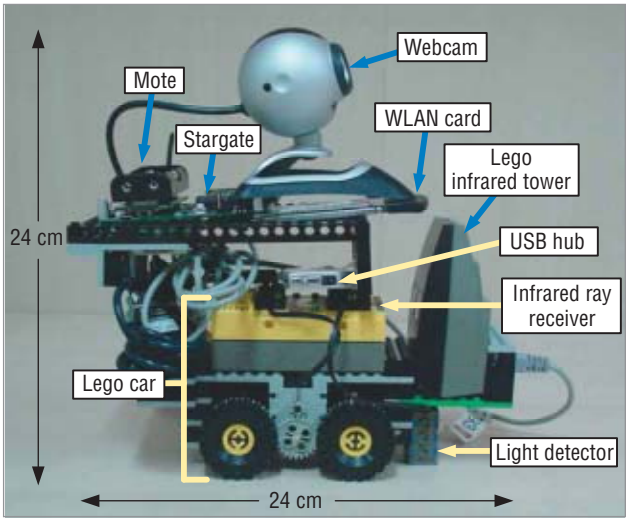


Figure 2. The mobile sensor. Attached to the Stargate processing board are a mote, a webcam, and an IEEE 802.11 WLAN card. A Lego car provides mobility.

Mobile sensor dispatch and traversal problems

Because mobile sensors are battery powered, we assign them to emergency sites to conserve their energy as much as possible. Specifically, we consider a set L of m emergency sites to be visited by a set S of n mobile sensors, where each site must be visited by one mobile sensor. We allow an arbitrary relationship between m and n . The goal is to maximize the mobile sensors’ total remaining energy after sites are visited.

Our dispatch solution depends on the relationship of m and n . When $m \leq n$, we can convert the problem to one of finding a maximum matching in a weighted bi-partite graph $G = (S \cup L, S \times L)$, where the vertex set is $S \cup L$ and the edge set is the product S

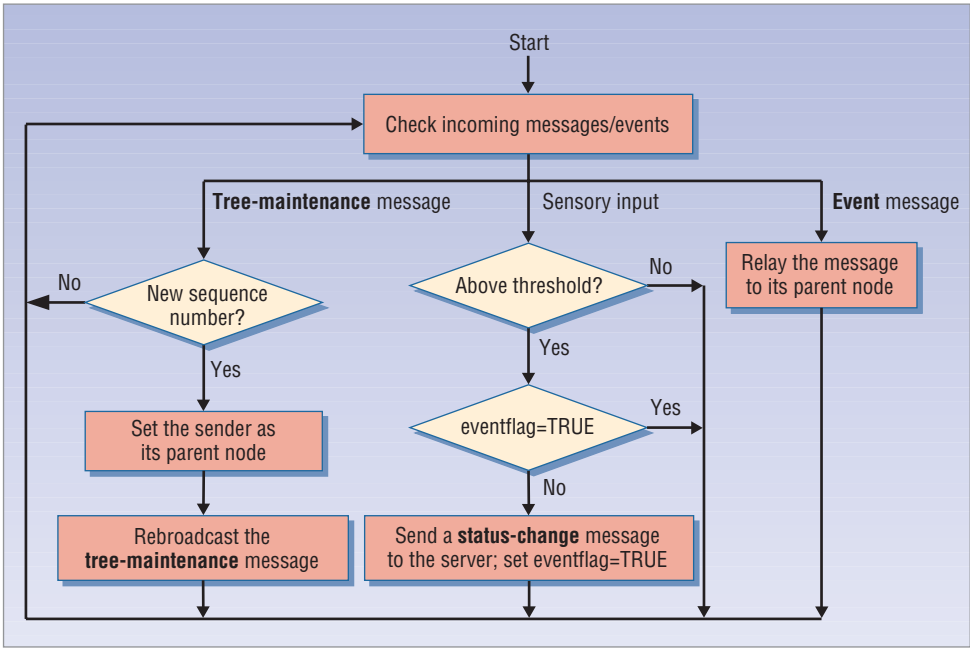


Figure 3. The algorithm executed by static sensors. Three types of messages activate a static sensor: tree-maintenance message, sensory input, and event message.

$\times L = \{(s_i, l_j) | s_i \in S, l_j \in L\}$. We set the weight of (s_i, l_j) to $e_i - e_{mv} \times d(s_i, l_j)$, where e_i is the current energy of s_i ; e_{mv} is the energy cost for a mobile sensor to move by one unit; and $d(s_i, l_j)$ is the distance from s_i 's current location to l_j . The solution is the maximum matching P of G , which we can find through traditional maximum-weight matching solutions.² Alternatively, we can set our objective to minimizing mobile sensors' total moving distances. We can also use maximum-matching to achieve this by setting the weight of (s_i, l_j) to $-e_{move} \times d(s_i, l_j)$.

When $m > n$, some mobile sensors must visit multiple sites. To solve this problem, we divide emergency sites into n clusters (for example, by the classical K -means method) and assign each group to one mobile sensor. In this case, each mobile sensor's cost will include moving to the closest site in each group and then traversing the rest of the sites one by one. Given a set of locations to be visited, we can use a heuristic to the traveling salesman problem² to determine the traversal order.

IMPLEMENTATION AND EXPERIMENTAL RESULTS

Our static sensors are MICAz motes (www.xbow.com/Products/productdetails.aspx?sid=164). A MICAz is a 2.4-GHz, IEEE 802.15.4-compliant module allowing low-power operations and offering a 250-Kbps data rate with a direct-sequence spread-spectrum (DSSS) radio.

The Stargate processing platform consists of a 32-bit, 400-MHz Intel PXA-255 XScale reduced-instruction-set computer (RISC) with a 64-Mbyte main memory and 32-Mbyte extended flash memory. It also has a daughterboard with an RS-232 serial port, a PCMCIA slot, a USB port, and a 51-pin extension connector, which can be attached to a mote. It drives the webcam through a USB port and the IEEE 802.11 WLAN card through its PCMCIA slot. The Stargate controls the Lego car via a USB port connected to a Lego

Related Work in Wireless Surveillance

Traditional visual surveillance systems continuously videotape scenes to capture transient or suspicious objects. Such systems typically need to automatically interpret the scenes and understand or predict actions of observed objects from the acquired videos. For example, Wu-chi Feng and his colleagues proposed a video-based surveillance network in which an 802.11 WLAN card transmits the information that each video camera captures.¹

Researchers in robotics have also discussed the surveillance issue.² Robots or cameras installed on walls identify obstacles or humans in the environment. These systems guide robots around these obstacles. Such systems normally must extract meaningful information from massive visual data, which requires significant computation or manpower.

Some researchers use static WSNs for object tracking.^{3,4} These systems assume that objects can emit signals that sensors can track. However, results reported from a WSN are typically brief and lack in-depth information. Edoardo Ardizzone and his colleagues propose a video-based surveillance system for capturing intrusions by merging WSNs and video-processing techniques.⁵ The system complements data from WSNs with videos to capture the possible scenes with intruders. However, cameras in this system lack mobility, so they can only monitor some locations.

Researchers have also proposed mobilizers to move sensors to enhance coverage of the sensing field⁶ and to strengthen the network connectivity.⁷ Other work addresses the pursuer-evader game, in which a pursuer must intercept an evader in the field with the assistance of WSNs.⁸ To our knowledge, no one has adequately addressed the integration of WSNs with surveillance systems, which motivates us to propose the iMouse system.

References

1. W.C. Feng et al., "Panoptes: Scalable Low-Power Video Sensor Networking Technologies," *ACM Trans. Multimedia Computing, Comm., and Applications*, vol. 1, no. 2, 2005, pp. 151-167.
2. J.H. Lee and H. Hashimoto, "Controlling Mobile Robots in Distributed Intelligent Sensor Network," *IEEE Trans. Industrial Electronics*, vol. 50, no. 5, 2003, pp. 890-902.
3. X. Ji et al., "Dynamic Cluster Structure for Object Detection and Tracking in Wireless Ad-Hoc Sensor Networks," *Proc. IEEE Int'l Conf. Comm.*, IEEE Press, 2004, pp. 3807-3811.
4. W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," *IEEE Trans. Wireless Comm.*, vol. 3, no. 5, 2004, pp. 1689-1701.
5. E. Ardizzone et al., "An Integrated Architecture for Surveillance and Monitoring in an Archeological Site," *Proc. ACM Int'l Workshop Video Surveillance and Sensor Networks*, ACM Press, 2005, pp. 79-85.
6. N. Heo and P.K. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks," *IEEE Trans. Systems, Man and Cybernetics—Part A: Systems and Humans*, vol. 35, no. 1, 2005, pp. 78-92.
7. P. Basu and J. Redi, "Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks," *IEEE Network*, vol. 18, no. 4, 2004, pp. 36-44.
8. C. Sharp et al., "Design and Implementation of a Sensor Network System for Vehicle Tracking and Autonomous Interception," *Proc. 2nd European Workshop Wireless Sensor Networks*, IEEE Press, 2005, pp. 93-107.

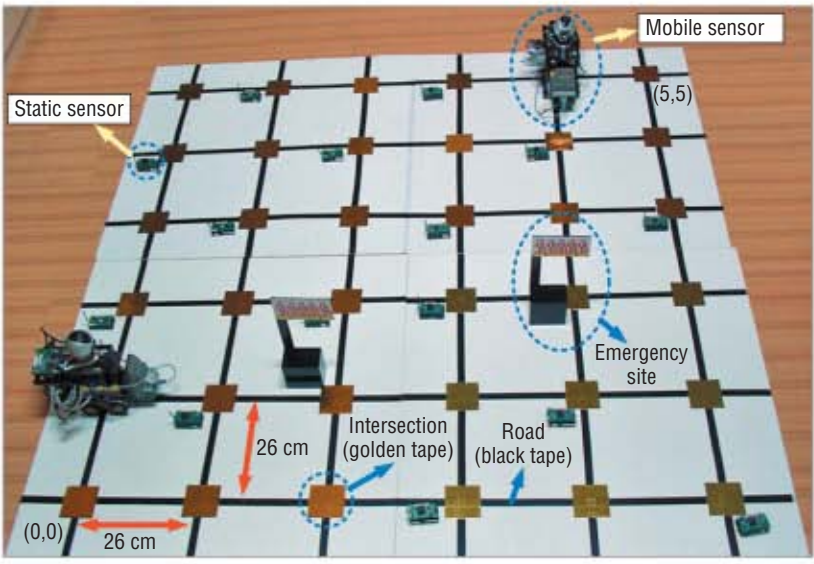


Figure 4. A 6×6 grid-like sensing field used in our experiment. Colored tape placed on the floor (black for roads, golden for intersections) is used to navigate the Lego car.

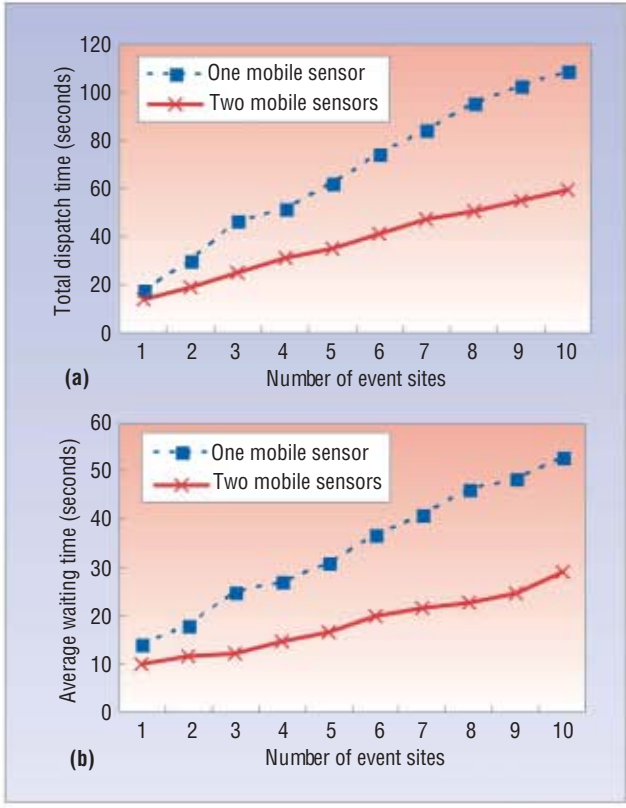


Figure 5. Experimental performance of (a) dispatch time and (b) average waiting time. As the graphs show, using two mobile sensors reduces dispatch and waiting times.

infrared tower, as Figure 2 shows. An infrared ray receiver on the front of the Lego car receives commands from the tower, and two motors on the bottom drive the wheels.

Navigating a mobile sensor or robot is difficult without some auxiliary devices. David Johnson and colleagues used wallboard cameras to capture mobile sensors' locations,³ while Jang-Ping Sheu and his colleagues suggested using signal strength to do so.⁴

Our current prototype uses the light sensors on the Lego car to navigate mobile sensors. We stick different colors of tape on the ground, which lets us easily navigate the Lego car on a board. In our prototyping, we implemented an experimental 6×6 grid-like sensing field, as Figure 4 shows. Black tape represents roads, and golden tape represents intersections. We constructed the system by placing two mobile sensors and 17 static sensors on the sensing field. For static sensors, a light reading below 800 watts simulates an event, so we cover a static sensor with a box to

model a potential emergency.

We use a grid-like sensing field and a grid-like static sensor deployment only for ease of implementation. In general, the static WSN's topology can be irregular.

Three factors affect the mobile sensors' dispatch time:

- the time that a mobile sensor takes to cross one grid-unit (about 26 centimeters),
- the time that a mobile sensor takes to make a 90-degree turn, and
- the time that a mobile sensor takes to make snapshots and report the results.

In our current prototype, the times are 2.5, 2.2, and 4.0 seconds, respectively.

Figure 5 shows experimental results with one mobile sensor initially placed at (0, 0) and two mobile sensors placed at (0, 0) and (5, 5). We generate some random events and evaluate the dispatch time (from when the server is notified of these events to when all event sites are visited) and the average time of each site (from when an event is detected to when a mobile sensor visits the site). Clearly, using two mobile sensors significantly reduces dispatch and waiting times.

At the external server, users monitor the system's status and control mobile sensors through a user interface, as Figure 6 shows. The user interface includes six major components:

- The *configure* area lets users input system configuration information, such as mobile sensors' IP addresses, ports, and sensors' positions.
- The *system-command* area provides an interface to let users control the overall system, such as issuing a

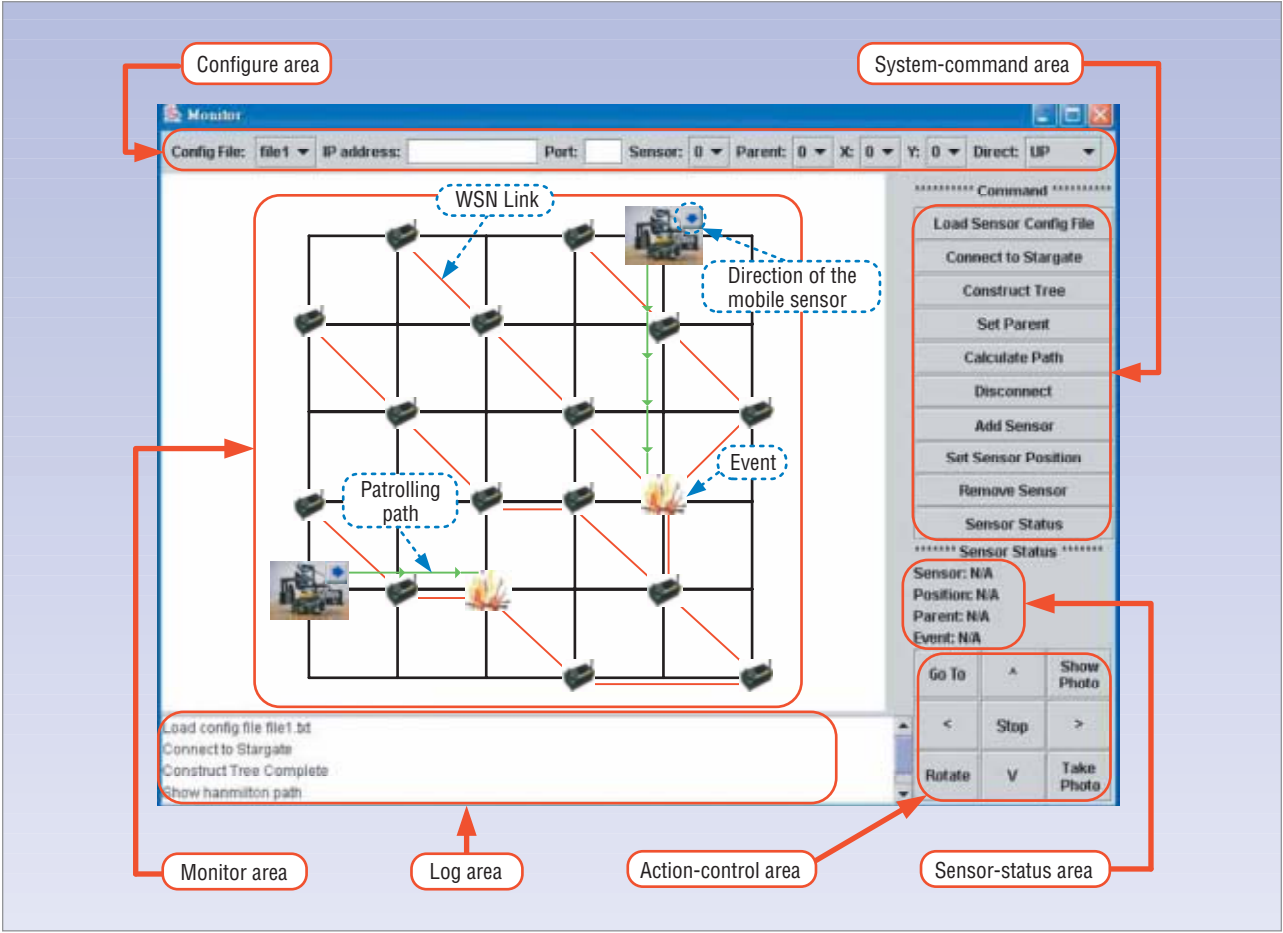


Figure 6. User interface at the external server. The interface consists of six areas through which the user can monitor the system's status and control the mobile sensors.

tree-maintenance message, adjusting the WSN's topology, and connecting and disconnecting a specified mobile sensor.

- The *sensor-status* area shows the current status of a static sensor being queried.
- The *action-control* area lets users control the mobile sensors' actions, including movement and taking snapshots.
- The *monitor* area shows the WSN's network topology and the mobile sensors' patrolling paths. When a sensor detects an event, a fire icon appears in the corresponding site.
- The *log* area displays some of the system's status messages.

SIMULATION RESULTS

Our experiments considered only grid networks. To help us understand the sensor dispatch problem in general irregular WSNs, we developed a simulator. We evaluated the average waiting time for an event location being visited and mobile sensors' total energy consumption. We compared a greedy algorithm against the *K*-means algorithm.

Given a set *S* of mobile sensors and a set *L* of emergency sites, the greedy algorithm contains a sequence of iterations. In each iteration, we assigned the mobile sensor with the smallest distance to the nearest location. We repeated this process until all mobile sensors were assigned to locations. If there were unvisited locations, the first mobile sensor reaching its destination picked its next location in the same greedy manner. This continued until all locations were visited.

In our simulation, the sensing field was 15 × 15 meters. We assumed that moving one meter takes one Joule and 10 seconds. A mobile sensor moves in a straight line from one location to another. Each experiment had 100 rounds, and in each round a certain number of events were generated at random locations. After each round, mobile sensors stayed at their final destinations and waited for the next schedules. We marked each simulation result with a 90 percent confidence interval.

Our comparison results under different numbers of event sites and mobile sensors showed that the greedy algorithm performs better than the *K*-means algorithm because of its time-critical nature. The *K*-means algorithm gives mobile sensors unbalanced job assignments, caus-

ing some event sites to wait longer even when some mobile sensors are idle. On the other hand, the *K*-means algorithm is more energy-efficient because of its clustering approach, which exploits event locality.

The iMouse system integrates WSN technologies into surveillance technologies to support intelligent mobile surveillance services. We can enhance or extend iMouse in several ways. First, we can improve mobile sensor navigation by, for example, integrating localization schemes to guide mobile sensors instead of using color tapes. Second, we can exploit coordination among mobile sensors, especially when they're on the road. Finally, we need to further investigate how we can use mobile sensors to improve the network topology. ■

Acknowledgments

Yu-Chee Tseng's research is cosponsored by Taiwan's Ministry of Education ATU Program; National Science Council of Taiwan grants no. 93-2752-E-007-001-PAE, 96-2623-7-009-002-ET, 95-2221-E-009-058-MY3, 95-2221-E-009-060-MY3, 95-2219-E-009-007, 95-2218-E-009-209, and 94-2219-E-007-009; Taiwan's Ministry of Economic Affairs under grant no. 94-EC-17-A-04-S1-044; the Industrial Technology Research Institute of Taiwan; Microsoft; and Intel.

References

1. T. He et al., "Range-Free Localization Schemes for Large-Scale Sensor Networks," *Proc. 9th ACM Int'l Conf. Mobile Computing and Networking* (MobiCom 03), ACM Press, 2003, pp. 81-95.

2. J.R. Evans and E. Minieka, *Optimization Algorithms for Networks and Graphs*, 2nd ed., Marcel Dekker, 1992.

3. D. Johnson et al., "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed," *Proc. 25th Ann. IEEE Conf. Computer Comm.* (Infocom 06), IEEE Press, 2006.

4. J.P. Sheu, P.W. Cheng, and K.Y. Hsieh, "Design and Implementation of a Smart Mobile Robot," *Proc. IEEE Int'l Conf. Wireless and Mobile Computing, Networking and Comm.* (WiMob 05), IEEE Press, 2005, pp. 422-429.

Yu-Chee Tseng is chair of the Department of Computer Science at National Chiao Tung University. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Tseng received a PhD in computer and information science from the Ohio State University. He is a member of the ACM and a senior member of the IEEE. Contact him at yctsens@cs.nctu.edu.tw.

You-Chiun Wang is a postdoctoral research associate in the Department of Computer Science at National Chiao Tung University. His research interests include wireless communication and mobile computing, QoS management and wireless fair scheduling, mobile ad hoc networks, and wireless sensor networks. Wang received a PhD in computer science from National Chiao Tung University. Contact him at wangyc@cs.nctu.edu.tw.

Kai-Yang Cheng is a master's student in the Department of Computer Science at National Chiao Tung University. His research interests include wireless communication and mobile computing, mobile ad hoc networks, and wireless sensor networks. Cheng received an MS in computer science and information engineering from National Chiao Tung University. Contact him at kycheng@cs.nctu.edu.tw.

Yao-Yu Hsieh is a graduate student in the Department of Computer Science at National Chiao Tung University. His research interests include wireless communication and mobile computing, mobile ad hoc networks, and wireless sensor networks. Hsieh received a BS in computer science and information engineering from National Chiao Tung University. Contact him at shiehyyg@cs.nctu.edu.tw.

Join the IEEE Computer Society online at
www.computer.org/join/



Complete the online application and get

- immediate online access to **Computer**
- a free e-mail alias — **you@computer.org**
- free access to 100 online books on technology topics
- free access to more than 100 distance learning course titles
- access to the IEEE Computer Society Digital Library for only \$118

Read about all the benefits of joining the Society at
www.computer.org/join/benefits.htm

66 Computer

Computer

Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

qMags



IEEE DISTRIBUTED SYSTEMS ONLINE

*a monthly online magazine brought to you
by the IEEE Computer Society*



IEEE Distributed Systems Online, the IEEE's first online-only publication, aims to promote professional awareness of developments, trends, activities, and editorial coverage in distributed systems. The magazine features free peer-reviewed articles as well as expert-moderated topic areas. For detailed guidelines on how to submit, see <http://dsonline.computer.org/author.html>.

Topics include:

- *Cluster Computing*
- *Grid Computing*
- *Web Systems*
- *Mobile & Pervasive*
- *Middleware*
- *Distributed Agents*
- *Security*
- *Parallel Processing*
- *Operating Systems*

<http://dsonline.computer.org>

RESEARCH FEATURE

Password-Based Authentication: Preventing Dictionary Attacks

Saikat Chakrabarti and Mukesh Singhal

University of Kentucky

Password-based authentication is susceptible to attack if used on insecure communication channels like the Internet. Researchers have engineered several protocols to prevent attacks, but we still need formal models to analyze and aid in the effective design of acceptable password protocols geared to prevent dictionary attacks.

Authentication provides a means of reliably identifying an entity. The most common verification technique is to check whether the claimant possesses information or characteristics that a genuine entity should possess. For example, we can authenticate a phone call by recognizing a person's voice and identify people we know by recognizing their appearance.

But the authentication process can get complicated when visual or auditory clues aren't available to help with identification—for example, when a print spooler tries to authenticate a printer over the network, or a computer tries to authenticate a human user logging in.

A computer can authenticate humans through

- biometric devices such as retinal scanners, fingerprint analyzers, and voice-recognition systems that authenticate *who the user is*;
- passwords that authenticate *what the user knows*;
- and smart cards and physical keys that authenticate *what the user has*.

Because they're cheap and convenient, passwords have become the most popular technique for authenticating users trying to access confidential data stored in computers. However, password-based authentication is vulnerable to several forms of attack.

People generally select short, easily memorized passwords to log in to a server without considering that pass-

word-based authentication methods are susceptible to attacks if used on insecure communication channels like the Internet. Meanwhile, complex passwords might get lost or stolen when users write them down, defeating the purpose of constructing secure password-based authentication schemes in the first place.

COMMON MECHANISMS AND SECURITY CONCERNS

Transmitting a password in plaintext from the user to the server is the simplest (and most insecure) method of password-based authentication. To validate a user password, the server compares it with a password (either in plaintext or an image of the password under a one-way function) stored in a file. However, this method lets an adversary passively eavesdrop on the communication channel to learn the password.

Challenge-response protocol

To secure against passive eavesdropping, researchers have developed *challenge-response protocols*.¹ To initiate a challenge-response protocol, Entity A sends a message containing A's identity to Entity B. Then B sends A a random number, called a *challenge*.

A uses the challenge and its password to perform some computation and sends the result, called a *response*, to B. Then B uses A's stored password to perform the same computation and verify the response. Since B chooses a different challenge for every run of the protocol, an

adversary can't simply eavesdrop, record messages, and resend them at a later time (a *replay attack*) to impersonate an entity.

Dictionary attacks

The challenge-response protocol is vulnerable to a *password-guessing attack*. In this kind of attack, we assume that an adversary has already built a database of possible passwords, called a dictionary. The adversary eavesdrops on the channel and records the transcript of a successful run of the protocol to learn the random challenge and response. Then the adversary selects passwords from the dictionary and tries to generate a response that matches the recorded one. If there's a match, the adversary has successfully guessed A's password.

After every failed matching attempt, the adversary picks a different password from the dictionary and repeats the process. This noninteractive form of attack is known as the *offline dictionary attack*.

Sometimes an adversary might try different user IDs and passwords to log in to a system. For popular Internet services like Yahoo!, the adversary can trivially choose any reasonable user ID due to the large number of registered users. An adversary can also find user IDs within interactive Web communities such as auction sites. If the system rejects the password as being incorrect for that particular user, the adversary picks a different password from the dictionary and repeats the process. This interactive form of attack is called the *online dictionary attack*.

Other security issues

Password-based authentication also can involve other security issues. Let's consider a scenario in which two entities, A and B, are trying to authenticate each other through a password protocol. An adversary can intercept messages between the entities and inject his own messages. In this *man-in-the-middle attack*, the adversary's goal is to play the role of A in the messages he sends to B and the role of B in the messages he sends to A.

In an *insider attack*, a legitimate user might try to attack other accounts in the system. Any additional information regarding a certain user might help in guessing that user's password.

PREVENTING OFFLINE DICTIONARY ATTACKS

Seeking convenience, people tend to choose weak passwords from a small sample space, which an adversary can easily enumerate. Thus, systems need something stronger than simple challenge-response protocols that can use these cryptographically weak passwords to securely authenticate entities. Such an authentication protocol would be deemed secure if, whenever an entity accepts an authentication session with another entity,

- 1. A : (E_A, D_A) .
- 2. A \rightarrow B : A, $K_{\text{pwd}}(E_A)$.
- 3. B : Compute $E_A = K_{\text{pwd}}^{-1}(K_{\text{pwd}}(E_A))$. Generate random secret key K_{AB} .
- 4. B \rightarrow A : $K_{\text{pwd}}(E_A(K_{AB}))$.
- 5. A : $K_{AB} = D_A(K_{\text{pwd}}^{-1}(K_{\text{pwd}}(E_A(K_{AB}))))$. Generate unique challenge C_A .
- 6. A \rightarrow B : $K_{AB}(C_A)$.
- 7. B : Compute $C_A = K_{AB}^{-1}(K_{AB}(C_A))$ and generate unique challenge C_B .
- 8. B \rightarrow A : $K_{AB}(C_A, C_B)$.
- 9. A: Decrypt message sent by B to obtain C_A and C_B . Compare the former with his own challenge. If they match, go to next step, else abort.
- 10. A \rightarrow B : $K_{AB}(C_B)$.
- 11. B : Decrypt message A sends and compare with challenge C_B . If they match, B knows that A has the ability to encrypt subsequent messages using key K_{AB} .

Figure 1. Algorithm 1: Encrypted key exchange. The EKE protocol uses a combination of symmetric and asymmetric cryptography.

it should have indeed participated in the authentication session.²

Guarantees of mutual authentication are essential for remote users trying to access servers over insecure networks like the Internet. The goal of a password-based authentication protocol aimed at preventing offline dictionary attacks is to produce a cryptographically strong shared secret key, called the *session key*, after a successful run of the protocol. Both entities can use this session key to safely encrypt subsequent messages.

Encrypted key exchange

Steven Bellovin and Michael Merritt³ made the first attempt to protect a password protocol against offline dictionary attacks. They developed a password-based encrypted key exchange (EKE) protocol using a combination of symmetric and asymmetric cryptography. Algorithm 1 in Figure 1 describes the EKE protocol, in which users A and B serve as the participating entities in a particular run of the protocol, resulting in a session key (stronger than the shared password) the users can later apply to encrypt sensitive data.

In Step 1, user A generates a public/private key pair (E_A, D_A) and also derives a secret key K_{pwd} from his password *pwd*. In Step 2, A encrypts his public key E_A with K_{pwd} and sends it to B. In Steps 3 and 4, B decrypts the message using the stored password of A, and uses E_A together with K_{pwd} to encrypt a session key K_{AB} and sends it to A.

In Steps 5 and 6, A uses this session key to encrypt a unique challenge C_A and sends the encrypted challenge to B. In Step 7, B decrypts the message to obtain the challenge and generates a unique challenge C_B .

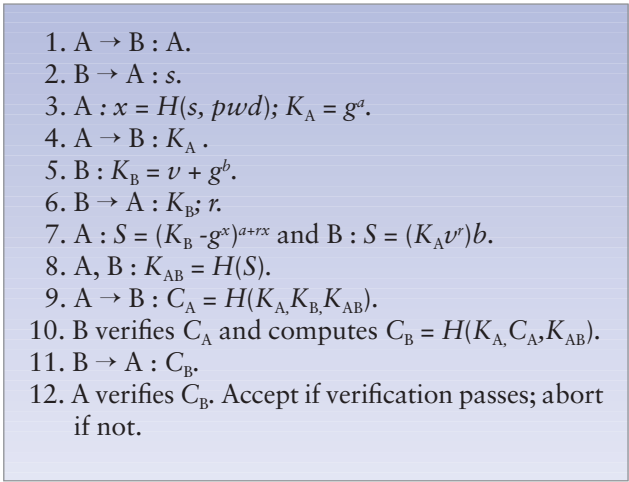


Figure 2. Algorithm 2: Secure remote-password protocol. SRP successfully eliminates plaintext equivalence.

In Step 8, B then encrypts both C_A and C_B with the session key K_{AB} and sends it to A. In Step 9, A decrypts this message to obtain C_A and C_B and compares the former with his own challenge. A match verifies the correctness of B's response.

In Step 10, A encrypts B's challenge C_B with the session key K_{AB} and sends it to B. In Step 11, B decrypts this message and compares it with his own challenge C_B . If they match, B knows that A can use K_{AB} to encrypt subsequent messages.

Bellovin and Merritt also developed augmented EKE (A-EKE),⁴ which stores passwords under a one-way function. The objective is to prevent an adversary who obtains the one-way encrypted password file from mimicking the user to the host. They implemented A-EKE using digital signatures and a family of commutative one-way functions. Researchers subsequently developed a gamut of protocols that provide stronger security guarantees than EKE and have additional desirable properties.

The EKE protocol and its variants (except A-EKE) suffer from *plaintext equivalence*, which means the user and the host have access to the same secret password or hash of the password. Intuitively, there are disadvantages to plaintext equivalence.

Imagine a simple case in which entity A (the user) enters his password in the client software, which uses a one-way function to hash the password and sends the hashed password over the network to entity B (the server). An adversary can eavesdrop on the channel to obtain entity A's hashed password and can impersonate entity A by resending the hashed password later.

To understand the problem of plaintext equivalence, we can extend the simple case to more complex challenge-response protocols, like EKE. This vulnerability will arise whenever two entities share a secret and perform symmetric operations, however complex, based on the shared secret and exchanged messages.

Secure remote password

Thomas Wu¹ combined *zero-knowledge proofs* with *asymmetric key-exchange protocols* to develop *secure remote password* (SRP), a verifier-based protocol that eliminates plaintext equivalence. If the password is a private key with limited entropy, we can think of the corresponding verifier as a public key. It's easy to compute the verifier from the password, but deriving the password, given the verifier, is computationally infeasible.

However, unlike with a public key, the entity doing the validation can keep the verifier secret. All SRP computations are carried out on the finite field F_n , where n is a large prime. Let g be a generator of F_n . Let A be a user and B be a server. Before initiating the SRP protocol, A and B do the following:

- A and B agree on the underlying finite field.
- A picks a password pwd and a random salt s , and computes the verifier $v = gx$, where $x = H(s, pwd)$ is the long-term private key and H is a cryptographic hash function.
- B stores the verifier v and the salt s corresponding to A. Now, A and B can engage in the SRP protocol.

Algorithm 2 in Figure 2 describes the SRP protocol, which works as follows:

In Step 1, A sends its username A to server B. In Step 2, B looks up A's verifier v and salt s and sends A the salt. In Steps 3 and 4, A computes its long-term private-key $x = H(s, pwd)$, generates an ephemeral public key $K_A = ga$ where a is randomly chosen from the interval $1 < a < n$ and sends K_A to B.

In Steps 5 and 6, B computes ephemeral public-key $K_B = v + gb$ where b is randomly chosen from the interval $1 < a < n$ and sends K_B and a random number r to A. In Step 7, A computes $S = (K_B - gx)a + rx = gab + brx$ and B computes $S = (K_A vr)b = gab + brx$. The values of S that A and B compute will match if the password A enters in Step 3 matches the one that A used to calculate the verifier v that is stored at B.

In Step 8, both A and B use a cryptographically strong hash function to compute a session key $K_{AB} = H(S)$. In Step 9, A computes $C_A = H(K_A, K_B, K_{AB})$ and sends it to B as evidence that it has the session key. C_A also serves as a challenge. In Step 10, B computes C_A itself and matches it with A's message. B also computes $C_B = H(K_A, C_A, K_{AB})$. In Step 11, B sends C_B to A as evidence that it has the same session key as A. In Step 12, A verifies C_B , accepts if the verification passes and aborts otherwise.

Unlike EKE, the SRP protocol doesn't encrypt messages. Since neither the user nor the server has access to the same secret password or hash of the password, SRP successfully eliminates plaintext equivalence. SRP is unique in its swapped-secret approach to developing a verifier-based, zero-knowledge protocol that resists offline dictionary attacks.⁵

A formal approach to prevention

Password protocols need more than heuristic arguments to provide security guarantees. The use of formal methods to analyze and validate security issues is of paramount importance in constructing “acceptable” password protocols.

Shai Halevi and Hugo Krawczyk² carried out the first rigorous security analysis of password-based authentication protocols, examining the use of password protocols for strong authentication and key exchange in asymmetric scenarios.

In an asymmetric scenario, the authentication server can store a private key for public-key encryption, but the client uses a weak password and doesn’t have a means to authenticate the server’s public key via a trusted third party. Halevi and Krawczyk presented and analyzed the security of simple and intuitive password-based authentication protocols—like a generic encrypted challenge-response protocol and a mutual authentication/key exchange protocol. They also proved that every authentication protocol that attempts to resist offline dictionary attacks needs public-key encryption and demonstrated that they could build a secure key-exchange protocol, given any such password protocol.

**Password protocols
need more than
heuristic arguments
to provide
security guarantees.**

Standard model

Other researchers including Mihir Bellare and his colleagues⁶ subsequently proposed formal models for password-authenticated key exchange. However, the formal validations of security don’t constitute proofs in the standard model. For example, Bellare used ideal ciphers to achieve provable security. The standard model is commonly used in modern cryptography.

Since we still don’t have proofs that any of the standard cryptographic building blocks have computational lower bounds, achieving common cryptographic goals requires making some complexity-theoretic hardness assumptions.⁷ Examples of such assumptions include the following:

- Factoring the product of large primes is hard.
- Computing the discrete logarithm is hard in certain sufficiently large groups.
- The Advanced Encryption Standard (AES) is a good pseudorandom permutation.

Although the proofs performed under the standard model use such assumptions, the cryptographic community widely accepts the standard model.

Alternative models

When constructing proofs, researchers often resort to

an alternative when proofs in the standard model are unappealing or provably impossible (<http://eprint.iacr.org/2005/210.pdf>). One such model is the *random-oracle model*, which constitutes a public random function that takes any string $s \in \{0,1\}^*$ as input and outputs n bits. For every input string, the output is uniform and independent of all other outputs. Powerful as it is, the random oracle doesn’t exist in the real world. A cryptographic hash function usually instantiates it.

Another alternative, the *ideal-cipher model*, uses a *block cipher*, an algorithm that accepts a fixed-length block of plaintext and a fixed-length key as input and outputs a block of cipher text that’s the same length as the block of plaintext. The block cipher is constructed with a k -bit key and an n -bit block size and is chosen

uniformly from the set of all possible block ciphers of the same form. Somewhat analogous to the random oracle model, a practical block cipher must instantiate the ideal-cipher model’s black box.

If a password-authenticated key-exchange protocol uses the random-oracle model or the ideal-cipher model to construct a formal analysis of its security and achieves provable security under that model, what guarantees do we get once we instantiate those alternative models?

Some cryptographers have doubted protocols using such alternative models to claim provable security. There are cases where instantiations of idealized models have resulted in erroneous outcomes. Zhu Zhao and his colleagues⁷ presented examples of real ciphers that resulted in broken instantiations of Bellare and his colleagues’ password protocol.

To the best of our knowledge, achieving provable security in a password-based authentication protocol (preventing offline dictionary attacks) based on the standard model is still an open problem. At the current stage of research, the best we can do is aim for achieving provable security under a formal model, maybe an idealized one, and not construct a protocol claiming security attributes based on heuristic arguments.

PREVENTING ONLINE DICTIONARY ATTACKS

Password-based authentication will continue to be the most commonly used authentication technique, and hacking and identity thefts will be the wave of the future. However, several techniques are available to help withstand online dictionary attacks, where the adversary tries to impersonate a user to the server by repeatedly trying different passwords from a dictionary of passwords.

Prevention techniques and drawbacks

In 2002, online dictionary attacks were blamed for eBay accounts being taken over and used to set up

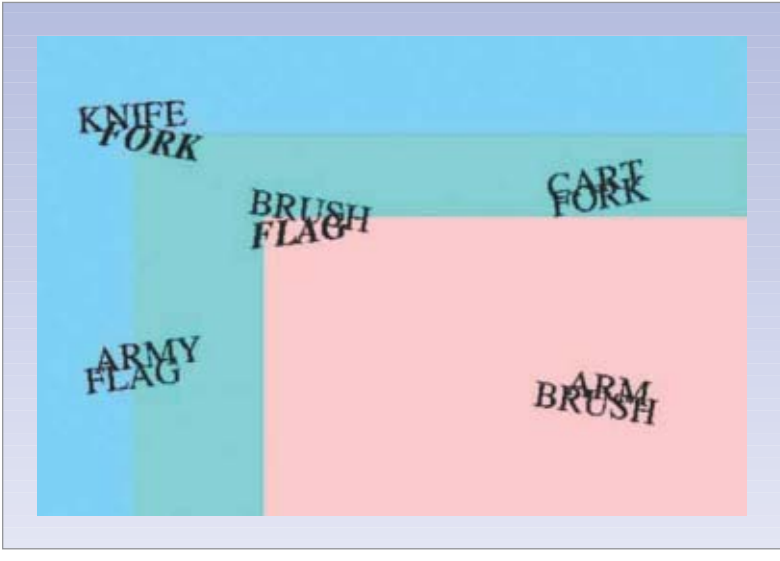


Figure 3. A Gimp is a Captcha based on optical-character recognition.

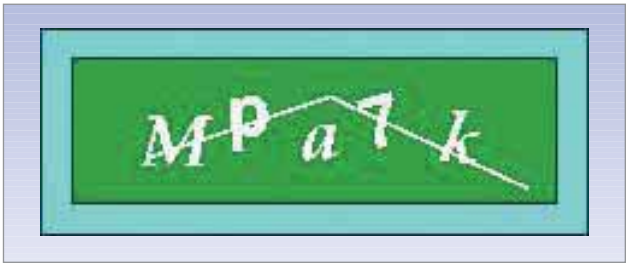


Figure 4. Yahoo! uses EZ-Gimp, which presents a distorted image over a textured background.

fraudulent auctions (http://news.zdnet.com/2100-9595_22-868306.html). Users clearly found this vulnerability unacceptable. The attacks pointed to the need for countermeasures, and Benny Pinkas and Tomas Sander responded with several mechanisms.⁸

Delayed response. After receiving a user ID/password pair, the server sends a slightly delayed response. This prevents an adversary from checking a sufficiently large number of passwords for a user ID in a reasonable amount of time.

However, just as a server can process several user logins in parallel, an adversary can try several login attempts in parallel to work around the delayed-response approach. For popular Internet services like Yahoo!, the adversary can trivially choose a user ID due to the large number of users. User IDs also can be found in interactive Web communities like auction sites.

Account locking. To prevent an adversary from trying many passwords for a particular user ID, systems can lock accounts after a certain number of unsuccessful login attempts. However, an adversary can mount a denial-of-service attack by choosing a valid user ID and trying several passwords until the account gets locked. This would cause a great inconvenience to the owners

of locked accounts, and setting up customer service to handle user calls regarding locked accounts wouldn't be cost-effective.

Performing extra computation. Originally developed to combat junk e-mail, this technique requires a user to perform some nontrivial computation and send proof of it while trying to log in.⁹ The idea is that the computation would be negligible for a single login attempt, but too expensive for a large number of login attempts.

For example, a server could require the following computation to be performed for every login attempt: Choose a value x so that the last 20 bits of $H(x, \text{user ID}, \text{password}, \text{time})$ are all 0, where H is a cryptographic hash function like SHA. If we assume H to be preimage resistant (given a message digest y , it's computationally infeasible to find x , such that $y = H(x)$), it would be necessary to check 219 values for x on the average to satisfy the condition.

A legitimate user might do this computation once, presenting a negligible overhead. However, performing this computation repeatedly for a large number of trial login attempts would present an extreme burden. The user's computer must run special software for the computation. In addition, the adversary might have a more powerful computer, and since the computation shouldn't be too time-consuming for a legitimate user, the adversary might have an edge in performing the dictionary attack.

A legitimate user might do this computation once, presenting a negligible overhead. However, performing this computation repeatedly for a large number of trial login attempts would present an extreme burden. The user's computer must run special software for the computation. In addition, the adversary might have a more powerful computer, and since the computation shouldn't be too time-consuming for a legitimate user, the adversary might have an edge in performing the dictionary attack.

Pinkas and Sander⁸ observed that an automated program must carry out such interactive forms of attacks, whereas legitimate users are humans. Thus, any login attempt must involve a test that a person can easily pass but an automated program can't.

Reverse Turing tests

Colorful images with distorted text have become commonplace at Web sites like Yahoo!, Hotmail, and PayPal. They're called reverse Turing tests (RTT) or Completely Automated Public Turing Tests to Tell Computers and Humans Apart (Captcha; <http://captcha.net>). Humans can easily pass the tests, but computer programs can't, even if they're knowledgeable about complete descriptions of the algorithms that created such tests.

Pinkas and his colleagues have implemented RTTs to prevent dictionary attacks. People can easily use the login accompanied by the RTT, but automated programs trying to carry out an online dictionary attack can't. The RTT should constitute a test with a small probability of a random guess producing a correct answer. For example, a test asking the user to identify whether an image is a man or a woman wouldn't be permissible since a random guess produces a correct answer with 50 percent probability.

Figure 3 shows Gimpy, a Captcha based on optical-character recognition. It renders a distorted image containing 10 words (some repeated), overlaid in pairs. Human users can easily read three different words from the distorted image, but computer programs can't.

Yahoo! uses an easier version called EZ-Gimpy, which presents a distorted image of a single word presented on a cluttered textured background, as Figure 4 shows.

Figure 5 illustrates Bongo, a Captcha that presents a visual pattern-recognition problem. Bongo asks users to distinguish between two blocks, then presents a single block and asks the user to determine whether it belongs to the right or left block.

A basic password-based authentication protocol using RTTs requires the user to pass an RTT before entering the user ID and password. This method has some drawbacks because it's demanding to ask users to solve an RTT for every login attempt. Currently, RTTs are more commonly generated for filling out online registration forms.

It's unknown whether the algorithm-generating RTTs can scale up to be used for every login attempt. Pinkas assumed that users log in from a limited set of computers containing activated cookies. So instead of using RTTs for every login, the user is asked to pass an RTT when initially trying to log in from a new computer or when entering a wrong password. The decision whether to present an RTT or not is a deterministic function of the entered user ID/password pair.

Stuart Stubblebine and Paul van Oorschot¹⁰ observed that RTT-based protocols are vulnerable to *RTT relay attacks*. Suppose an adversary wants to perform an online dictionary attack at the eBay Web site. For this, it needs correct responses to the RTTs. But an adversary can hack a high-volume Web site such as cnn.com and install attack software, which initiates a fraudulent attempt to login at ebay.com when a visitor goes to cnn.com. The RTT challenge is redirected to the user trying to view the cnn.com page.

Many nontechnical users will solve the RTT, unaware that the attack software will relay the answer to eBay, thus solving the RTT challenge. Solving the RTT, along with a sufficient number of password guesses, can crack an eBay account password. To counter these kinds of RTT relay attacks, Stubblebine developed a protocol based on a user's login history, suggesting modifications to Pinkas's RTT-based protocol.

Stubblebine suggested that only trustworthy machines store cookies. He also recommended that systems track users' failed-login attempts and set failed-login thresh-

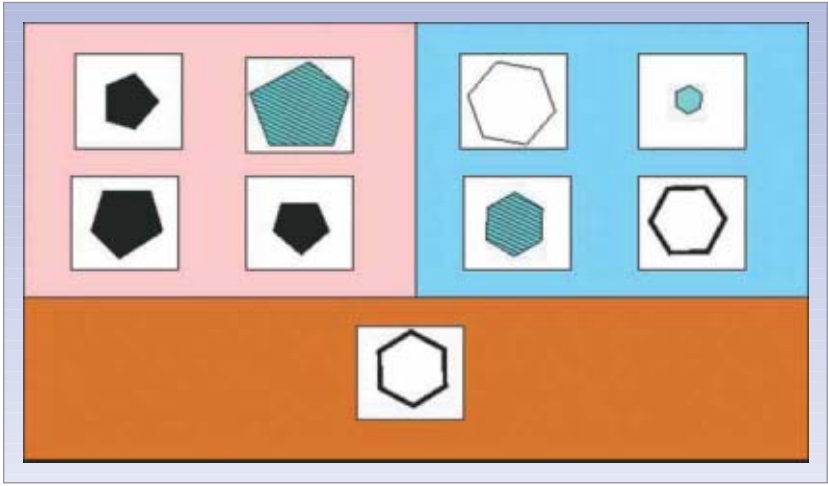


Figure 5. Bongo. This Captcha asks users to solve a visual pattern-recognition problem.

olds. His analysis of the protocol showed that it's more secure and user friendly.

Engineers or security architects wanting to select appropriate authentication techniques should be careful if they want to implement Captchas commercially. Hewlett-Packard holds a US patent on several forms of Captchas¹¹ and Yahoo! has applied for a patent on an image-verification system to prevent messaging abuse.¹²

Password-based authentication should continue to be the most common technique for user verification, as will attacks on it through a combination of hacking and identity theft. Password protocols preventing offline dictionary attacks need more than heuristic arguments to provide a guarantee of security. Although researchers have developed formal models for password-authenticated key exchange, the formal validations of security don't constitute proof in the standard model. While RTTs serve as tests that humans, but not automated programs, can pass, it's demanding to ask users to solve an RTT for every login attempt. Consequently, effective design of password protocols using RTTs requires a good balance between tight security and user friendliness. ■

Acknowledgments

The authors thank the anonymous reviewers whose valuable comments helped improve this article. This research was partially supported by grant no. T0505060 from the US Treasury Department.

References

1. T. Wu, "The Secure Remote Password Protocol," *Proc. Network and Distributed System Security (NDSS)*, The Internet Soc., 1998, pp. 97-111; <http://isoc.org/isoc/conferences/ndss/98/wu.pdf>.

2. S. Halevi and H. Krawczyk, "Public-Key Cryptography and Password Protocols," *ACM Trans. Information System Security*, ACM Press, vol. 2, no. 3, 1999, pp. 230-268.
3. S.M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Proc. IEEE Symp. Security and Privacy*, IEEE CS Press, 1992, pp. 72-84.
4. S.M. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise," *Proc. ACM Conf. Computer and Comm. Security*, ACM Press, 1993, pp. 244-250.
5. D. Bleichenbacher, "Breaking a Cryptographic Protocol with Pseudoprimes," *Proc. 8th Int'l Workshop Theory and Practice in Public-Key Cryptography (PKC 2005)*, LNCS 3386, Springer, 2005, pp. 9-15.
6. M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated Key Exchange Secure Against Dictionary Attacks," *Advances in Cryptology—EUROCRYPT 2000, Proc. Int'l Conf. Theory and Application Cryptographic Techniques*, LNCS 1807, Springer, 2000, pp. 139-155.
7. Z. Zhao, Z. Dong, and Y. Wang, "Security Analysis of a Password-Based Authentication Protocol Proposed to IEEE 1363," *Theoretical Computer Science*, vol. 352, no. 1, Elsevier, 2006, pp. 280-287; <http://sis.uncc.edu/~yonwang/papers/TCSsrp5.pdf>.
8. B. Pinkas and T. Sander, "Securing Passwords Against Dictionary Attacks," *Proc. 9th ACM Conf. Computer and Comm. Security*, ACM Press, 2002, pp. 161-170.
9. C. Dwork and M. Naor, "Pricing via Processing or Combating Junk Mail," E.F. Brickell, ed., *Advances in Cryptology—CRYPTO '92*, LNCS 740, Springer, 1993, pp. 139-147.
10. S.G. Stubblebine and P.C. van Oorschot, "Addressing Online Dictionary Attacks with Login Histories and Humans-in-the-Loop," *Financial Cryptography*, LNCS 3110, Springer, 2004, pp. 39-53; www.ccsf.carleton.ca/paper-archive/pvanoorschot-fc-04.pdf.
11. M.D. Lillibridge et al., *Method for Selectively Restricting Access to Computer Systems*, US patent 6,195,698, Patent and Trademark Office, 1998.
12. *Method and System for Image Verification to Prevent Messaging Abuse*, US patent application, 2004/0199597, Patent and Trademark Office, 2004.

Saikat Chakrabarti is a PhD student in the Computer Science Department at the University of Kentucky. His research interests are network and distributed-system security and applied cryptography. He received a BS in electrical engineering from Bengal Engineering and Science University, Shibpur, India. He is a student member of the IEEE and the ACM. Contact him at schak2@cs.uky.edu.

Mukesh Singhal is the Gartner Group Endowed Chair in Networking in the Computer Science Department at the University of Kentucky. His research interests are computer network security, distributed computing and operating systems, and wireless and high-speed networks. He received a PhD in computer science from the University of Maryland. He is an IEEE Fellow. Contact him at singhal@cs.uky.edu.

Sign Up Today



For the
IEEE
Computer Society
Digital Library
E-Mail Newsletter

- Monthly updates highlight the latest additions to the digital library from all 23 peer-reviewed Computer Society periodicals.
- New links access recent Computer Society conference publications.
- Sponsors offer readers special deals on products and events.

Available for FREE to members, students, and computing professionals.

Visit http://www.computer.org/services/csdl_subscribe

FREE!**YOUR
ORGANIZATION**

may qualify for a

**FREE
30-DAY TRIAL.**

Send an e-mail to
csdl@computer.org
for details.



IEEE
computer
society

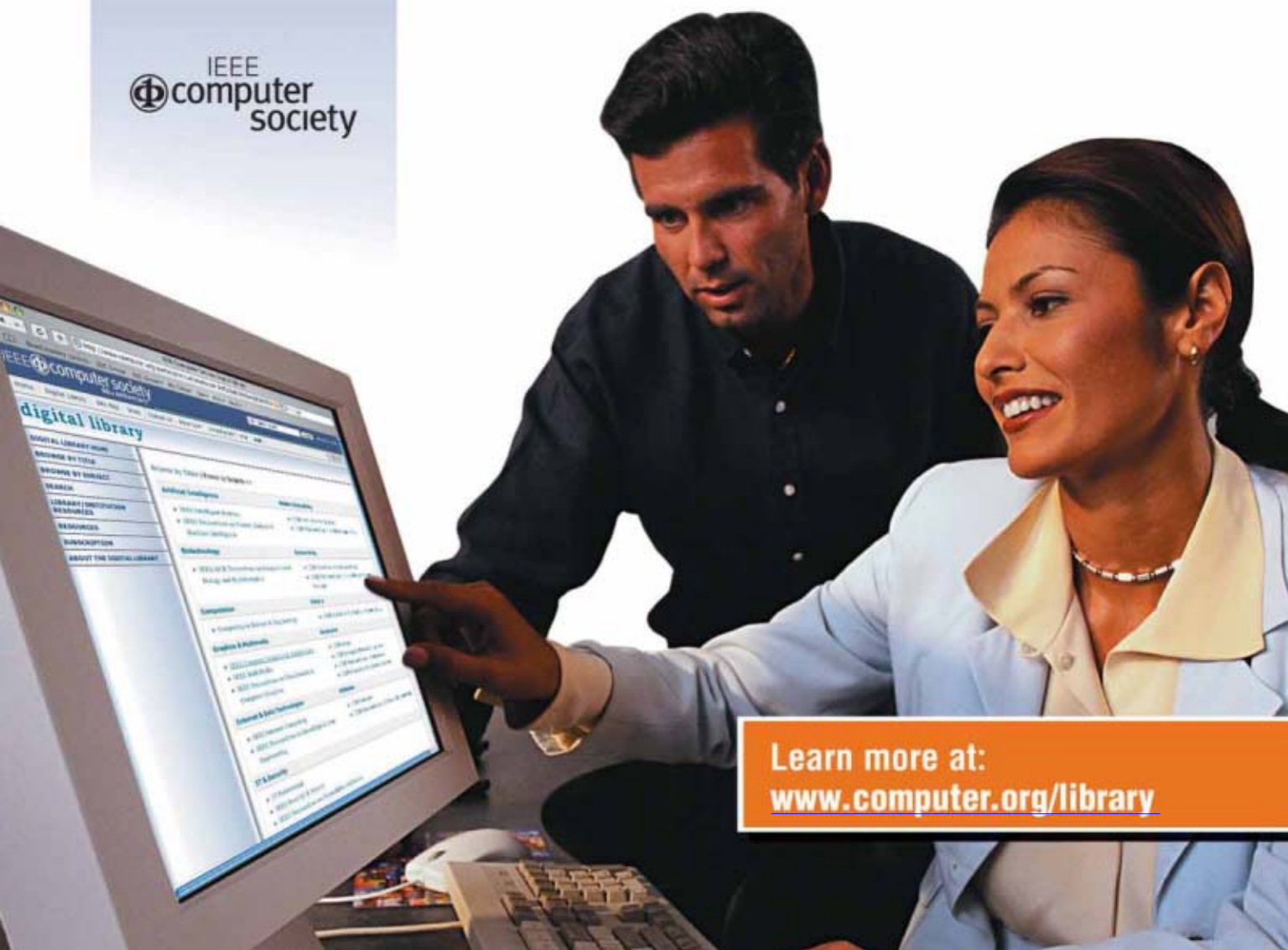
180,000 Computing Articles

... are as close as your keyboard!

IEEE Computer Society Digital Library

A critical computer science and information technology resource for academic, government, and corporate libraries around the world ... **does your organization subscribe?**

- 180,000+ full text documents
- 23 peer-reviewed journals
- 2,200+ conference publications
- Saved search and other enhanced features
- Plus smaller collections to fit any budget



Learn more at:
www.computer.org/library

CAREER OPPORTUNITIES

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Miami, FL. **Territory Sales Specialist.** Reqs. exp. in selling; customer relations; selling strategies & designing strategies. Reqs. incl. Bachelor's degree or foreign equiv. in Business Admin., Business Mgt. or related & 5 yrs of related exp. Send resume & refer to job #MIARGO. 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 San Diego, California. **ITO Consultant III - Master.** Responsible for representing Enterprise Application Org. tower during the company's new deal pursuit process. Reqs. SAP cert. incl. latest technologies; project mgt skills/exp.; presentation skills; leadership skills; willing to travel to other companies and/or customer sites in US & Canada. Reqs. incl. Bachelor's degree or foreign equiv. in CS, CE or related & 8 years of related exp. Send resume & refer to job #SANSNA. 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.

COMPUTER PROGRAMMER sought by Ruxmann LLC, N. Bergen, NJ, w/2 yrs. exp. Bachelors MIS or Comp. Sci. to plan & support hotel info processing needs. Eval, dvlp, coord & implmt new systms for host data & telecom. Maintain inventories, propose & create new systms to integrate w/bus., maintain, monitor & troubleshoot in-house systms, oversee telecoms systms. Resume to Attn: Mahesh Ratanji, Ruxmann LLC, 1270 Tonnelle Ave, N. Bergen, NJ 07047.

PROGRAMMER ANALYST. Install, configure, administer, tune, troubleshoot and upgrade Sybase Replication Server 11.5.1 and SQL servers using Async I/O. Transfer data between various servers using BCP. Add logins, roles, users, and groups to the Sybase SQL Server and maintain security procedures. Perform data modeling-logical and physical design and create logical devises and databases. Req:

Master's in Comp. Sci, Comp. Eng., or Electrical Eng. 40 hr/wk. Job/Interview Site: Naperville, IL. Send Resume to: LUCEO, Inc. @ 1631 Kallien Ave, Naperville, IL 60540.

PROGRAMMER ANALYST. Develop web pages using ASP/HTML/VBscript/Javascript, created COM components using Visual Basic, Turning SQL scripts for website performance optimization, client interaction. Create test plans & conduct unit /integration testing. Identify logical errors and modify programs. Provide production support, website configuration and release management. Perform database backup procedures and restoration procedures. Req: Master's Deg. in Comp Sci, Comp Eng or Elec Eng. 40 hr/wk. Job/Interview Site: Newport Beach, CA. Send resume to: Restoration Media, Inc @ 15143 Woodlawn Ave. Tustin, CA 92780.

COMPUTER ENGINEER. Design, code, test, and debug different modules of our custom software products. Prepare technical reports and other documentation. Encode, test, debug and install the oper-



NVIDIA Corporation, market leader in graphics and digital media processors, has employment opportunities for engineering professionals at our facilities in the following areas:

Santa Clara, CA • Austin, TX • Bellevue, WA • Durham, NC.

Current openings for various levels include:

- Analyst, Business Systems (ABS00)
- Applications Engineer (APPENG00)
- Architecture Engineer (ARCENG00)
- ASIC Design Engineer (ASICDE00)
- Compliance Engineer (COMPE00)
- Development Relations Engineer (DRE00)
- Engineer, Information Systems (ISENG00)
- Hardware Engineer (HW00)
- Human Resources Business Partner Coordinator (HRBP00)
- Program Manager (PROGM00)
- Programmer Analyst (PROGA00)
- Signal Integrity Engineer (SIGE00)
- SAP Programmer Analyst (SAP00)
- Software Engineer (SWE00)
- System Bios Engineer (SBE00)
- System Design Engineer (SYSDE00)
- Systems Software Manager (SWMGR00)
- Systems SW Engineer (SSWE00)

These opportunities require various education and experience backgrounds. If interested, please reference job code and forward your resume to: NVIDIA Corporation. ATTN: MS04 (D. Lopez), 2701 San Tomas Expressway, Santa Clara, CA 95050.

SUBMISSION DETAILS: Rates are \$299.00 per column inch (\$320 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to: Marian Anderson, Classified Advertising, *Computer Magazine*, 10662 Los Vaqueros Circle, PO Box 3014, 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.

ating programs and procedures. Perform tests on utility software, development software, and diagnostic software. 40hrs/wk. Req: Bachelor's in Comp Sci, MIS, CIS, Eng., or Electronics Eng., or Foreign Equiv. *A 3 yr. foreign bachelor's equiv. to a U.S. Bachelor's will meet requirement. Job/Interview Site: Hawthorne, CA. Send resume to Abacus Security Services, Inc. @ 12509 Crenshaw Blvd, Hawthorne, CA 90250.

INDUSTRY SOLUTION PRINCIPAL sought by Ascendant Technology, Austin, TX. Req.: BS (or for. equiv.) in Engg. or Tech. field + 5yr. IT exp. incl. prog. mgmt. Resume only attn.: C. Jones (File #071097) 10215 161st Pl. NE Redmond, WA 98052. Job Order: #6021076.

ANALYST. Amgen has an opportunity for a Business Analyst II. Requires related Master's & 3 yrs exp. or Bachelor's & 5 yrs. exp; and exp. with. Siebel; Cognos Reportnet; data interfaces/file transfer; reporting and analysis skills, documentation management; project management; technical diagram/process flow charting including Popkin, RUP, UML; MS Office Applications including MS Project; and

MS Windows. Job site: Thousand Oaks, CA. Applicants send resume and reference #6MQREW to: Kyle Foster, Amgen, Inc., One Amgen Center Drive, Mailstop 19-1-A Thousand Oaks, CA 91320. No phone calls or e-mails please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

SOFTWARE ENGINEER wanted to dvlp tools for monitoring n/work & systm performance. Must have Bach deg in Comp Sci or related field & 2yrs exp in job offd. Mail resumes to: United Computer Solutions, Inc., Attn: Mohammed Ahmed, 5215 Church St, Skokie, IL 60077. Ref to Code 105-11745-B. No calls.

SYSTEMS ANALYST (Accounting). Design, develop, & test Info Systems for financial & accounting projects using Oracle based programming languages/ technologies to meet specific business requirement. Identify logical errors & modify systems as required. Test the applications performance, data integrity & validations issues Design, analyze, develop & implement the different accounting rules. Troubleshoot servers related to FSG, ADI, & Data Interfacing

and Batch payments used in production, testing and development environment. Req: MBA w/courses in Acctg, or Master's in Comp. Sci. 40hr/wk. Job/Interview Site: Monrovia, CA. Send resume to: MSN Solutions, Inc. @ 150 N Santa Anita Ave, Suite # 300, Arcadia, CA 91006.

ENGINEERING. Nokia Inc. has the following exp/degree position in Irving, TX. Travel to unanticipated U.S. worksites may be required. NA Sales Area Allocation Manager: Develop, manage and coordinate supply chain operational processes including deploying, testing and supporting planning applications. ID# 07-TX-SAM. Send resumes to nokusjobs1@nokia.com, and reference ID#. Equal Opportunity Employer.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Roseville, California. **Systems/Software Engineer VI.** Reqs. BS in CS, CE, EE or related and 5 yrs related exp. Knwldge of Fibre channel expertise; Fibre channel analyzer usage and debug capability; knwldge of operating systems such as Linux, HPUX, AIX, Solaris, VMWare, Netware; knwldge of Oracle and Syman-

Head of Department of Computer Science and Engineering



THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

The Hong Kong University of Science and Technology (HKUST), opened in October 1991, comprises four Schools: Science, Engineering, Business & Management, and Humanities & Social Science. The University's mission is to advance learning and scholarship; to promote research, development, and entrepreneurship, and to contribute to the region's economic and social well-being. Together with other universities in Hong Kong, HKUST will change from its current 3-year undergraduate system to a 4-year system in 2012. Significant increase in the student population and faculty strength is expected as a result of this transition.

The School of Engineering, the largest School of the University, currently enrolls about 40% of the University's total undergraduate and postgraduate students of approximately 8600. It comprises six departments: Chemical Engineering, Civil Engineering, Computer Science & Engineering, Electronic & Computer Engineering, Industrial Engineering & Logistics Management, and Mechanical Engineering.

The Department of Computer Science & Engineering (CSE) currently has 39 faculty members, teaching about 400 undergraduate students and 140 postgraduate students. The Department conducts comprehensive teaching and research programs in both basic and applied aspects of Computer Science & Engineering. The academic degrees offered by the Department are: BEng, MSc, MPhil and PhD. Research activities in the department are broadly categorized into theoretical computer science; artificial intelligence; data, knowledge and information management; networking and computer systems; software technologies; vision and graphics. For more information, please visit the Department website at <http://www.cse.ust.hk/>.

Applications/nominations are invited from well-qualified and accomplished scholars for the position. In addition to extensive teaching and research experience, the successful candidate must have demonstrated leadership qualities necessary to lead and manage the Department in its diverse academic and administrative functions and to interact effectively with industry and commerce.

Salary will be highly competitive with generous benefits. Applications/nominations together with a detailed curriculum vitae and the names and addresses/fax numbers/email addresses of three referees should be sent to Professor Khaled BEN LETAIEF, Chair of Search Committee for Headship of CSE, c/o School of Engineering, HKUST, Clearwater Bay, Kowloon, Hong Kong [Fax No.: (852) 2358-1458, e-mail: eeekhaled@ust.hk] before 31 July 2007.

tec (Veritas suite); knwldge of Enterprise Storage Arrays architecture and functionality; knwldge of Storage Area Networks and Switch tech; knowledge of HBA drivers and firmware/architecture; knwldge of multi-pathing native and 3rd Party software solutions. Send resume referencing # ROSADI. Please send resumes with reference 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 is accepting resumes for the position of **Software Designer** in Vancouver, WA (Reference # VANATO). Please send resumes with reference number to: Hewlett-Packard Company, 19483 Pruneridge Avenue, Mail Stop 4206, Cupertino, California 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

COMPUTER SYSTEMS ANALYST sought by Import/Export Co. in Chino, CA. Req'd degree. Respond by resume only to Wendy Wong-Controller M/L-#10. Advance Components Specialist, Inc., 13971 Norton Ave., Chino, CA 91710.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in San Diego, California. **Engineering Project Manager III.** Reqs. MS in CS, CE, Engineering or related and 4 yrs related exp. Knowledge of Enterprise software development; Software development life cycle; JavaScript; and Enterprise level service management applications. Send resume referencing #SANMDR. Please send resumes with reference 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 San Diego, California. **SW Designer V – Research Scientist** to work with customers, prdct dvlpers and prdct mgrs to define and dvlp core analytic capabilities that are embedded in company's DecisionCenter SW. Reqs. MS in Computer Science, Operations Research, Information Technology, Engineering or related and 6 yrs related exp. Exp. w/ Linear/Logistic Regression or Neural Networks, Linear and Non-Linear Programming, Clustering, Decision Tree, Data

Cleaning, SAS, MatLab, and Scripting/ Parsing language. Send resume referencing #SANGBA. Please send resumes with reference 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 is accepting resumes for the following positions in San Diego, California. Hardware Engineer (Reference # SDCTH). Business Strategy Manager (Reference # SDHMI). Please send resumes with reference number to: Hewlett-Packard Company, 19483 Pruneridge Avenue, Mail Stop 4206, Cupertino, California 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

HEWLETT-PACKARD COMPANY is accepting resumes for the following position in Cupertino, CA. **Financial Analyst** (Reference # CUPAMA). Please send resumes with reference number to: Hewlett-Packard Company, 19483 Pruneridge Avenue, Mail Stop 4206, Cupertino, California 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, California. **Systems/Software Engineer.** Reqs. exp. w/ Ethernet drivers dev.; exp. w/HP-UX; knowledge of Ethernet, DLPI & transport; knowledge of PCI-X bus standards; knowledge of PCI-express standard; exp. w/ Logic and Protocol analyzers; knowledge of HP Server archit. on I/O concepts; & good understanding of product life cycles. Reqs. incl. Bachelor's degree or foreign equiv. in CS or related & 5 yrs of related exp. Send resume & refer to job #CUPRNA. 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.

CONSULTANT/PROGRAMMER ANALYST wanted f/t in Poughkeepsie, NY. M-F, 40hr/wk. Bach deg or equiv in Engg or Comp Sci & 2 yrs exp testing applics using Oracle, Visual Basic, Java & J2EE. Send resume: Indotronix International Corp., Attn: Recruiting (VSS), 331 Main St, Poughkeepsie, NY 12601.

HEWLETT-PACKARD COMPANY is accepting resumes for the following positions in Palo Alto, CA. IT Optimization Leader (Technical Analyst) (Reference # PALMMA). Research Scientist (Reference # PALJWY). Please send resumes with reference number to: Hewlett-Packard Company, 19483 Pruneridge Avenue, Mail Stop 4206, Cupertino, California 95014. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

CONSULTANT/PROGRAMMER ANALYST F/T (Poughkeepsie, NY). Must have Bach deg or equiv in Comp Sci, Engg or related & 1 yr exp dvlpng DTS packages. Send resume: Indotronix International Corporation, Recruiting (MM), 331 Main St, Poughkeepsie, NY 12601.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in Cupertino, California. **Systems/Software Engineer VI.** Reqs. BS in Computer Engineering, Electrical Engineering, Electronics Engineering or related and 5 yrs related exp. Reqs. Driver dvlpmt skills in a Unix kernel environment. Multiprocessor Kernel issues. SAS and RAID protocol dvlpmt. SCSI protocol. Dvlpmt of virtualization technologies. SCSI protocols and ability to read and interpret SCSI protocol analyzer traces. Send resume referencing #CUPALN. Please send resumes with reference 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.

DATA ADMINISTRATOR. China Times Printing, Inc in San Gabriel seek db admin. Design/implement. Coordinate data/codes/tests. Customize db for company business. Enforce data integrity. Calculate optimum values for db parameters. Specify user access. Good in Chinese verbal & writing, Exp.& MS deg. in Comp. Sci. Req. Pls. send CV to G.M. 626-308-2037.

HEWLETT-PACKARD COMPANY has an opportunity for the following position in San Ramon, California. **Account Delivery Manager.** Resp. for serving as the technical/service liaison between the company/client and the company and client software end users. Reqs. exp. w/ERP applications & PeopleSoft Hosting Model. Reqs. incl. Bachelor's degree or foreign equiv. in CS, CE, EE, Elect. & Comm. Eng. or related & 5 yrs of related

exp. Send resume & refer to job #SAN-BCH. 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.

COMPUTER NETWORK ARCHITECT

(NJ) Dsgn innovative application-layer middleware solutions for optimizing communication as well as caching & load-balancing in distributed multi-component applcs. Implmt algorithms in Service-Oriented Architecture; eval performance w/web svc benchmarks. Ph.D. in Comp Sci or Comp Engg req. Knowl/exp of: protocols for distributed coordination, group communication & data consistency; dsgn, implmtn, testing & measurement of large, distributed s/ware systems; dsgn & implmtn of n/work protocols at the applic & transport layers; dsgn of algorithms for publish-subscribe communication & content-based routing; technologies for overlay networking & peer-to-peer n/works; Unix/Linux system internals & admin; Unix system prgmg; Service-Oriented Architecture & Web Svcs technologies; C,

C++, Java, & Perl prgmg. Mail resume to Ms. Mansour, NEC Laboratories, 4 Independence Way, Princeton, NJ 08540 (Code: CNA-01).

PEROT SYSTEMS TSI (AMERICA)

INC., has positions in sales and business development, project management, programming and software engineering in Plano, TX, Longmont, CO, and Santa Clara, CA. Master's degree or Bachelor's degree with experience required based on position. Follow this link to apply online: www.perotsystems.com/careers/jo11689.

HEWLETT-PACKARD COMPANY

has an opportunity for the following position in Cupertino, California. **Systems/Software Engineer V.** Reqs. BS in Computer Engineering, Electrical Engineering, Mechanical Engineering or related and 5 yrs related exp. Reqs. Device Driver Development skills, Unix operating environment, Software Development Lifecycle, C programming language, Kernel debugging tools, HP-UX Kernal Internals. Send resume referencing #CUPSPR. Please send

resumes with reference 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.

UGS CORP and its subsidiaries including

Tecnomatix Technologies, Inc. have positions in software/implementation engineering, sales engineering, PLM and technical/software marketing in various locations including Plano, TX, Richardson, TX, Cypress, CA, San Jose, CA, Milford, OH, Ann Arbor, MI, Detroit, MI metropolitan area and Seattle, WA metropolitan area. Ph.D., Master's degree or Bachelor's degree with experience required based on position. Send resumes to UGS Corp. at PLMCareers@ugs.com and list location of interest. Job code J05 must be referenced in email subject line. EOE.

SENIOR SOFTWARE ENGINEER,

Jacksonville, FL: Analyze, design, develop sophisticated web based applications using ASP.NET, VB.NET, ADO.NET, C#, Visual Basic, COM, SQL Server,

JOIN AN INTERNATIONAL FACULTY AT SRM UNIVERSITY



SRM University is a private University that offers undergraduate and graduate programs in Engineering, Medicine, Dentistry, Para-medical sciences, Arts and Humanities.

As part of our University's globalization efforts, we are in search of Deans, Professors at various levels in the College of Engineering. Faculty duties include teaching at graduate and undergraduate levels, research and supervision of student research. Candidates with an active interest and background in all areas of Engineering such as Electrical Engineering, Electronics Engineering and Computer Engineering will be considered.

We are soliciting professors at various levels who can relocate, preferably for atleast 2-3 years. Professors who can stay for at least 6 months in India and teach a course for a semester are also encouraged to apply. The positions are

open to competent professors from the International academia with vast experience in academics and research. NRI professors from other countries who wish to work in India for a period of 6 months to 3 years are welcome to submit their applications. Suitable work visas will be arranged by us wherever necessary. Remuneration will be commensurate with international standards and will not be a constraint for candidates who have excelled in their chosen academic fields.

Interested candidates may send their latest resume to registrar@srmuniv.ac.in



SRM
UNIVERSITY
(Under section 3 of UGC Act 1956)

JavaScript, VB Script, ASP, IIS, Visio, ERWin, MS Project, Oracle, DB2, Sybase, XML. Coordinate team of 2-3 programmers. Troubleshoot, maintain existing applications. Reply to: Global Infonet, Inc., 7236 Merrill Road, Jacksonville, FL 32277.

ENGINEERING. Nokia Siemens Networks US LLC has the following exp/degree position in Irving, TX. Travel to unanticipated U.S. worksites may be required. System Specialist: Programming involving analysis of hardware/software testing and verification, fault management, competence transfer, integration testing, troubleshooting, and field trial activities. Send resumes to nokusjobs1@nokia.com, and reference ID# 07-TX-SS. Equal Opportunity Employer.

OPERATIONS RESEARCH ANALYST (Manh). Engage in biz ops analysis & dvlpmnt of computerized, intl supply chain mgmt system w/full WAN integration in all global mkts & w/manAGERIAL back-end interfaces w/suppliers to create a "pull" just-in-time supply chain & managerial acctg controls; perform on-going mgmt analysis of all biz operations. B.S. in Comp Sci or Ops Research w/5 yrs exp req. Knowl in both h/ware & s/ware:

SAS/other quantitative analyses s/ware, proficient in LAN & WAN systems, in scripting skills (PERL, shell scripting), Java or C/C++ & relational d/bases a+. Resume to Sung Su Jo, Eastern Nationwide Supply, Inc, 22 W. 32nd St, Ste 201, NY, NY 10001.

SYSTEMS ANALYST. Amgen has an opportunity for a Senior Associate Programmer/Analyst. Requires related Master's; 2 yrs. exp.; and exp. with SDLC in a regulated industry; GLP, GCP, GMP requirements, 21 CFR Part 11 regulations; flowcharting business analysis, process mapping, testing and application support; change management and configuration management. Job site: Thousand Oaks, CA. Applicants send resume and reference #6GMV4B to: Kyle Foster, Amgen, Inc., One Amgen Center Drive, Mailstop 19-1-A Thousand Oaks, CA 91320. No phone calls or e-mails please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

SOFTWARE DEVELOPER (Cimnet, Inc./Robeson, PA): responsible to enhance & maintain web based reporting tool based on mfr'g plant floor data. Work performed utilizing SQL 2000, Oracle 9i, Microsoft.Net technologies VB.Net &

C++. S/ware development will be primarily in the area of developing reports using Data Dynamic Components. Req: Masters deg in comp sci + 1yr exp in job offered or 1 yr exp as a programmer, s/ware developer or analyst. Must have significant exp developing reports using Data Dynamic Components. Must have exp w/ VB.Net, C++, Microsoft.Net, Oracle 9i & SQL 2000. 40 hrs/wk 8-5; Salary commensurate w/exp. Send resume to IEEE Computer Society, 10662 Los Vaqueros Circle, Box# COM31, Los Alamitos, CA 90720.

COMPUTERS: Senior Programmer Analyst position avail. in the Princeton, NJ area for qual. candidates. Duties include: Analyze, design, develop & test client server & web-based application software, GUI, & Object Oriented Objects. Work w/Oracle Developer, Oracle PL/SQL, EDI, Visual C++, Crystal Reports, ASP & COBOL. Travel req. & reloc. may be req. Send res. to Attn: Tom Martucci, Interpool, Inc., 211 College Road East, Princeton, NJ 08540.

ENTERPRISE APPLICATION DEVELOPER for credit union network. Job site San Dimas, CA. Send resume to Job # jsmith@wescorp.org.

ADVERTISER INDEX JUNE 2006

Advertiser		Page Number	Advertising Personnel	
Hot Chips 2007		5	Marion Delaney	Sandy Brown
IEEE Computer Society Awards		Cover 4	IEEE Media, Advertising Director	IEEE Computer Society,
IEEE Computer Society Membership		86-88	Phone: +1 415 863 4717	Business Development Manager
Microsoft Corp.		Cover 2	Email: md.ieeemedia@ieee.org	Phone: +1 714 821 8380
scitopia.org		Cover 3	Marian Anderson	Fax: +1 714 821 4010
Classified Advertising		76-80	Advertising Coordinator	Email: sb.ieeemedia@ieee.org
			Phone: +1 714 821 8380	
			Fax: +1 714 821 4010	
			Email: manderson@computer.org	
Advertising Sales Representatives				
Mid Atlantic (product/recruitment)	Midwest (product)	Midwest/Southwest (recruitment)	Northwest/Southern CA (recruitment)	
Dawn Becker	Dave Jones	Darcy Giovengo	Tim Matteson	
Phone: +1 732 772 0160	Phone: +1 708 442 5633	Phone: +1 847 498-4520	Phone: +1 310 836 4064	
Fax: +1 732 772 0161	Fax: +1 708 442 7620	Fax: +1 847 498-5911	Fax: +1 310 836 4067	
Email: db.ieeemedia@ieee.org	Email: dj.ieeemedia@ieee.org	Email: dg.ieeemedia@ieee.org	Email: tm.ieeemedia@ieee.org	
New England (product)	Will Hamilton	Southwest (product)	Southeast (product)	
Jody Estabrook	Phone: +1 269 381 2156	Steve Loerch	Bill Holland	
Phone: +1 978 244 0192	Fax: +1 269 381 2556	Phone: +1 847 498 4520	Phone: +1 770 435 6549	
Fax: +1 978 244 0103	Email: wh.ieeemedia@ieee.org	Fax: +1 847 498 5911	Fax: +1 770 435 0243	
Email: je.ieeemedia@ieee.org	Joe DiNardo	Email: steve@didierandbroderick.com	Email: hollandwfh@yahoo.com	
New England (recruitment)	Phone: +1 440 248 2456	Northwest (product)	Japan	
John Restchack	Fax: +1 440 248 2594	Peter D. Scott	Tim Matteson	
Phone: +1 212 419 7578	Email: jd.ieeemedia@ieee.org	Phone: +1 415 421-7950	Phone: +1 310 836 4064	
Fax: +1 212 419 7589	Southeast (recruitment)	Fax: +1 415 398-4156	Fax: +1 310 836 4067	
Email: j.restchack@ieee.org	Thomas M. Flynn	Email: peterd@pscottassoc.com	Email: tm.ieeemedia@ieee.org	
Connecticut (product)	Phone: +1 770 645 2944	Southern CA (product)	Europe (product/recruitment)	
Stan Greenfield	Fax: +1 770 993 4423	Marshall Rubin	Hilary Turnbull	
Phone: +1 203 938 2418	Email: flynttom@mindspring.com	Phone: +1 818 888 2407	Phone: +44 1875 825700	
Fax: +1 203 938 3211		Fax: +1 818 888 4907	Fax: +44 1875 825701	
Email: greenco@optonline.net		Email: mr.ieeemedia@ieee.org	Email: impress@impressmedia.com	



CALLS FOR IEEE CS PUBLICATIONS

IEEE Internet Computing magazine is seeking articles for a special issue on creating and managing virtual organizations. The March/April 2008 issue addresses flexible networks of independent, globally distributed entities (individuals or institutions) that share knowledge and resources and work toward a common goal.

Possible topics include the management of trust relationships in dynamic virtual organizations, quality-of-service issues, and resource allocation and provisioning.

Submissions are due by **15 July 2007**. To view the complete call for papers, visit www.computer.org/portal/pages/internet/content/cfp.html.

IEEE Computer Graphics and Applications magazine is seeking articles for a special issue on computational aesthetics. The March/April 2008 issue addresses the study of computational methods for eliciting a specified emotional response from a human.

Possible topics include image analogies, style transfer methods, sketching, visual balance, and nonphotorealistic rendering. Also welcome are papers that describe empirically based metrics of aesthetic attributes.

Submissions are due by **20 August 2007**. To view the complete call for papers, visit <http://www.computer.org/portal/site/cga/index.jsp>.

CALLS FOR PAPERS

WICSA 2008, Working IEEE/IFIP Conf. on Software Architecture, 18-21 Feb., Vancouver, Canada; Submissions due 17 Sept. www.wicsa.net

EDCC 2008, 7th European Dependable Computing Conf., 7-9 May, Kaunas, Lithuania; Submissions due 20 Sept. <http://edcc.dependability.org/call-for-contributions/call-for-papers>

CALENDAR

JUNE 2007

3-4 June: MSE 2007, Int'l Conf. on Microelectronic Systems Education (with DAC 2007), San Diego; www.mseconference.org

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.

3-6 June: SWTW 2007, 17th Ann. IEEE Semiconductor Wafer Test Workshop, San Diego; www.swtest.org

6-8 June: TASE 2007, IEEE & IFIP Int'l Symp. on Theoretical Aspects of Software Eng., Shanghai; www.sei.ecnu.edu.cn/TASE2007

11-15 June: ICAC 2007, 4th IEEE Int'l Conf. on Autonomic Computing, Jacksonville, Fla; www.autonomic-conference.org

13-15 June: Policy 2007, 8th IEEE Int'l Workshop on Policies for Distributed Systems & Networks, Bologna, Italy; www.policy-workshop.org

18 June: IWAS 2007, IEEE WoWMoM Workshop on Autonomic Wireless Access (with WoWMoM), Helsinki; www.netlab.tkk.fi/IWAS2007

18-20 June: DCOSS 2007, Int'l Conf. on Distributed Computing in Sensor Systems, Santa Fe, N.M.; www.dcss.org/dcss07/index.php

18-20 June: WETICE 2007, 16th IEEE Int'l Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Paris; www.inf.int-evry.fr/WETICE

18-21 June: WoWMoM 2007, IEEE Int'l Symp. on a World of Wireless, Mobile and Multimedia Networks, Helsinki; www.tml.tkk.fi/IEEE-wowmom/index.html

18-23 June: CVPR 2007, IEEE Conf. on Computer Vision & Pattern Recognition, Minneapolis; <http://cvpr.cv.ri.cmu.edu/>

19-20 June: SOCA 2007, IEEE Int'l Conf. on Service-Oriented Computing and Applications, Newport Beach, Calif.; <http://linux.ece.uci.edu/soca07>

21-22 June: WISES 2007, Fifth Workshop on Intelligent Solutions in Embedded Systems, Madrid, Spain; www.enti.it.uc3m.es/wises07/

25 June: VisSoft 2007, 4th IEEE Int'l Workshop on Visualizing Software for Understanding & Analysis (with ICPC), Banff, Canada; www.program-comprehension.org/vissoft07

25-27 June: Arith 2007, 18th IEEE Symp. on Computer Arithmetic, Montpellier, France; www.lirmm.fr/arith18

25-28 June: DSN 2007, Int'l Conf. on Dependable Systems & Networks, Edinburgh; www.dsn.org



Crypto 2007

The 27th International Cryptology Conference brings together researchers and practitioners of all forms of cryptology.

Organizers have solicited original research papers on the technical aspects of encryption, cryptography, decryption, and other aspects of cryptology, including software schemes and other computer-based approaches. Conference events include invited talks, refereed papers, and birds-of-a-feather sessions. An informal "rump session" features short, entertaining presentations in an informal atmosphere.

Crypto 2007 is sponsored by the IEEE Computer Society Technical Committee on Security and Privacy in cooperation with the International Association for Cryptologic Research and the Computer Science Department of the University of California, Santa Barbara.

Crypto 2007 will take place in Santa Barbara, California, from **19-23 August**.

Visit www.iacr.org/conferences/crypto2007/cfp.html for further conference details, including registration and activities information.

25-29 June: ICDCS 2007, 27th IEEE Int'l Conf. on Distributed Computing Systems, Toronto; www.eecg.utoronto.ca/icdcs07

26-29 June: ICPC 2007, 15th IEEE Int'l Conf. on Program Comprehension, Banff, Canada; www.cs.ualberta.ca/icpc2007

29 June: IWSAWC 2007, 7th Int'l Workshop on Smart Appliances and Wearable Computing (with ICDCS), Toronto; www.mis.informatik.tu-darmstadt.de/events/iwsawc2007

JULY 2007

1-4 July: ISCC 2007, 12th IEEE Symp. on Computers & Comm., Aveiro, Portugal; www.av.it.pt/iscc07

2-5 July: ICME 2007, IEEE Int'l Conf. on Multimedia & Expo, Beijing; <http://research.microsoft.com/conferences/icme07>

9-13 July: ICWS 2007, IEEE Int'l Conf. on Web Services, Salt Lake City; <http://conferences.computer.org/icws/2007>

9-13 July: SCC 2007, IEEE Int'l Conf. on Services Computing (with ICWS), Salt Lake City; <http://conferences.computer.org/scc/2007>

11-13 July: ICIS/COMSAR 2007, 6th IEEE Int'l Conf. on Computer & Information Science (with 2nd IEEE/ACIS Workshop on Component-Based Software Eng., Software Architecture, & Reuse), Melbourne; <http://acis.cps.cmich.edu:8080/ICIS2007>

11-14 July: ICECCS 2007, 12th IEEE Int'l Conf. on Eng. of Complex Computer Systems, Auckland, New Zealand; www.cs.auckland.ac.nz/iceccs07

12-13 July: DIMVA 2007, 4th Int'l Conf. on Detection of Intrusions & Malware, and Vulnerability Assessment, Lucerne, Switzerland; www.dimva2007.org

12-14 July: NCA 2007, 6th IEEE Int'l Symp. on Network Computing and Applications, Cambridge, Mass; www.ieee-nca.org

18-20 July: ICALT 2007, 7th IEEE Int'l Conf. on Advanced Learning Technologies, Niigata, Japan; www.ask.iti.gr/icalt/2007

24-27 July: COMPSAC 2007, 31st Annual Int'l Computer Software and Applications Conf., Beijing; <http://conferences.computer.org/compsac/2007>

24-27 July: EMOBS 2007, First IEEE Int'l Workshop on Eng. Mobile-Based Software and Applications (with COMPSAC), Beijing; <http://conferences.computer.org/compsac/2007/workshops/EMOBS07.html>

AUGUST 2007

19-23 Aug: Crypto 2007, 27th Int'l Cryptology Conf., Santa Barbara, Calif.; www.iacr.org/conferences/crypto2007

27-30 Aug: ICGSE 2007, Int'l Conf. on Global Software Eng., Munich; www.inf.pucrs.br/icgse

SEPTEMBER 2007

10-14 Sept: SEFM 2007, 5th Int'l Conf. on Software Eng. & Formal Methods, London; www.iist.unu.edu/SEFM07

12-13 Sept: BWA 2007, Broadband Wireless Access Workshop (with NGMAST), Cardiff, Wales, UK; <http://bwaws.nginet.de>

15-19 Sept: PACT 2007, 16th Int'l Conf. on Parallel Architectures & Compilation Techniques, Brasov, Romania; <http://parasol.tamu.edu/pact07>

17-20 Sept: Cluster 2007, IEEE Int'l Conf. on Cluster Computing, Austin, Texas; www.cluster2007.org

20-21 Sept: ESEM 2007, Int’l Symp. on Empirical Software Eng. & Measurement, Madrid; www.esem-conferences.org

24-26 Sept: GSEM 2007, 4th Int’l Conf. on Grid Services Engineering and Management, Leipzig, Germany; www.ict.swin.edu.au/conferences/gsem2007

24-27 Sept: MSST 2007, 24th IEEE Conf. on Mass Storage Systems & Technologies, San Diego; <http://storageconference.org/2007>

30 Sept-1 Oct: SCAM 2007, 7th IEEE Int’l Working Conf. on Source Code Analysis and Manipulation, Paris; www2007.ieee-scam.org

OCTOBER 2007

2-5 Oct: ICSM 2007, 23rd IEEE Int’l Conf. on Software Maintenance, Paris; <http://icsm07.ai.univ-paris8.fr>

3-5 Oct: CRITIS 2007, 2nd Int’l Workshop on Critical Information Infrastructures Security, Malaga, Spain; <http://critis07.lcc.uma.es>

8-10 Oct: WiMob 2007, 3rd IEEE Int’l Conf. on Wireless & Mobile Computing, Networking, & Comm., White Plains, N.Y.; www.gel.usherbrooke.ca/WiMob2007

9-11 Oct: ATS 2007, 16th Asian Test Symp., Beijing; <http://ats07.ict.ac.cn>

10-12 Oct: Tabletop 2007, 2nd IEEE Int’l Workshop on Horizontal Interactive Human-Computer Systems, Newport, R.I.; www.ieeetabletop2007.org

10-13 Oct: FIE 2007, Frontiers in Education Conf., Milwaukee, Wis.; www.fie-conference.org/fie07

11-13 Oct: IPC 2007, Int’l Conf. on Intelligent Pervasive Computing, Jeju, Korea; www.sersc.org/IPC2007

11-13 Oct: ISWC 2007, Int’l Symp. on Wearable Computers, Boston; <http://iswc.net>

12-13 Oct: WRTLT 2007, 8th Workshop on RTL & High-Level Testing (with ATS), Beijing; <http://wrtlt07.ict.ac.cn>

14-21 Oct: ICCV 2007, 11th IEEE Int’l Conf. on Computer Vision, Rio de Janeiro; www.research.rutgers.edu/~iccv2007

15-17 Oct: BIBE 2007, IEEE 7th Symp. on Bioinformatics & Bioengineering, Boston; www.cs.gsu.edu/BIBE07

Events in 2007

JULY 2007

- 1-4ISCC 2007
- 2-5ICME 2007
- 9-13ICWS 2007
- 9-13SCC 2007
- 11-13ICIS/COMSAR 2007
- 11-14ICECCS 2007
- 12-13DIMVA 2007
- 12-14NCA 2007
- 18-20ICALT 2007
- 24-27COMPSAC 2007
- 24-27EMOBS 2007

AUGUST 2007

- 19-23Crypto 2007
- 27-30ICGSE 2007

SEPTEMBER 2007

- 10-14SEFM 2007
- 12-13BWA 2007
- 15-19PACT 2007
- 17-20Cluster 2007
- 20-21ESEM 2007
- 24-26GSEM 2007
- 24-27MSST 2007
- 30 Sept-1 Oct:SCAM 2007

15-18 Oct: LCN 2007, 32nd IEEE Conf. on Local Computer Networks, Dublin, Ireland; www.ieeelcn.org

15-18 Oct: LCN ON-MOVE 2007, IEEE Workshop On User Mobility and Vehicular Networks (with LCN), Dublin, Ireland; www.ieeelcn.org

15-19 Oct: EDOC 2007, 15th IEEE Enterprise Distributed Object Computing Conf., Annapolis, Maryland; www.edoconference.org

15-19 Oct: RE 2007, 15th IEEE Int’l Requirements Eng. Conf., Delhi, India; www.re07.org

16-19 Oct: CIT 2007, IEEE 7th Int’l Conf. on Computer and Information Technology, Aizu, Japan; www.u-aizu.ac.jp/conference/cit07

20-23 Oct: FOCS 2007, IEEE 48th Ann. Symp. on Foundations of Computer Science, Providence, R.I.; www.focs2007.org

24-27 Oct: SBAC-PAD 2007, 19th Int’l Symp. on Computer Architecture and High-Performance Computing, Gramado, Brazil; www.sbc.org.br/sbac/2007



Society Introduces New Technical Task Forces

The IEEE Computer Society Technical Activities Board recently created two new collaborative bodies for furthering participation in society activities. Computer Society Technical Committees, Technical Councils, and Task Forces form an international network of professionals who share common interests in computer hardware, software, applications, and other related fields. They serve as the focal point for the Computer Society's activities within a technical discipline and directly influence Society policy on standards development, conferences, publications, and educational initiatives.



IEEE COMPUTER SOCIETY TASK FORCE ON HAPTICS

The IEEE Technical Committee and Task Force on Haptics was founded in 2006 under the joint sponsorship of the IEEE Computer Society and the IEEE Robotics and Automation Society (where it enjoys full Technical Committee status). Initiated by a group of haptics researchers, the TC/TF on Haptics is home to the international interdisciplinary haptics research community. The group provides leadership and organization for scientific work in haptics, which covers a wide range of disciplines from engineering to neurophysiology.

The new haptics group will coordinate the scheduling of major haptics conferences; sponsor special conference sessions, tutorials, and journal issues on haptics; and contribute to the new *IEEE Transactions on Haptics*.

Founding chair of the new group is Hong Z. Tan of Purdue University. Cochairs are Matthias Harders of the Swiss Federal Institute of Technology in Zurich and Hiroyuki Kajimoto of the University of Electro-Communications in Tokyo.

Membership in the IEEE Haptics TC/TF is open to all individuals interested in haptics research at a professional level. There are no fees for membership, and IEEE membership is not required for joining the group. Visit www.worldhaptics.org to learn more.

Editor: Bob Ward, bnward@computer.org

IEEE COMPUTER SOCIETY TASK FORCE ON GAME TECHNOLOGY

As the field of computer-based gaming has matured in the past few years, the genre known as "serious" games has grown in number and purpose. More than 30 years of unob-

structed development in a highly competitive commercial market has yielded game technologies and design processes that can be used in applications outside entertainment.

The new IEEE Computer Society Task Force on Game Technology advocates formalizing the collaborative frameworks that contribute to the existing grassroots cohesion of the serious games and gaming communities. In particular, task force organizers suggest that applied game technology can result in better software applications, especially in areas of collaboration, user interface, collective intelligence, visualization, and artificial intelligence.

Founding chair of the IEEE Task Force on Game Technology is James R. Parker of Canada's University of Calgary. Parker is the author of *Start Your Engines* (Paraglyph, 2005), the first book published on the art of designing and developing driving and racing games.

Learn more about the IEEE Task Force on Game Technology at www.ucalgary.ca/~jparker/TFGT.

Apply Now for UPE and Larson Scholarships

Each year, the IEEE Computer Society offers scholarships to both graduate and undergraduate Society student members. Two opportunities for student support—the Upsilon Pi Epsilon Student Award for Academic Excellence and the Lance Stafford Larson Student Scholarship—seek applicants by **31 October**.

UPSILON PI EPSILON STUDENT AWARD FOR ACADEMIC EXCELLENCE

Presented by the IEEE Computer Society in conjunction with international computing honor society Upsilon Pi Epsilon, the Upsilon Pi Epsilon Student Award for Academic Excellence recognizes

Computer Science Enrollments Drop in 2006

The percentage of incoming undergraduates among all degree-granting institutions who indicated they would major in computer science and related fields declined by 70 percent between 2000 and 2005, according to the Computing Research Association's annual Taulbee Survey of PhD-granting computer science and computer engineering departments in North America.

The number of students who declared their major in computer science and related fields among the PhD-granting departments surveyed by the CRA also fell. After six years of declines, the number of new majors in 2006 was nearly half of what it was in 2000 (15,958 versus 7,798). This is a slight decline from the 7,952 new majors reported in 2005. Overall enrollments in computer science and engineering dropped 14 percent between 2004/2005 and 2005/2006, to 34,898. Overall, enrollments have dropped 39 percent from their height in 2001/2002.

These declines in enrollment are, predictably, being reflected at the other end of the pipeline. Following several years of increases, the total number of bachelor's degrees awarded by PhD-granting computer science departments fell 28 percent between 2003/2004 and 2005/2006, to 10,206. The median number of degrees granted per department declined 30 percent (to 48). The sustained drop in total enrollment combined with waning student interest in computer science or engineering as a major suggests that degree production numbers will continue to drop in the near term.

A steep drop in degree production among computer science departments has happened before. According to the National Science Foundation, the number of undergraduate computer science degrees granted each year nearly quadrupled between 1980 and 1986 to more than 42,000. This period was followed by a swift decline and leveling off during the 1990s, with several years in which the number of degrees granted hovered around 25,000. During the late 1990s, undergraduate computer science degree production again surged to more than 57,000 in 2004. In light of the economic downturn and slow job growth of the early 2000s, the CRA has projected for several years the current decline in degree production.

The Taulbee Survey is named for the late Orrin E. Taulbee of the University of Pittsburgh, who launched the survey in 1974 for the Computer Science Board (predecessor of the CRA) and conducted it until 1984. The survey is a key source of information on the enrollment, production, and employment of PhDs in computer science and computer engineering. It also provides salary and demographic data for computer science and computer engineering faculty in North America. Results from the Taulbee Survey can be compared with data produced by the National Science Foundation, which surveys all institutions that grant computer science degrees.

Full results are posted each May on the CRA Web site at www.cra.org/statistics. ■

high achievement in the computing discipline.

The UPE scholarship is awarded based on a student's academic record, letters of recommendation, and extra-curricular involvement related to the computing field. Any Society member who is a full-time undergraduate or graduate student with a minimum 3.0 GPA—the required GPA for Upsilon Pi Epsilon membership—can apply. Up to four awards of \$500 each are given each year.

LARSON BEST PAPER CONTEST

The Lance Stafford Larson Student Scholarship awards \$500 to a Computer Society student member for the best paper submitted on a computer-related topic. A competitive scholarship, it was established in memory of Lance Larson, the son of former IEEE presi-

dent Robert Larson, and a University of Maryland undergraduate at the time of his death. The Larson competition was created to encourage engineering students to improve their communication skills. Only papers concerning computer-related subjects are eligible. Papers will be judged on technical content, writing skill, and overall presentation. Any undergraduate student member with a GPA of 3.0 or above is welcome to compete. First-, second-, and third-place winners also receive a certificate of commendation.

Recipients of either honor also enjoy a complimentary one-year subscription to any Computer Society periodical of their choice. For information on entering either contest, see www.computer.org/students/schlrshp.htm.

Helping you lead the way and manage change in
computer, software, and information technology.

IEEE
Computer
Society

HOW QUICKLY CAN YOU FIND THE RIGHT SOLUTION?



JOIN
NOW!



VERY QUICKLY with help from these cutting-edge
resources readily available to you from the IEEE Computer Society ...

*"Absolutely fantastic! The
Distance Learning Campus is
one of the best benefits the
Computer Society can give
its members. I plan to take
a number of these courses."*

Enrique Madrona, Engineering Manager

- FREE Subscription to *Computer Magazine*
- FREE Online Access to 500 IT Books and Technical Articles*
- FREE Online Access to 1,300
Self-Paced IT Training Courses*
- Discounted Publications in Your Area of Expertise
- Career-Enhancing Volunteer Committees

*New IT books and training courses from Element K® become available in January 2007.
Members get access to other online books and courses for the remainder of 2006.

NOT A MEMBER?
DON'T MISS OUT!
Join today at www.computer.org/join



2007 RATES for IEEE COMPUTER SOCIETY

Membership Dues and Subscriptions

Membership and periodical subscriptions are annualized to and expire on 31 December 2007. Pay full or half-year rate depending upon the date of receipt by the IEEE Computer Society as noted below.

Membership Options*

All prices are quoted in U.S. dollars.

	FULL YEAR Applications received 16 Aug 06–28 Feb 07	HALF YEAR Applications received 1 Mar 07–15 Aug 07
<input type="checkbox"/> I do not belong to the IEEE and I want to join only the Computer Society:	<input type="checkbox"/> \$110.00	<input type="checkbox"/> \$55.00
<input type="checkbox"/> I want to join both the Computer Society and the IEEE:		
I reside in the USA	<input type="checkbox"/> \$209.00	<input type="checkbox"/> \$104.50
I reside in Canada	<input type="checkbox"/> \$190.00	<input type="checkbox"/> \$95.00
I reside in Africa/Europe/Middle East	<input type="checkbox"/> \$182.00	<input type="checkbox"/> \$91.00
I reside in Latin America	<input type="checkbox"/> \$175.00	<input type="checkbox"/> \$87.50
I reside in Asia/Pacific	<input type="checkbox"/> \$176.00	<input type="checkbox"/> \$88.00
<input type="checkbox"/> I already belong to the IEEE, and I want to join the Computer Society:	<input type="checkbox"/> \$48.00	<input type="checkbox"/> \$24.00

Are you now or were you ever a member of the IEEE?
☐ Yes ☐ No If yes, please provide member # if known: _____

Add Periodicals**

	ISSUES PER YEAR	FULL YEAR Applications received 16 Aug 06–28 Feb 07 PRINT + ONLINE	HALF YEAR Applications received 1 Mar 07–15 Aug 07 PRINT + ONLINE
BEST VALUE!			
IEEE Computer Society Digital Library (online only)	n/a	<input type="checkbox"/> \$119	<input type="checkbox"/> \$60
ARTIFICIAL INTELLIGENCE			
IEEE Intelligent Systems	6	<input type="checkbox"/> \$41	<input type="checkbox"/> \$21
IEEE Transactions on Pattern Analysis and Machine Intelligence	12	<input type="checkbox"/> \$49	<input type="checkbox"/> \$25
BIOTECHNOLOGY			
IEEE/ACM Transactions on Computational Biology and Bioinformatics	4	<input type="checkbox"/> \$35	<input type="checkbox"/> \$18
COMPUTATION			
Computing in Science & Engineering	6	<input type="checkbox"/> \$45	<input type="checkbox"/> \$23
COMPUTER HARDWARE			
IEEE Computer Architecture Letters	4	<input type="checkbox"/> \$28	<input type="checkbox"/> \$14
IEEE Micro	6	<input type="checkbox"/> \$39	<input type="checkbox"/> \$20
IEEE Design & Test of Computers	6	<input type="checkbox"/> \$39	<input type="checkbox"/> \$20
IEEE Transactions on Computers	12	<input type="checkbox"/> \$45	<input type="checkbox"/> \$23
GRAPHICS & MULTIMEDIA			
IEEE Computer Graphics and Applications	6	<input type="checkbox"/> \$42	<input type="checkbox"/> \$21
IEEE MultiMedia	4	<input type="checkbox"/> \$37	<input type="checkbox"/> \$19
IEEE Transactions on Visualization and Computer Graphics	6	<input type="checkbox"/> \$39	<input type="checkbox"/> \$20
HISTORY OF COMPUTING			
IEEE Annals of the History of Computing	4	<input type="checkbox"/> \$33	<input type="checkbox"/> \$17
INTERNET & DATA TECHNOLOGIES			
IEEE Internet Computing	6	<input type="checkbox"/> \$42	<input type="checkbox"/> \$21
IEEE Transactions on Knowledge and Data Engineering	12	<input type="checkbox"/> \$47	<input type="checkbox"/> \$24
IT & SECURITY			
IT Professional	6	<input type="checkbox"/> \$40	<input type="checkbox"/> \$20
IEEE Security & Privacy	6	<input type="checkbox"/> \$24	<input type="checkbox"/> \$12
IEEE Transactions on Dependable and Secure Computing	4	<input type="checkbox"/> \$31	<input type="checkbox"/> \$16
MOBILE COMPUTING			
IEEE Pervasive Computing	4	<input type="checkbox"/> \$41	<input type="checkbox"/> \$21
IEEE Transactions on Mobile Computing	12	<input type="checkbox"/> \$40	<input type="checkbox"/> \$20
NETWORKING			
IEEE Transactions on Parallel and Distributed Systems	12	<input type="checkbox"/> \$43	<input type="checkbox"/> \$22
SOFTWARE			
IEEE Software	6	<input type="checkbox"/> \$47	<input type="checkbox"/> \$24
IEEE Transactions on Software Engineering	12	<input type="checkbox"/> \$38	<input type="checkbox"/> \$19

All prices are in U.S. dollars. Periodicals purchased at member prices are for the member's personal use only.

Payment Information

Payment required with application

Membership fee
\$

Periodicals total
\$

Applicable sales tax***
\$

TOTAL:
\$

Enclosed:

☐ Check/Money Order****

Charge my:

- ☐ MasterCard
☐ VISA
☐ American Express
☐ Diner's Club

Card Number

Exp Date (month/year)

Signature

USA Only include
5-digit billing zip code

* Member dues include \$19 for a 12-month subscription to Computer.

** Periodicals purchased at member prices are for the member's personal use only.

*** Canadian residents add 14% HST or 6% GST to total. AL, AZ, CO, DC, GA, IN, KY, MD, MO, NM, and WV add sales tax to periodical subscriptions. European Union residents add VAT tax to IEEE Computer Society Digital Library subscription.

**** Payable to the IEEE in U.S. dollars drawn on a U.S. bank account. Please include member name and number (# known) on your check.

Allow up to 8 weeks for application processing. Allow a minimum of 6 to 10 weeks for delivery of print periodicals.

Please complete both
sides of this form.

For fastest service,
apply online at
www.computer.org/join

Personal Information

Enter your name as you want it to appear on correspondence. As a key identifier in our database, circle your last/surname.

☐ Male ☐ Female

Date of birth (Day/Month/Year)

Title

First name

Middle

Last/Surname

Home address

City

State/Province

Postal code

Country

Home telephone

Home facsimile

Preferred e-mail

Send mail to: ☐ Home address ☐ Business address

Educational Information

First professional degree completed

Month/Year degree received

Program major/course of study

College/University

State/Province

Country

Highest technical degree received

Program/Course of study

Month/Year received

College/University

State/Province

Country

Business/Professional Information

Title/Position

Years in current position

Years of practice since graduation

Employer name

Department/Division

Street address

City

State/Province

Postal code

Country

Office phone

Office facsimile

I hereby make application for Computer Society and/or IEEE membership and agree to be governed by IEEE's Constitution, Bylaws, Statements of Policies and Procedures, and Code of Ethics. I authorize release of information related to this application to determine my qualifications for membership.

Signature

Date

IF7F

NOTE: In order for us to process your application, you must complete and return BOTH sides of this form to the office nearest you:

Asia/Pacific Office
IEEE Computer Society
Watanabe Bldg.
1-4-2 Minami-Aoyama
Minato-ku, Tokyo 107-0062 Japan
Phone: +81 3 3408 3118 • Fax: +81 3 3408 3553
E-mail: tokyo.ofc@computer.org

Publications Office
IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314 USA
Phone: +1 800 272 6657 (USA and Canada)
Phone: +1 714 821 8380 (worldwide) • Fax: +1 714 821 4641
E-mail: help@computer.org

BPA Information

This information is used by society magazines to verify their annual circulation. Please refer to the audit codes and indicate your selections in the box provided.

A. Primary line of business

1. Computers

2. Computer peripheral equipment

3. Software

4. Office and business machines

5. Test, measurement, and instrumentation equipment

6. Communications systems and equipment

7. Navigation and guidance systems and equipment

8. Consumer electronics/appliances

9. Industrial equipment, controls, and systems

10. ICs and microprocessors

11. Semiconductors, components, sub-assemblies, materials, and supplies

12. Aircraft, missiles, space, and ground support equipment

13. Oceanography and support equipment

14. Medical electronic equipment

15. OEM incorporating electronics in their end product (not elsewhere classified)

16. Independent and university research, test and design laboratories, and consultants (not connected with a manufacturing company)

17. Government agencies and armed forces

18. Companies using and/or incorporating any electronic products in their manufacturing, processing, research, or development activities

19. Telecommunications services, telephone (including cellular)

20. Broadcast services (TV, cable, radio)

21. Transportation services (airlines, railroads, etc.)

22. Computer and communications, and data processing services

23. Power production, generation, transmission, and distribution

24. Other commercial users of electrical, electronic equipment, and services (not elsewhere classified)

25. Distributor (reseller, wholesaler, retailer)

26. University, college/other education institutions, libraries

27. Retired

28. Others (allied to this field)

B. Principal job function

1. General and corporate management

2. Engineering management

3. Project engineering management

4. Research and development management

5. Design engineering management— analog

6. Design engineering management— digital

7. Research and development engineering

8. Design/development engineering— analog

9. Design/development engineering— digital

10. Hardware engineering

11. Software design/development

12. Computer science

13. Science/physics/mathematics

14. Engineering (not elsewhere classified)

15. Marketing/sales/purchasing

16. Consulting

17. Education/teaching

18. Retired

19. Other

C. Principal responsibility

1. Engineering or scientific management

2. Management other than engineering

3. Engineering design

4. Engineering

5. Software: science/management/engineering

6. Education/teaching

7. Consulting

8. Retired

9. Other

D. Title

1. Chairman of the Board/President/CEO

2. Owner/Partner

3. General Manager

4. V.P. Operations

5. V.P. Engineering/Director Engineering

6. Chief Engineer/Chief Scientist

7. Engineering Manager

8. Scientific Manager

9. Member of Technical Staff

10. Design Engineering Manager

11. Design Engineer

12. Hardware Engineer

13. Software Engineer

14. Computer Scientist

15. Dean/Professor/Instructor

16. Consultant

17. Retired

18. Other Professional/Technical

Computer

Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

qMags

BOOKSHELF

Object-Oriented Analysis and Design with Applications, 3rd ed., Grady Booch, Robert A. Maksimchuk, Michael W. Engel, Bobbi J. Young, Jim Conallen, and Kelli A. Houston. The third edition of this venerable reference to object-oriented technology can help readers learn to apply OO methods using new paradigms such as Java, the Unified Modeling Language 2.0, and .NET.

The authors draw upon their extensive experience to offer improved methods for object development, along with many examples that tackle the complex problems software engineers face. These include systems architecture, data acquisition, cryptanalysis, control systems, and Web development. The book covers essential concepts, explains the method, and shows its successful application in several fields. It also offers pragmatic advice on issues such as classification, implementation strategies, and cost-effective project management.

New material in this edition includes an introduction to UML 2.0, from the notation's most fundamental and advanced elements, with an emphasis on key changes; new domains and contexts; and an examination of the conceptual foundation for the widely misunderstood fundamental elements of the object model, such as abstraction, encapsulation, modularity, and hierarchy.

Addison-Wesley; www.awprofessional.com; 0-201-89551-X; 720 pp.

Formal Models of Operating System Kernels, Iain D. Craig. This book shows that the formal specification of kernels is necessary for operating systems to achieve the levels of reliability and security demanded of them today. The author includes specifications for a sequence of increasingly complex kernels that can serve as models to help developers identify and reason about the design's properties—thus making explicit what is too often left implicit or even unknown.

The author explores what can be inferred about a design through rigorous reasoning. He also describes



essential properties of data structures and mechanisms. Designers can easily become bogged down in complexity issues when considering kernels. This book's clear and concise style, and its prescriptive rather than descriptive approach, shed light on this topic, showing clearly how an operating system's kernel can affect these systems' reliability and performance.

Springer; www.springer.com; 1-84628-375-2; 338 pp.

Creating Agile Business Systems with Reusable Knowledge, Amit Mitra and Amar Gupta. Developers need agility and innovation to achieve global excellence and customer value in 21st-century business, yet most approaches to business process engineering sacrifice these attributes in favor of operational efficiency and economics. Moreover, the IT systems used to automate and encapsulate business processes are unresponsive to the dynamic business environment.

The authors strive to close this gap, showing how innovation can be systematized with normalized patterns of information, how business processes and information systems can be tightly aligned, and how these processes and systems can be designed to automatically adapt to change by reconfiguring shared patterns of knowledge. They also present a modular approach to building business systems that parallels that of object-oriented software and provide the practical templates required for accelerating integration, analysis, and design.

Cambridge University Press; www.cambridge.org; 0-521-85163-7; 404 pp.

Probabilistic Methods for Financial and Marketing Informatics, Richard E. Neapolitan and Xia Jiang. This book shows how to apply informatics to areas such as managerial options and decision making, investment science, marketing,

and data mining. The authors concentrate on the probabilistic and decision-theoretic approaches to informatics, emphasizing the use of Bayesian networks.

Rather than dwelling on rigor, algorithms, and proofs of theorems, the book focuses on problem solving and practical applications. Many examples and exercises can be found throughout the book, as well as six chapters that walk the reader through pragmatic situations. Many solutions are expanded on when the authors discuss their final implementation, which uses the Netica software package.

Morgan Kaufmann; www.mkp.com; 0-12-370477-4; 432 pp.

How the Body Shapes the Way We Think: A New View of Artificial Intelligence, Rolf Pfeifer and Josh Bongard. The authors demonstrate that thought is not independent of the body, but tightly constrained and, at the same time, enabled by it. They argue that the kinds of thoughts we can have are predetermined by their foundation in our embodiment: in our morphology and the material properties of our bodies.

This crucial notion underlies fundamental changes in the field of artificial intelligence over the past two decades. The authors use the basic methodology of artificial intelligence—understanding by building—to describe their insights. If we understand how to design and build intelligent systems, they reason, we will better understand intelligence in general. In accessible, nontechnical language, with many examples, they introduce the basic concepts, drawing from recent developments in robotics, biology, neuroscience, and psychology to outline a possible theory of intelligence. They illustrate their applications of such a theory in ubiquitous computing, business and management, and the psychology of human memory.

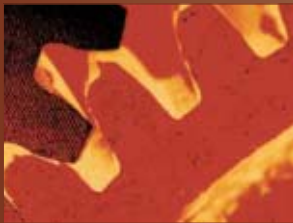
MIT Press; mitpress.mit.edu; 0-262-16239-3; 394 pp.

Send book announcements to
newbooks@computer.org.

HOW THINGS WORK

Binary Arithmetic

Neville Holmes, University of Tasmania



The basic aspects of computer arithmetic are how numbers are represented and the operations performed on those representations.

Digital technology works so well because at the heart of digital representation there are only a few basic components. Nowadays, these basic components usually are binary digits, *bits* for short, called binary because they are designed to stand for only two different values, conveniently called zero and one. A stored or transmitted bit's state can deteriorate quite badly before a processing device will be mistaken in deciding which of the two possible values is the original.

Most modern computers store the data uniformly in blocks in their main store. The blocks are numbered serially so that each block has its own number—its *address*. Thus, each block has two aspects: its address and its content. Addresses are a simple sequence of serial numbers. The content located at a specific address is the value of its bits, usually eight in number and called a *byte*, divided into two groups of four contiguous bits called *nibbles*.

As each bit can independently be either 0 or 1, a byte can hold 2^8 or 256 different values overall. There are too many of these to conveniently give a name to each, so bytes are usually con-

sidered as divided into nibbles of *hexadecimal digits*—*hex digits* for short. Table 1 lists binary digits and their hexadecimal equivalent.

For clarity, the hex digits beyond 9 are sometimes spoken as *able*, *baker*, *charlie*, *dog*, *easy*, and *fox*.

The value that any byte stores can then be shown as a pair of these hex digits. Strings of bytes can represent anything you need represented digitally. Being able to represent anything depends on having adopted a convention for representing things of that kind.

The earliest and still quite popular class of thing to be represented is numbers, though properly speaking these are not things but properties of things. The numerical operations that a computer carries out on numbers as numbers are together called its arithmetic.

ADDRESS ARITHMETIC

Working with a computer's data addresses requires only the simplest arithmetic, called address arithmetic or logical arithmetic. The arithmetic is simple because there is only a limited number of addresses and these are unsigned—that is, never negative—and start at zero.

If a computer used an address of one byte, it would have a main store of 256 bytes with addresses from 0 to 255 (decimal) or 00 to FF (hex). A 2-byte address would allow addresses from 0 to 65,535 or 0000 to FFFF, a 3-byte address 0 to 16,777,215 or 000000 to FFFFFFFF, and so on. Nowadays, computers usually have addresses larger than this. However, a one-byte address will serve here to illustrate logical arithmetic.

The main operation of logical arithmetic is addition. Addition is a series of steps starting with an augend and an addend. For simplicity, we use an 8-bit byte to illustrate the process, although 32-bit and 64-bit words are more common today.

Each step computes three new quantities: an augend, a carry, and an addend. The steps use all n bits of the operands in parallel (here $n = 8$). The new augend bit is 1 if and only if (denoted iff) one and only one of the two incoming bits is a 1 (that is, iff corresponding bits of the augend and addend are 01 or 10). The new carry bit is 1 iff both incoming bits are 1. The adder forms the new addend by shifting the carry bits one position to the left and putting a 0 in the right-most bit position. The adder repeats the three computations of a step (new augend, carry, and addend) until all the carry bits are 0. Figure 1 illustrates the example of adding 85 and 103 (decimal).

Quirks

The time taken for a straightforward addition depends on the number of steps needed, which can vary widely. However, shortcuts can remove this dependency.

When adding two numbers, the high order carry bit could be a 1. This is simply lost in shifting. The effect is like the hour-hand of a clock passing through 12, except that in the hexadecimal example the clock would have 256 hours labeled 00 to FF.

Circuitry could be provided for subtraction, but it's simpler to add the complement. For a hexadecimal clock, this hinges on 1 back, say, being the same as 255 forward, and vice versa.

Table 1. Hexadecimal digits.			
Binary	Hex	Binary	Hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

For the binary representation of a byte, the complement is its bitwise complement with a one added in to the lowest position.

Logical arithmetic is error free. An address can be invalid, but only because the main store is not big enough to hold a value at every possible address, a fault that virtual addressing eliminates.

INTEGER ARITHMETIC

When possible values are quantitative and not indicative, the arithmetic must be able to handle negative numbers and to multiply and divide with them.

An early method of representing negativity was to use the leading bit as a simple arithmetic sign, but this raised the ambiguity of having two different zeroes, one negative and one positive.

The usual method nowadays is to offset the values in such a way that the arithmetic is not directly concerned with negativity, but the leading bit nevertheless acts as a negative sign. Thus the values for a single byte binary integer range from -128 (80) to -1 (FF) and from 0 (00) to 127 (7F). The four examples of addition in Figure 2 illustrate this, showing that the operation is practically the same as for logical arithmetic, even though the meanings have been changed.

Subtraction is carried out by adding the negation of the subtrahend. Multiplication is carried out by repeated addition, and division by repeated subtraction. Shortcuts are used to speed up these operations. Simple operations such as negation

Step	Operand	Binary	Hex	Decimal
Add decimal 85 + 103				
	Augend	01010101	55	85
	Addend	01100111	67	103
	New augend bit is 1 iff incoming bits are 01 or 10			
1a	Augend	00110010	32	50
	Carry bit is 1 iff incoming bits are 11			
1b	Carry	01000101	45	69
	New addend is now the left-shifted carry with 0 fill			
1c	Addend	10001010	8A	138
	Repeat the three substeps until the carry is all zeroes			
2a	Augend	10111000	B8	184
2b	Carry	00000010	02	2
2c	Addend	00000100	04	4
	Repeat the three substeps until the carry is all zeroes			
3a	Augend	10111100	BC	188
3b	Carry	00000000	00	0
Finished because carry is all zeroes; answer is in the last augend				

Figure 1. Address arithmetic example. The arithmetic is simple because there is only a limited number of addresses and these are never negative.

Example	Binary	Hex	Decimal
(a) 85 + 39 = 124			
Augend	01010101	55	85
Addend	00100111	27	39
Result	01111100	7C	124
(b) 85+ (-39) = 46			
Augend	01010101	55	85
Addend	11011001	D9	-39
Result	00101110	2E	46
(c) (-85) + 39 = -46			
Augend	10101011	AB	-85
Addend	00100111	27	39
Result	11010010	D2	-46
(d) (-85) + (-39) = -124			
Augend	10101011	AB	-85
Addend	11011001	D9	-39
Result	10000100	84	-124

Figure 2. Integer addition examples. The values for a single byte binary integer range from -128 (80) to -1 (FF) and from 0 (00) to 127 (7F).

and magnitude are typically also provided in a computer's instruction set.

Quirks

Just as for address arithmetic, the results are exact if they will fit into the space provided for the result in the result register. However, because the values are no longer cyclic, the result of an addition can be longer than that space. If adding two positive numbers results in an apparently negative number, then the correct sum is too long for the result register to hold. This is called an *overflow* and must be signaled so that a program can deal with this exceptional result. An apparent nonnegative result from adding two negative numbers is also an overflow. Negation and magnitude of the lowest representable number will also cause an overflow, which will need to be

HOW THINGS WORK

Sign	Exponent	Significand	Binary	Decimal
0	1000 0000	01000000. . . 0	$+1.01 \times 2^{128-127}$	$(1 + .25) \times 2^1 = 2.5$
1	1000 0001	11000000. . . 0	$-1.11 \times 2^{129-127}$	$-(1 + .5 + .25) \times 2^2 = -7$
0	0111 1111	00000000. . . 0	$+1 \times 2^{127-127}$	1

Figure 3. Examples of scaled values. The most significant bit is the sign of the number (0 indicates positive, and 1 indicates negative). The next eight bits are the exponent in base 2, expressed as an integer. The 8-bit exponent value is biased by +127, which makes the representable range -127 to +128. The significand is termed “normalized” because its arithmetic value is always less than 2 but not less than 1.

signaled. In the example using eight bits, -128 cannot be negated because +128 is larger than the largest positive number representable in eight bits (+127). While a program can satisfactorily deal with an occasional overflow, simple multiplication would overflow far too often to be tolerated. The usual way to deal with this is to couple two result registers to provide a double-length product. It’s then up to the program to use the two halves appropriately.

Integer division is more complicated still. First, while the quotient, as an integer, usually can be accommodated, the division will leave a remainder, so two result registers are needed. Second, the quotient can in effect overflow when the divisor is zero or when the lowest representable integer is divided by negative one.

SCALED ARITHMETIC

In scientific and engineering computing, multiplications and divisions are as frequent as additions and subtractions. Repeated multiplication, as in polynomial evaluation, is common. Such computation requires scaling. Before electronic computers became available, scientists and engineers commonly used slide rules and logarithm tables for multiplication and division, and they did scaling mentally or with the help of pencil and paper. Although Konrad Zuse’s early computers used automatic scaling, later machines, such as John von Neumann’s IAS computer, did not. This forced the programmer to anticipate what scaling would be needed, although the IAS machine used 40-bit numbers to protect against the unexpected.

When the unexpected proved all too frequent, users built scaling into their programs but, because this was very slow, scientific computers soon did their scaling in hardware. Rather unfortunately, such arithmetic and the representation of scaled values came to be called *floating-point* arithmetic and numbers. The adjective *semilogarithmic* is sometimes used for clarity.

A floating-point number has three parts: the base *b*, the scale or exponent *e*, and the significand *s*. The value represented is $s \times b^e$. The exponent and the significand are variable in floating-point representations, but the base is fixed. Scaled values are printed out using a base of 10, so 45E6 represents 45,000,000 and 4.5E-6 represents 0.0000045. Internally, most computers use binary floating-point arithmetic and representation in which the base is 2.

IEEE Standard 754 for Binary Floating-Point Arithmetic specifies the format of floating-point numbers for both single-precision (32-bit) and double-precision (64-bit) representations. For simplicity, we consider only the single-precision format. The standard says that the most significant bit is the sign of the number (0 indicates positive, and 1 indicates negative). The next eight bits are the exponent in base 2, expressed as an integer. But because the number’s true exponent must be allowed to be positive or negative, the 8-bit exponent value is biased by +127, which makes the representable range -127 to +128. The remaining 23 bits are the mantissa, also called the significand. Those 23 bits are used as fraction bits appended to an implied integer of 1, sometimes

called the “hidden” bit. The significand is termed “normalized” because its arithmetic value is always less than 2 but not less than 1.

In floating-point multiplication the exponents are added and the significands multiplied. The exponent and significand of the result might need slight adjustment to bring the integer part for the hidden bit back to 1 before the leading 23 fraction bits of the product’s significand, perhaps with rounding, are stored in the result. Division is handled in much the same way, but using subtraction on the exponents and full division (without remainder) on the significands.

Addition and subtraction are quite complex because the values of the exponents must be used to adjust the alignment of the two significands before the addition or subtraction can take place. For further explanation of floating-point arithmetic, see wikipedia.org/wiki/floating_point.

Quirks

The most significant quirk of scaled arithmetic is the loss of exactness. Basic laws of arithmetic no longer hold. For example,

$$(a + b) - a = a + (b - a) = b$$

in exact arithmetic but, because the significand of any result needs to be truncated or rounded in floating-point arithmetic before it is stored, the result of $(a + b) - a$ might not be the same as that of $a + (b - a)$. Such errors can accumulate significantly when a program carries out trillions of floating-point operations. Sophisticated use of interval arithmetic can avoid this problem, but this requires the type of rounding to be selectable so that it can preserve the interval properties.

Because the significand is normalized, there is no straightforward way to represent zero. A tweak is needed. Once this tweak is provided, the various possible results of division by zero need tweaks in turn to represent them. For example, different representations are needed for $0 \div 0$, $1 \div 0$ and $-1 \div 0$. The arithmetic’s various

operations must be able to handle all these special values in combination with each other and with ordinary values. To handle these special cases, exponent values of all zeroes and all ones are reserved to signal special values such as 0, denormalized numbers, infinity, and not-a-number. Thus, for ordinary arithmetic the exponent actually only has a range of -126 to +127.

Floating-point arithmetic is not only still subject to overflow when a result becomes too large to represent, but a result also can be too small to represent, an exception called *underflow*.

COMPLETE ARITHMETIC

Traditional floating-point arithmetic tolerates the introduction of error, but the errors tend to accumulate in unpredictable ways. In the past, providing longer and longer representations lessened the error, but this is a losing battle as computers become faster and faster and problems larger and larger. The result of a large scientific computation might now need many trillions of floating-point operations.

Such computations typically include focused subsections traversing very large arrays of values to arrive at only a few results such as the sum of products. The truncation error within such subsections can be unpredictably large, but it can be eliminated by using a result register large enough to keep the complete result, which is exact. With computers as capacious as today's, that exact result can then be kept as is for intermediate results, pressing it into floating-point format only for final results.

Complete arithmetic was available more than 20 years ago as a special feature for an IBM mainframe computer, but was later removed from sale. Perhaps it was before its time, as implementation in a microprogram was relatively slow.

With today's integrated circuitry, complete arithmetic has become quite practical and has been satisfactorily implemented on special chips. Its widespread adoption is overdue as its support for a branch of computation

called *validated numerics* is crucial.

Validated numerics can provide solutions to most scientific computations with certainty. Prominent among its techniques is a sophisticated application of interval arithmetic, arithmetic that works with paired values that specify the bounds within which an entity's value must lie. Control of rounding type ensures that values stay within their bounds. Mathematicians can design algorithms so that convergence of intervals is proof that the solution is completely valid. Complete arithmetic can greatly speed up convergence, and it can induce convergence that would not be possible with traditional floating-point arithmetic.

Quirks

While complete arithmetic can greatly reduce the incidence of overflows and underflows, it cannot completely eliminate them. Valid results can be too large or small to be represented even in a complete result register—for example, in the unlikely event of repeated exponentiation of extreme values. Such invalidity is not of practical concern, however, because it can be completely avoided in its most particular use—the ubiquitous scalar or dot product. A more significant problem is the inability to fit extreme complete results into standard floating-point formats.

Large arrays of complete results need large amounts of storage space and time to store and fetch, although using a variable-length format for external storage of complete results could greatly reduce both the space and time needed. Nevertheless, programs need to keep down the number of times they convert complete results to floating-point format.

An arithmetic and representation such as symmetric-level indexing, which compares to semilogarithmic arithmetic somewhat the way semilogarithmic compares to integer, can eliminate overflow and underflow. The drawback is the severe complexity of arithmetic operations.

Perhaps a more serious problem with binary arithmetic is the isolation

of scaled arithmetic from integer arithmetic. This isolation means that programmers must make decisions about which to use when coding programs, and making such decisions is not always easy. Also, they might need to write several versions of a computational function for different representations of its arguments. An early Burroughs mainframe computer in which an integer representation would switch to scaled rather than signaling an overflow implemented such an arithmetic.

Further discussion of these possibilities can be found in "Composite Arithmetic" (*Computer*, Mar. 1997, pp. 65-73). Ulrich Kulisch's *Advanced Arithmetic for the Digital Computer: Design of Arithmetic Units* (Springer-Verlag, 2002) provides a description of complete and interval arithmetics and their implementation. ■

Neville Holmes is an honorary research associate at the University of Tasmania's School of Computing. Contact him at neville.holmes@utas.edu.au.

Computer welcomes your submissions to this bimonthly column. For additional information, or to suggest topics that you would like to see explained, contact column editor Alf Weaver at weaver@cs.virginia.edu.

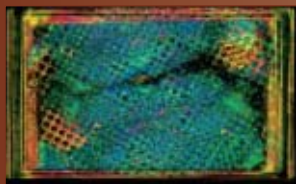
Join the
IEEE
Computer Society
online
at

www.computer.org/join/

ENTERTAINMENT COMPUTING

iPhones Target the Tech Elite

Michael Macedonia



Apple's iPhone will entice early adopters with the cachet of coolness.

The new Apple iPhone is scheduled to debut in June 2007, 30 years to the month after the Apple II, the world's first popular PC, appeared. As if it were a comet we have been speculating about and tracking over the past two years, the iPhone is no surprise, but we still can't keep our eyes off it. Like the Apple II, the iPhone will arrive with an impact that affects several domains—financial, technical, and cultural. Although felt immediately, understanding its full impact will take years.

Strategically, the iPhone will help Apple protect and expand its base of iPod devices into the domain of smart phones and converging devices. On the surface, the iPhone is an iPod with an integrated phone. There is, of course, much more.

TECHNICAL MARVEL

The iPhone's technical specifications are impressive (www.apple.com/iphone/technology/specs.html). In the tradition of Apple, the iPhone emphasizes aesthetics over features. The device is thin at 11.6 mm, but provides a large, clean display.

The iPhone lacks keys, instead using a 3.5-inch, 480 × 320 pixel touch screen for interacting with the system

features. The phone's lack of tactile feedback, other than its two hardware controls—the sleep switch and home button—has provoked much discussion in design and UI blogs. For example, the iPhone's screen will likely fall prey to smudges and fingerprints. Moreover, the display must be lighted to access the simulated keys and other inputs. However, Apple has had plenty of experience with virtual controls such as the click-wheel interface on the iPod and seems confident users will accept using the human finger as a stylus for the iPhone.

The device also has a plethora of wireless capabilities, including multiple flavors of GSM, Bluetooth 2.0, and WiFi 802.11b/g. The phone will use AT&T's EDGE technology for two-and-a-half generation (2.5G) data rates of around 250 Kbps, which is much slower than competing 3G technologies such as evolution data only (EVDO). But the primary broadband access method will be the iPhone's Wi-Fi capability, with 50 Mbps in the 802.11g mode.

The iPhone provides small but useful elements such as a proximity sensor to turn the display on and off to save energy because it's not helpful to light up the display if the user isn't

positioned to see it. An accelerometer reorients an image depending on how the user holds the handset. The iPhone also extends the Mac OS X franchise by using a smaller, embedded version of the BSD Unix variant. This might make the iPhone attractive to third-party developers if Apple decides to open up the device.

USER FRIENDLY

From a user perspective, the device incorporates a 2-megapixel camera, a Web browser, and iTunes. All the device's communications, computing, and graphics require serious computing horsepower and batteries. The iPhone will have four ARM processors, and Apple claims that battery life will be around five hours for talking, watching videos, and Web browsing on the Internet.

The iPhone represents mature technology that several other mobile phones or PDAs have demonstrated. What makes it different is that the device's aesthetic appeal hides much of its complexity, as Figure 1 shows. Moreover, the design integrates well with the vertical market Apple has created for TV, music, and movies. The phone syncs with iTunes just like an iPod, and its software works like (and with) the Mac's.

The iPhone follows a pattern of what Clive Grinyer (www.clivegrinyer.com/sitebuildercontent/sitebuilderfiles/lipstickonapig2007.pdf) calls "strategic design"

Steve Jobs is just as much the designer, defining the values, creating the environment for Jon [Ives] and his team to deliver, and using design strategically to design not just great products but a complete service experience—design across every touchpoint.

REVENUE GENERATOR

The rationale for Apple to create the iPhone is that the financial impact for the company, its supply chain partners, and consumers could be huge. Steve Jobs said at Macworld this January: "We've ... had

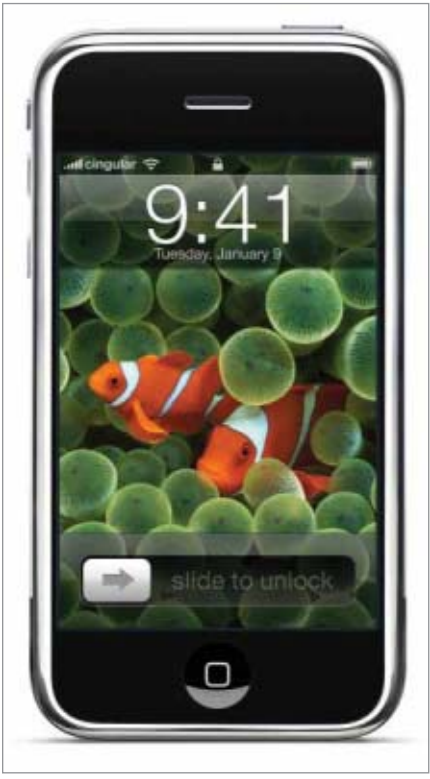


Figure 1. Built atop a foundation of mature technology that other mobile devices and PDAs have demonstrated, the iPhone’s aesthetic appeal hides much of its complexity. Photo courtesy of Apple.

some real revolutionary products. The Mac in 84, the iPod in 2001, and we’re gonna do it again with the iPhone in 2007. Exactly what we’re trying to do, 1 percent market share in 2008, 10 million units and we’ll go from there....” That’s more than \$5 billion in revenues for the mobile phone-camera-iPod combination. The initial cost of the iPhone follows iPod’s stair-step pricing model (www.roughlydrafted.com/RD/RDM.Tech.Q1.07/4DD0941D-9097-4FAE-A3BB-29DA5CA07199.html), as Table 1 shows.

Services, excluding iTunes purchases, will take the bigger chunk. AT&T, the mobile-phone service provider, will likely charge between \$75 and \$100 per month for a two-year plan. This will provide the cell-phone giant with revenues of up to \$2,400 per phone and could raise the total cost of an iPhone to well above \$3,000 over two years.

Purchasers will thus need to be fairly affluent to afford such a device—a market that Apple clearly desires. Moreover, those affluent buyers will generate new revenue from music, video, and movie downloads.

COOLNESS FOR THE WIN

Version 1.0 of a product is always a market test, albeit an expensive one. In the iPhone’s case, the real target date is 2009, when the second version will come out and the first set of two-year service agreements with AT&T will expire. By that time, AT&T will finally have fully deployed 3G phone service in the US. Apple will then also have its production geared up to start selling different models for different market segments.

Apple also plans to release new features for free during the two-year agreement cycle. This will give it the opportunity to test-market services—in a sense, echoing Google’s beta/new-feature approach for its Web applications.

The breadth and depth of Apple’s dominance in the online media market is stunning. Consumers have made more than 2 billion downloads from its iTunes store since it debuted in 2003. The store now has more than 5 million songs, 350 TV shows, and 500 movies in its catalog. iTunes holds

85 percent of the market share of legal downloads and represents a major revenue growth area for Apple. The 10 million iPhones will soon join the more than 100 million iPods Apple has already sold.

The iPhone faces some of the same criticism as the early iPod did in the MP3 market—too expensive given a market full of similarly capable but pedestrian music players. Yet the world continues to buy iPods more than any other MP3 player because they offer something others can’t: a bite of Apple’s cool factor, generated by the company’s great design and a fabulous user experience.

The iPhone promises to build on Apple’s coolness legacy. Moreover, it could quickly become known as *the* elite experience in a world that already has 10 billion mobile phones. Apple’s initial 10 million iPhones will represent barely a tenth of one percent of the world’s cell phone market. Early adopters will thus be special, members of the tech elite.

If the iPhone meets expectations, it could create a new, unpredictable dynamic in the mobile phone marketplace. Already, for example, Sprint reduced the cost of its music downloads in March from \$2.50 a song to 99 cents.

Returning to the comet analogy, the iPhone could have as much of a disruptive effect on the existing cell phone market as the dinosaur-killing comet’s impact did 65 million years ago. Or, as with the Apple II in 1977, it could be the beginning of a new age for devices we haven’t named yet. ■

Table 1. Apple’s iPod stair-step pricing model.		
iPod Model	Price	Description
Shuffle	\$79	Ultra small and simple
Nano	\$149-\$249	Very small and thin
5G iPod	\$249-\$349	Hard-drive-based, with large capacity
iPhone	\$499-\$599	Phone and Internet features

Michael Macedonia is a member of Computer’s editorial board. Contact him at macedonia@computer.org.

Social Scripting for the Web

Tessa Lau
IBM Almaden Research Center



Koala makes it possible to capture and share how-to knowledge with others.

As our business and personal lives move online, we must learn to carry out increasingly complicated tasks on the Web—for example, checking a bank account balance, setting up automatic bill payments, sharing photos with family, searching for real-estate listings, and ordering new business cards. Wouldn't it be nice to have an expert watching over your shoulder and showing you how to do such things properly?

The Koala project at IBM's Almaden Research Center (www.research.ibm.com/koala) is aiming to create the next best thing: a wiki-type repository of instructions for Web applications that can help users automate common tasks.

In addition to being human-readable, Koala scripts are machine-understandable—the system can interpret each instruction and perform it automatically. At each step, Koala shows you what button to push and then does it for you. It can also fill in fields with your name, address, and other personal information.

SOCIAL SCRIPTING

Unlike knowledge bases where an individual or organization is respon-

sible for single-handedly documenting all procedures, Koala lets any user contribute a script to the repository. Just as Wikipedia leverages the expertise of people around the world to create a comprehensive encyclopedia of human knowledge, we envision Koala becoming a community-driven repository of how-to knowledge about Web applications.

This approach, which we call *social scripting*, relies on users to contribute scripts that benefit other members of their community. It might be particularly useful in an enterprise setting, where members have common needs such as purchasing equipment via the corporate procurement system or following a standard process for transferring an employee to a different department.

Koala also addresses the "long tail" of business processes: In a corporation where the IT department can afford to build tools to automate popular processes such as payroll, staffing, and benefits administration, many processes are not automated because doing so isn't cost-effective (<http://thelongtail.com>). Koala fills this need by enabling small teams to easily automate their own idiosyncratic workflows.

Asking ordinary users to document how-to knowledge is challenging, but you don't have to be a programmer to create scripts that others can use. Koala employs a *programming by demonstration* technique that lets non-programmers generate scripts automatically. As a user navigates through a Web site and fills in fields, Koala records the steps in a human-readable language consisting of instructions such as "click the Search button" and "type your name into the From field."

In this way we hope to lower the bar for creating scripts, making it possible for anyone who knows how to do something on the Web to easily document those steps and show others how to do it as well.

PRINTING PHOTOS ONLINE

Koala makes it easy to learn how to do something online. For example, Figure 1 shows how Koala can be used to print digital photos via an online service.

The Koala sidebar appears on the left side of the Web browser and displays the script "Print photos with Kodak EasyShare," comprising six steps, while the right side of the browser displays the photo service Web page. Koala highlights the next step to be followed in the script—clicking the "Buy Prints" link—along with the corresponding widget in the Web page. When the user clicks the "Step" button, Koala performs the corresponding action and advances to the next step.

The user can continue stepping through the script until it completes. A user who decides to do something different can simply abandon the script in midexecution or return to it later by clicking on a step and resuming execution from that point.

Some instructions require human input and thus can't be executed automatically. Koala parses each step to determine whether it can be automated or the user must be consulted. To recognize those steps the user should perform, the system looks for the word "you" in the instruction—a simple but extremely effective heuristic.

When Koala encounters a “you” step, it pauses execution and waits for the user to perform the described action. In the photo-printing scenario, for example, the user must choose which photos to print. After making a selection, the user can click the “Step” button to resume execution at the next step, adding these photos to the shopping cart.

This powerful *mixed-initiative interaction* feature can be used to partially automate processes that require human decisions and judgment.

Once a user has stepped through a script to see how it works, Koala also provides a “Run” button that will execute the entire script without stopping (unless it contains a “you” step, at which point execution will pause and wait for human input). Using this feature, people can automate routine tasks. For example, one could use a Koala script to automatically forward office phone calls to a different number when working remotely. Koala could also be used to automate testing of Web applications.

SEARCHING FOR REAL ESTATE

Creating new scripts is as simple as demonstrating how to perform a given task in a Web browser. Figure 2 illustrates an example of using Koala to create a script that searches for real estate in the San Francisco Bay Area. As the user visits Web pages and clicks buttons, Koala records these actions in the “Steps” field of the sidebar.

The script itself is text, and the script editor is a text editor. Users can rearrange steps or change them as needed simply by editing the text in the script. The process doesn’t require users to employ a precise programming language syntax; the system is capable of interpreting colloquial instructions such as “now scroll down and press the Go button.”

Koala’s approach to parsing plain-text commands, nicknamed “sloppy programming,” relies on two key principles: Actions on Web pages may be described with a fairly small vocabulary (click, push, type, enter, and so on), and the set of possible tar-

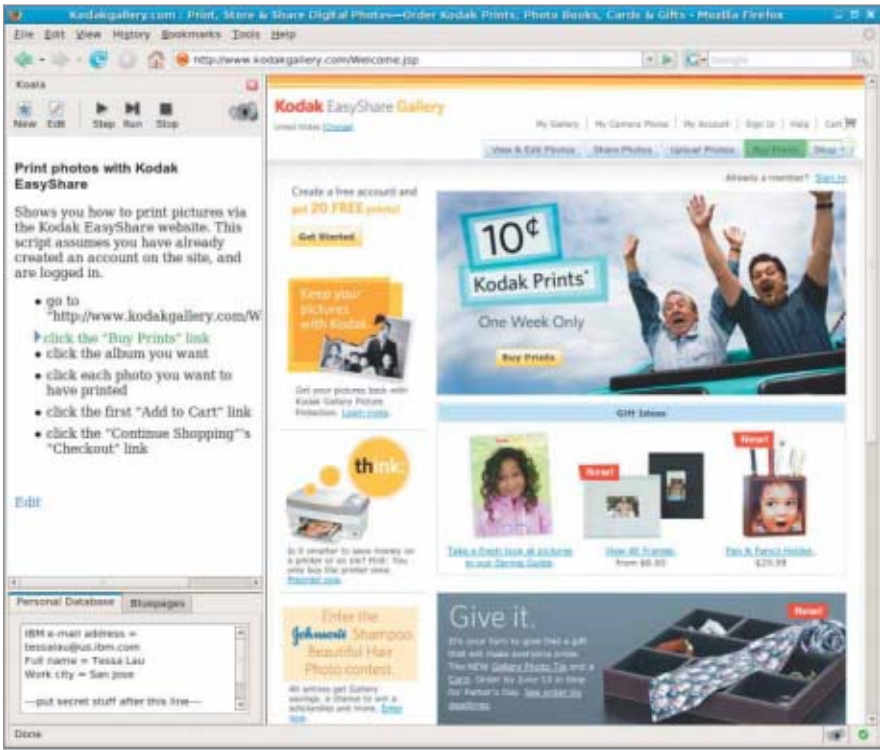


Figure 1. Using Koala to print digital photos. Koala highlights the script for the next step to be followed in the left side of the browser along with the corresponding widget in the printing service’s Web page.

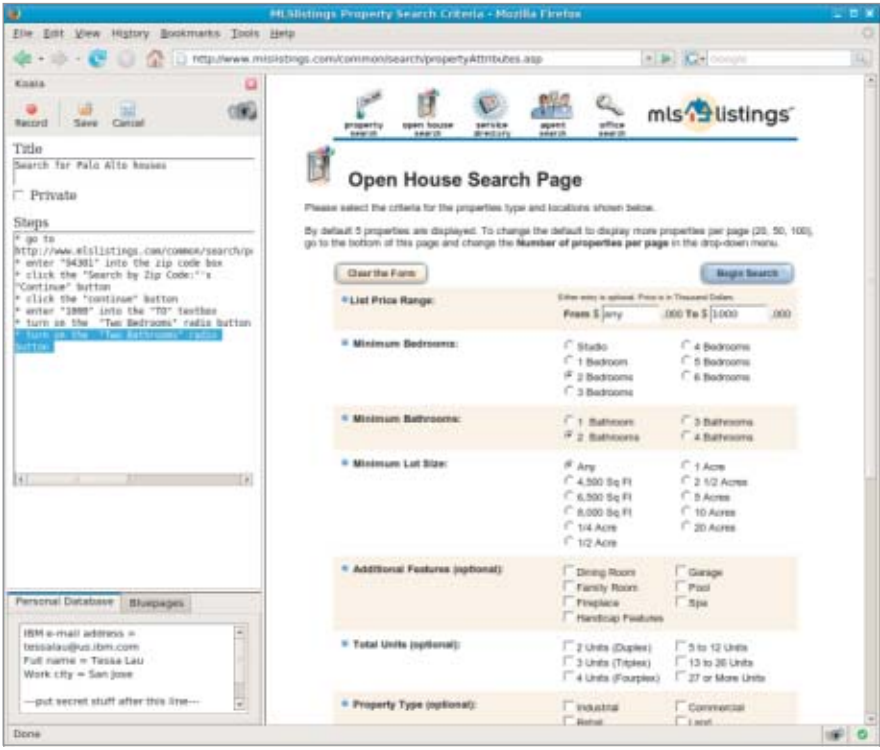


Figure 2. Using Koala to record a script for real-estate searches. As the user visits Web pages and clicks buttons, Koala records these steps in the browsers’ left-hand pane.

gets is highly constrained by the current Web page—for example, there is typically only one search button to click.

INVISIBLE COMPUTING

A user clicks the “Save” button to automatically store a created script on the project’s wiki, code-named Koalescence, for others to use as well. The fact that the script is represented purely as text makes it easy for others to read and edit the script in the same way that Wikipedia authors can correct and add details to one another’s articles.

Generalizing scripts

Koala also provides a way to generalize scripts so that they work across users with the “Personal Database” feature shown in the lower left corner of Figures 1 and 2. This database is simply a text file containing name/value pairs, which users can refer to in scripts.

For example, when a user enters an e-mail address in a form, rather than recording the literal action, Koala generalizes the instruction to read “enter your email address into the Email: field.” When playing back that script,

Koala will search the user’s personal data field, extract the e-mail address, and substitute that into the step.

This simple feature lets people create scripts that do not hard-code any personal details but can be customized at runtime for the user running the script.

CREATING A COMMUNITY

After a script has been uploaded to Koalescence, any user can load it into Koala and run it. This lets users automate routine online tasks. For example, instead of having to check the real-estate search engine manually each day, a person could run a Koala script, and with a single click be taken directly to the listings that match the desired search criteria.

However, what is routine for one person might be instructive for another. The scripts a user creates to solve a particular set of problems might be even more useful in a social

context. This is the reason social bookmarking sites such as del.icio.us (<http://del.icio.us>) have become so popular: Although people might bookmark Web sites primarily for personal recall, a side effect of sharing those bookmarks is to contribute to a public repository of tagged and filtered information that everyone can search.

Similarly, while we expect Koala to be used initially as a personal tool, over time the shared script repository will become more valuable as people start to leverage one another’s work.

To ensure Koala continues to develop into a comprehensive repository of how-to knowledge, we are taking steps to grow the community. One of the first will be to encourage users to assume different roles. However, creating scripts is only the tip of the iceberg: Some users could contribute by testing scripts and reporting when they fail, while others could tag scripts to create a folksonomy that makes scripts easier to find. A select few could serve as script gardeners, digging out the obsolete scripts and encouraging budding scripts to mature.

As our work evolves, we anticipate learning much more about what makes the Koala community thrive and what kinds of tools are most useful to help it evolve. ■

Tessa Lau is a research staff member at IBM’s Almaden Research Center. Contact her at tessalau@us.ibm.com.

My thanks to the members of the Koala team for sharing the dream: Allen Cypher, Clemens Drews, Eser K Erdogan, Eben Haber, James Lin, Jeffrey Nichols, Eric Wilcox, and honorary member Greg Little. I also thank all the early adopters who have used the system and given valuable feedback.

Editor: Bill N. Schilit, Google;
bill.schilit@computer.org,
<http://schilit.googlepages.com>

Computer Wants You

Computer is always looking for interesting editorial content. In addition to our theme articles, we have other feature sections such as Perspectives, Computing Practices, and Research Features as well as numerous columns to which you can contribute. Check out our author guidelines at

www.computer.org/computer/author.htm

for more information about how to contribute to your magazine.



ACM Digital Library

www.acm.org/dl

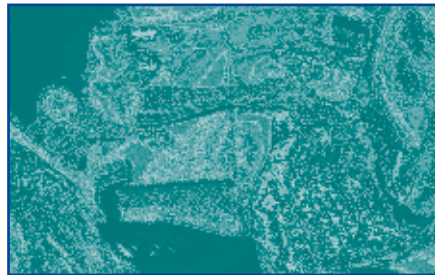
The Ultimate Online INFORMATION TECHNOLOGY Resource!



Powerful and vast in scope, the **ACM Digital Library** is the ultimate online resource offering unlimited access and value!

The **ACM Digital Library** interface includes:

- **The ACM Digital Library** offers over 40 publications including all ACM journals, magazines, and conference proceedings, plus vast archives, representing nearly 2 million pages of text. The ACM DL includes full-text articles from all ACM publications dating back to the 1950s, as well as third-party content with selected archives. www.acm.org/dl
- **The Guide to Computing Literature** offers an enormous bank of over one million bibliographic citations extending far beyond ACM's proprietary literature, covering all types of works in computing such as journals, proceedings, books, technical reports, and theses! www.acm.org/guide
- **The Online Computing Reviews Service** includes reviews by computing experts, providing timely commentary and critiques of the most essential books and articles.



Available only to ACM Members.
Join ACM online at www.acm.org/joinacm

To join ACM and/or subscribe to the Digital Library, contact ACM:

Phone: 1.800.342.6626 (U.S. and Canada)
+1.212.626.0500 (Global)
Fax: +1.212.944.1318
Hours: 8:30 a.m.-4:30 p.m., Eastern Time

Email: acmhelp@acm.org
Join URL: www.acm.org/joinacm
Mail: ACM Member Services
General Post Office
PO Box 30777
New York, NY 10087-0777 USA



Association for
Computing Machinery

Advancing Computing as a Science & Profession

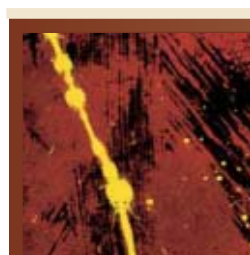
AD28

*Guide access is included with Professional, Student and SIG membership. ACM Professional Members can add the full ACM Digital Library for only \$99 (USD). Student Portal Package membership includes the Digital Library. Institutional, Corporate, and Consortia Packages are also available.

SOFTWARE TECHNOLOGIES

An Era of Change-Tolerant Systems

Shawn Bohner
Virginia Tech



Technologies are available to help develop and evolve systems that respond more quickly and efficiently to a changing environment.

The ancient Romans' ability to build cities was one mark of their technological advancement. After authorities identified a strategically suitable location, city developers would go there to design and plan for a city. To produce an effective plan, an architect would determine available resources such as water, building materials, suitable land for farming, and road access. These resources would define the number of people the city could sustain over time.

Builders would construct the city's walls first to determine the boundary for the maximum number of inhabitants and establish points from which its citizens could effectively defend the city. Artisans would erect key buildings and the requisite infrastructure from the architect's plans.

As the city prospered, more people would come to live in the area. While walls inhibited growth, residents would eventually build dwellings and lay out farms outside the walls despite the risk of potential attacks. City administrators would import goods to compensate for the lack of local resources; however, the city would be

vulnerable to external forces interfering with its supply lines. Ironically, the city's success could ultimately contribute to its downfall.

Successful software systems often suffer the same fate that befell prosperous ancient cities. Developers design software systems based on needs and constraints imposed by external factors that change over time. These systems might align well with the current mission, marketplace, or line of business, drawing a strong client base. But as the number of users grows, so do demands on the system.

These demands can imperil the system's performance by exceeding the intended resource level. Demands may be to develop capabilities not supported by the original system design.

In these cases, the pressure is to change the software in response to evolving requirements. Ultimately, developers address these demands and change the software within unrealistic resource constraints. Like building outside the city walls, this means potentially compromising resources and structures. With repeated changes, the software becomes less changeable—making the system brittle. Thus, the

software system's success could contribute to its demise.

While software engineers face a situation similar to that of their ancient city-building ancestors, technologies are available to help them develop and evolve systems that respond quickly and efficiently to a changing environment.

MANAGING COMPLEXITY AND CHANGE

Software complexity is the degree to which software is difficult to analyze, understand, or explain. Figure 1 illustrates a trend that has persisted since the mid-1970s: As society increasingly depends on software, the size and complexity of software systems continues to grow—making them progressively more difficult to understand and evolve.

This trend has dramatically accelerated in recent years with the advent of Web services, agent-based systems, autonomic and self-healing systems, reconfigurable computing, and other advances. Software's complexity has compounded in both volume (structure) and interaction (social) as the Internet has enabled delivering software functionality as services.

Yet, most technologies that we use to develop, maintain, and evolve software systems do not adequately cope with complexity and change.

Traditionally, software engineers respond to complexity by decomposing systems into manageable parts to accommodate the sheer number of elements and their structure. However, the Internet and the emergence of software as services have led to a new kind of complexity.

What José Luiz Fiadeiro describes as software's *social complexity* naturally arises from an increase in both the number and intricacy of system interactions ("Designing for Software's Social Complexity," *Computer*, Jan. 2007, pp. 34-39). Services are inherently social, and interactions stem from a range of dependencies and values.

Service-oriented architectures accordingly reflect the need for flexibility and self-assembly more than size and structure.

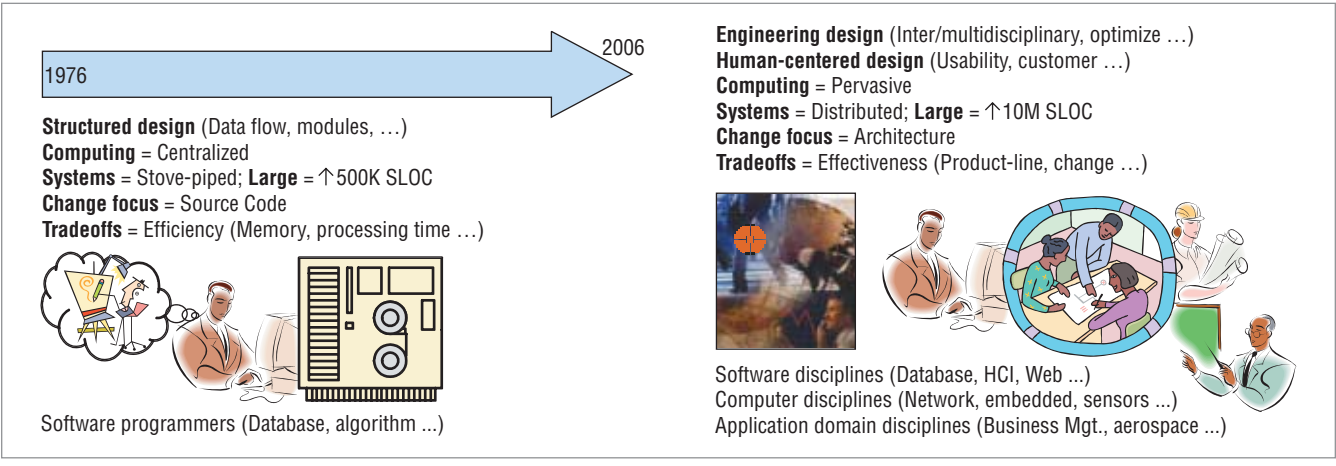


Figure 1. Software growth and complexity, 1976-2006. As society increasingly depends on software, software systems' size and complexity grow.

Web services

Consider, for example, a global firm that produces a monthly publication—originally in English—that it must translate into various languages for its customers worldwide. The firm employs a Web service broker to acquire and assemble the services necessary to translate the manuscript and distribute the electronic copies to the offices in each country.

Assuming there are competing suppliers for the translation services and acceptable translations of the documents are feasible, the supplier then submits the job to the service broker, who in turn identifies the appropriate provider based on the customer's profile and costs of the service. The customer agrees to the provision, and the provider translates and sends the publication.

At this point, if developers could construct the software service components for this specific provision, software impact analysis and visualization tools could readily provide the traditional structure and dependency depictions necessary to understand the system. However, the missing aspects are the business interaction dependencies necessary to resolve problems that might arise.

Continuing with this example, suppose the French translation is error-ridden and must be corrected. Who fixes the problem? Not the customer—the opportunity to operate

the service has passed with the initial use. Not the maintainer—since the code and even the executable image is inaccessible for local remedy. While in the same situation, the broker can at least negotiate a solution, but will it be done in time?

Because the process does not adequately capture the various interactions and their intricacies, problems arise around responsibility. Understanding the dependencies and relationships between the provisioned services and providers is daunting. With the broker often unclear on knowledge boundaries, discovering the problem using normal impact analysis can be out of reach.

This example shows, in a simple way, how the state of the practice for software impact analysis technology fails to address this new social complexity.

Autonomic and self-healing systems

Similar examples can be given for autonomic and self-healing systems. Understanding the number and intricacies of their interactions likewise provides insight into appropriate responses in a changing environment.

For autonomic systems, Michael Hinchey and colleagues suggest that micro and macro interactions with the environment as well as between members are key for technologies like swarms ("Swarms and Swarm

Intelligence," *Computer*, Apr. 2007, pp. 111-113).

For self-healing systems, anomaly detection, diagnosis, replacement planning, and execution timing are all functions of components' interactions in the operating environment as illustrated in David Garlan and colleagues' work with the Rainbow framework ("Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure," *Computer*, Oct. 2004, pp. 46-54).

With these emergent technologies, software engineers must cope with ever-increasing interactions and dependencies.

CHANGE-TOLERANCE SUPPORT

During the 1990s, the software industry shifted from a custom-solution paradigm to mass customization in packaged applications, and it is now transitioning to a service-oriented paradigm. However, this is not to say that custom solutions or mass customization have gone away.

Given that computing hardware is viewed as a commodity and the Internet makes delivery trivial, the economic weight now falls on using modular components assembled into evolving solutions—services composed based on canonical components from competing sources. Economically viable software components can be standardized and reused on many levels of scale.

SOFTWARE TECHNOLOGIES

A key aspect of software is its capacity or tolerance for change. Inspired by aspects of fault tolerance, *change tolerance* connotes software's ability to evolve within the bounds of its original design—the degree to which software change is intentional.

A maintenance view of corrective, adaptive, and perfective change is one type of software change. However, this type of change doesn't really manage the variant and invariant nature found in Bertrand Meyer's open/closed principle—open for extension, closed for modification (*Object-Oriented Software Construction*, Prentice Hall, 1988). Designing for change at the product level such as reconfigurable computing or at the process level such as model reuse are other types of software change.

Industry approaches software change using top-down model-based methods such as the Object Management Group's model-driven architecture and bottom-up agile methods such as extreme programming. Both address the risks of producing large volumes of software on shorter timelines, but from different perspectives.

Through a series of elaborations and refinements, model-based approaches systematically move from abstract computationally independent models, to platform-independent models, to concrete platform-specific models—organizing knowledge and leveraging reuse at appropriate levels. The complexities include interactions, mappings, and transforms in the populated model repositories that evolve over time.

In contrast, through a series of short, well-orchestrated releases, agile approaches employ proven techniques

such as test-driven development, refactoring, and pair programming to reduce risk and deliver value—changing software in manageable increments and leveraging the strengths of people working together.

Model-based and agile approaches are proving to be effective ways to develop and evolve software systems. Although both of these methods require considerable visibility into a product's complex nature to get it right, neither method specifically addresses the number and intricacy of interactions.

ANALYZING AND VISUALIZING SOFTWARE IMPACTS

The complexity of today's software systems often exceeds human comprehension. Automated support for analyzing and visualizing software impacts and navigating software artifacts is no longer a luxury. Understanding software impacts makes it easier to design, implement, and change software: Tradeoffs become clear, ripple effects become more certain, and estimates become more accurate.

Software-change impact analysis (SCIA) has largely been associated with software maintenance. Yet, software changes occur from the first day of development. The more artifacts that are produced, the more complexity becomes an issue, and the more engineers need instruments to see and understand what they are doing.

SCIA has evolved from the source-code-centric analyses demonstrated with the Y2K and Euro currency conversion efforts a decade ago. Since then, it has continued to incorporate

more software artifacts and semantically rich representations.

Employing information retrieval and search technologies has revealed new ways of identifying and reasoning about impacts through traceability relationships. Using change histories to show temporally related modifications from the past offers insight into potential change-tolerant design strategies for the future.

Perhaps the most significant advance in impact analysis is the use of software visualization technologies to illuminate patterns in software artifacts. Visualization reduces the perceived complexities of software and thereby helps engineers better analyze, understand, or explain aspects of software systems.

Whether using it to navigate the myriad mappings and transforms in model-driven architecture, to clarify a design refactoring in extreme programming, or discern the impacts of a maintenance change, the combination of SCIA and visualization provides an essential software technology.

Ultimately, clarifying underlying software interactions and dependencies aids the software community in designing more change-tolerant software as we move to support emerging technologies like Web services, autonomic and self-healing systems, and reconfigurable computing. These and other future advances will tax human capacity to visualize and navigate the system. Hence, software technologists must provide relevant instruments to make effective changes to software. ■

Shawn Bohner is an associate professor in the Department of Computer Science at Virginia Tech. Contact him at sbohner@vt.edu.

**Renew your
IEEE Computer Society
membership today!**

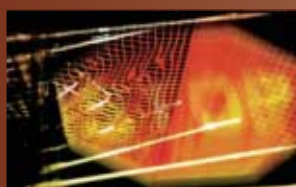
www.ieee.org/renewal

**Editor: Michael G. Hinchey,
Loyola College in Maryland;
mhinchey@loyola.edu**

SECURITY

Discription: Internal Hard-Disk Encryption for Secure Storage

Laszlo Hars, Seagate Research



Internal encryption to protect the confidentiality of stored data in disk drives has many advantages.

There have been many recent cases of information getting into unauthorized hands from lost or stolen laptops or insiders accessing unattended enterprise computers or storage devices. The Privacy Rights Clearinghouse maintains a long list of such reported cases at <http://privacyrights.org/ar/ChronDataBreaches.htm>.

Providing physical protection and using remote locations are two means of keeping stored data confidential. The least expensive secure-storage systems use local data encryption with optional data authentication, together with access control and physical tamper detection. Such devices, encrypting disk drives, are now being mass-produced after a period of sampling. Manufacturers are deploying them by large numbers in laptops, desktop PCs, data-center applications, portable media players, and TV broadcast video recorders.

The IEEE P1619 Security in Storage Working Group (<http://siswg.org>) is developing standard architectures for external encryption modules and tape drives. However, there's no standard yet for hard disks, specifying how developers can adapt the data layout to security needs and provide access control to the encrypted data.

That means an attacker can only see the ciphertext after disassembling the drive and examining the magnetic platters with multimillion-dollar equipment. And because of the attacks' destructive nature, if the disk drive is returned, the owner will notice the disk was tampered with and won't trust the stored information. This effectively renders all kinds of data-modification attacks harmless.

ADOPTING A DISCRPTION STANDARD

Adoption and utilization of a secure-disk architecture standard

would offer a number of advantages, including

- freeing an implementer from custom-designing a security architecture;
- reducing development costs and time to market by avoiding the expensive and time-consuming security analysis necessary for a proprietary solution;
- providing a secure architecture that has already met public scrutiny;
- increasing trust levels, since non-profits are viewed as more open than for-profit companies; and
- giving OEMs a second source of drives with similar security attributes.

The proposed *IEEE P1619 Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices* deals with the security of information in general storage devices that randomly read or write data in fixed-sized blocks. Its basic assumption is that attackers can access the stored data.

Discription can mitigate these inherent risks. The proposed P1619 standard describes a transparent encryption module that developers can insert into the data path without modifying the data layout. This restriction doesn't apply to discription, however, because developers can easily and transparently employ hidden disk areas or longer physical records. Hard disks can also accept new security-related interface commands.

EXTERNAL ENCRYPTION MODULES

The main drawback of encryption outside the storage device is the easy accessibility of the ciphertext. Connecting the hard disk to an unencrypting controller allows free access to the stored data. Inherent weaknesses result.

The most obvious weaknesses are evident with traffic-analysis attacks, which reveal the locality of changes when multiple snapshots were made on the disk. The attacker can copy back a disk sector's old content (unless

SECURITY

a system performs expensive large-set data authentication), allowing malicious manipulation of data, such as undoing banking transactions and online orders and unspending electronic cash.

Even just randomly changing certain disk blocks can have catastrophic consequences for the drive's unsuspecting owner. One example takes advantage of knowing the location of system files. An attacker who locates a jump address in the beginning of an encryption block can randomize it by changing the corresponding ciphertext block. There's a nonnegligible chance that this will alter the OS's behavior, such as opening some security backdoors for later attacks.

Furthermore, attackers can send maliciously crafted documents or programs to unsuspecting users. If users save those documents, attackers might be able to find their location (where disk blocks changed). Randomizing a certain block will then change the file in a predictable way, altering documents or program behavior.

HOST SOFTWARE ENCRYPTION

Encryption performed in host software has all the drawbacks of the external encryption modules and all the risks associated with an open environment. User errors, software bugs, and sloppy security policies could lead to the loss of secret keys or confidential data, and malware—rootkits, Trojans, viruses, and worms—could get into the system, compromising its security.

These weaknesses aren't present when developers do encryption in a closed environment with restricted I/O and unchangeable firmware.

DISCRYPTION ARCHITECTURE POSSIBILITIES

There are several options for discription architectures.

Threat model, attack scenarios

Even if spying hardware (such as a key logger or cable snooper) is attached to the host, secure authentication should be possible using challenge-response protocols with random

nonces. Since malicious host software can steal small amounts of data, users shouldn't enter any keys when the OS is running.

An attacker has at most one-time access to the ciphertext and can possibly read, modify, and copy other blocks, but not earlier content. Of course, attackers can carry out the usual general attacks, such as probes on data lines between electronic components and dictionary attacks on user passwords.

A master password can be used to reset drives that locked up after too many failed user-authentication attempts.

Encryption

Access control and encryption secure the data on disk drives. Available encryption modes include cipher-block chaining, various counter modes with location-dependent IV, location-tweaked electronic codebook (XTS of P1619), or wide-block encryption. Advanced Encryption Standard-128 or AES-256 are probably the best choices for the underlying cipher.

Key management

Developers can form data-encryption keys through secure combinations (cryptographic hash or encryption) of different entities: user key, platter signature, or a hidden root key in the electronics. The system randomly generates the user key, but it's not stored on the drive. For access control, the system also needs to store a secure hash of the user password. This way, the encryption key is of high entropy even if the user password is weak. After erasing the user-authentication information and key mixtures (for cryptographic disk erase or sanitation), a key search should deal with full key-length entropy.

Since the system might have several passwords, each with different asso-

ciated rights, the encryption key can't be derived from any of them. The drive can restrict the number of login attempts with invalid passwords, mitigating the negative effects of weak user passwords.

The system might need key export or import, where the key is wrapped in a secure envelope. This carries security risks, but it helps with data recovery after the device electronics fail, or authorities in certain markets might mandate it.

User authentication

To authenticate users, the system can ask them to prove their knowledge of a secret or possession of some device with secret information (such as a token, smart card, or fingerprint). The authentication process can be as simple as providing a valid password or as complex as a challenge-response protocol with random or sequential nonces (useful against message-replay or man-in-the-middle-type attacks).

The system can also support mutual authentication so users don't reveal their secrets to fake hard drives or the drive doesn't tell secrets to rogue hosts.

Administrators should limit the number of failed authentication attempts. For example, a drive could lock up after a user enters five wrong passwords, and only a higher-level authority could unlock it. This would hinder a search for weak user passwords.

The most basic authentication architecture features a user and master password. The latter can be used to reset drives that locked up after too many failed user-authentication attempts.

Access control

Without proper authentication, the disk drive shouldn't accept read/write commands, so an attacker won't see encrypted data or gain information about the locality of changes since a previous snapshot. Access control improves security, and export/import control authorities might require it. If the encrypted data was freely accessible, the user could use a secure disk as a stand-alone cryptographic coprocessor.

Disk drives can't provide absolute access control at a reasonable cost. Attackers can gain one-time access to the encrypted data when they remove the magnetic platters from a disk drive and place them on a multimillion-dollar spin stand. But when you employ tamper detection, legitimate users will detect tampering even when the disk drives are reassembled.

In this case, owners should no longer trust this data, so attackers can't make a second snapshot of new data, encrypted with the same keys, nor can they maliciously change the ciphertext and cause damage.

ADVANTAGES AND POSSIBLE FEATURES OF DISCRYPTION

Discription has many advantages over other encryption methods. It costs less to implement than external encryption modules and consumes less power than software encryption because of dedicated, optimized hardware.

With discription, encryption is transparently performed in full interface speed without any host processor load. Security is better than with host

software or controller-based encryption because of true random keys (no weak user keys) that are never stored in the drive. Disks drives are closed systems, making a malware infection impossible. In addition, developers can implement the security subsystem in a single chip, preventing debugger and bus-analyzer attacks.

The generation and storage of user keys in the drive provides further protection from malware. Hardware or hidden, protected ROM code performs security functions.

Discription is easy to set up, use, and deploy. Disk drives are fully operational and secure after setup in the factory because of internally generated secret keys and default passwords provided in tamperproof envelopes.

Erasing keys provides fast and secure disk sanitation. Partitions can use different keys to separate user partitions and multiple operating systems. Unmounted partitions remain safe from malware, user errors, and lunch-time attackers.

Discription can support multilevel and multifactor authentication as well

as third-party services and encrypted communication. It also allows for hierarchical key management, internal re-encryption, and forensic logging.

Encryption for protecting the confidentiality of stored data is the most secure internally in disk drives, as opposed to host software or external encryption modules. There are also speed, cost, and power consumption advantages; flexibility in applications; extra services; and simple and failsafe operation. A discussion about the "best" architecture would be valuable, leading the way to a discription standard. ■

Laszlo Hars is a researcher at Seagate Research. Contact him at Laszlo.Hars@seagate.com.

Editor: Jack Cole, US Army Research Laboratory's Information Assurance Center, jack.cole@ieee.org; <http://msstc.org/cole>

Practical Support for ISO 9001 Software Project Documentation


Using IEEE Software Engineering Standards



Susan K. Land
John W. Walz

978-0-471-76867-8 • October 2006
418 pages • Paperback • \$89.95
A Wiley-IEEE Computer Society Press

To Order:
1-877-762-2974 North America
+ 44 (0) 1243 779 777 Rest of World

Practical Support for ISO 9001 Software Project Documentation: Using IEEE Software Engineering Standards

 **WILEY**
Publishers Since 1807

 **IEEE**  **IEEE computer society**

www.wiley.com/ieeecs

ISO 9001 provides a tried and tested framework for taking a systematic approach to software engineering practices. Readers are provided with examples of over 55 common work products. This in-depth reference expedites the design and development of the documentation required in support of ISO 9001 quality activities. Also available:

- Practical Support for CMMI© - SW Software Project Documentation: Using IEEE Software Engineering Standards
- Jumpstart CMM©/CMMI© Software Process Improvements: Using IEEE Software Engineering Standards

15 % off for CS Members

THE PROFESSION

Continued from page 108

```
Initial_Topic( )
  MAKE Current_Directory with root topic name
  CALL Create_Topic ( Current_Directory )

Create_Topic( Current_Directory )
  DO
    Add a unique Subtopic
  LOOP UNTIL no more Subtopics

  IF no Subtopic exists THEN
    MAKE Directory
    "Current_Directory / Year / Month / Draft"
  ELSE
    FOR EACH Subtopic
      CALL Create_Topic
        ( Current_Directory / Subtopic )
    NEXT
  END IF
```

Figure 1. Pseudocode for a directory structure. The logic recursively creates subtopics within other subtopics under an initial topic.

```
Convention 1: D_Short Title_YYMMDDV_OWN.ext
Convention 2: D_Short Title_YYMMDDV_OWN_REV.ext
Convention 3: Full Title_YYMMDD.ext
```

Figure 2. Naming conventions. Document owners use Convention 1, reviewers use Convention 2, and both use Convention 3 to name final documents.

the collaboration group will provide the most concise naming scheme.

In some cases, the directory structure might exceed five or more levels. However, a directory tree that is too deep becomes more cumbersome to use. Deep directory structures should first be examined to validate their necessity.

Necessarily deep directory structures should be pruned to enhance usability. This involves taking a subtopic and its children to a higher level. Where the administrators place that subtopic depends on the logic the collaboration group’s expert members use to create the tree. It might make sense to extract a subtopic and its children several layers down and place it at the root topic level. This could require rewording the newly created topic so that users can easily associate it with the actual root topic.

Pruning reduces the tree’s depth but retains the structure’s logic so that the information is well organized. The overall strategy here involves logically

organizing information so that users can easily locate and identify it.

NAMING CONVENTIONS

Collaboration participants will likely fill the roles of document (or information) owners and reviewers during their tenure with the group. An owner is the principal author or topic owner, while reviewers are those who provide comments or approvals for draft and final documents.

Administrators implement the first two of the three naming conventions shown in Figure 2 to identify the roles used for creating draft documents: document owners use Convention 1, reviewers use Convention 2. They use Convention 3 to name final documents. No owner is associated with the final document because it represents the group’s collaborative efforts.

Each convention consists of unique fields. The date and version fields are concatenated; all others are delimited

with the underscore character for readability. Table 1 contains an explanation of the fields associated with each convention.

The following rules express the implementation for the conventions:

1. All documents within a directory use the same title-naming scheme.
2. In each case, the date selected is the working document’s last date.
3. Collaboration participants never alter other user documents or titles.
4. Comments or changes subsequent to comments are recorded in a copy of the former document.

Upon completion and acceptance of the last draft, the owner will move a copy of the latest draft to the parent of the Draft directory. This copy will be renamed using Convention 3 in the preceding list. The title chosen should closely match the document’s actual title. Keep in mind that shorter titles are easier to read.

PUTTING IT IN ACTION

The directory structure in Figure 3 shows how the framework and naming conventions might be used. The three collaboration members in this group, with their associated initials, are Alice (ALC), Bob (BOB), and Carol (CRL). ALC is the topic owner; BOB and CRL provide peer-review support.

As the following example shows, the collaboration group members can recreate an action timeline based on the naming conventions used:

- 1 July 2007: ALC creates a new draft document with the short title “Fireworks.”
- 1 July 2007: BOB and CRL provide comments to ALC in the form of new individual documents.
- 1 July 2007: ALC incorporates the reviewer comments, using the version letter “a” because it represents the latest draft by ALC on 1 July.
- 3 July 2007: ALC further updates and edits the work from 1 July. The update warrants a new document

that is considered the last draft prior to final acceptance.

- 4 July 2007: The last draft document in the list is copied to the parent of the Draft folder and renamed using Convention 3, indicating that this is the final approved version of the collaboration effort.

This approach’s elegance becomes clear once implemented. Most OS file systems display the directory and its files in alphabetical order. Using this naming convention guarantees that the latest draft document will be the last in the list. This makes it much easier for users not only to see the chronological order of the workflow but also to quickly find the latest draft document through a concise directory structure. Further, only allowing final documents to be kept in the same path as the associated Draft directory makes it easier to locate completed work.

ENHANCING SECURITY

In many collaborative situations not controlled with specialized tools, all users will likely have equal rights to the work in the mutually shared directories. This strategy makes it relatively simple for users to read and modify work done by the group. However, this situation proves problematic when malicious logic, such as a virus, begins modifying or destroying the group’s work. This threat can be countered with the proper use of the OS file system *access control list* (ACL).

The naming conventions presented provide an opportunity to leverage the OS access control mechanism to enhance security. We assume that the target system uses *discretionary access control* (DAC) for object management. Whenever users create a new document or a copy of an existing one, they become that new object’s owner.

Setting the ACL for the document so that only the owner retains full control and all others have read-only access provides an elevated assurance of the integrity of information used in the collaboration. This inherent DAC mechanism can prevent acci-

Table 1. Document-naming-convention fields.	
Field	Explanation
D	Represents a draft document
Short title	A very short title of the document; relevant acronyms preferred
Full title	A descriptive title of the document
YY	Last two digits of the current year
MM	Two digits for the current month
DD	Two digits for the current day
V	This is a version character starting with the letter “a” and incremented for succeeding versions of an original within the same day. This aspect of the convention is used only when multiple versions of a document are created on a given day.
OWN	Initials of the topic owner or manager
REV	Initials of the document reviewer
.ext	Appropriate document extension

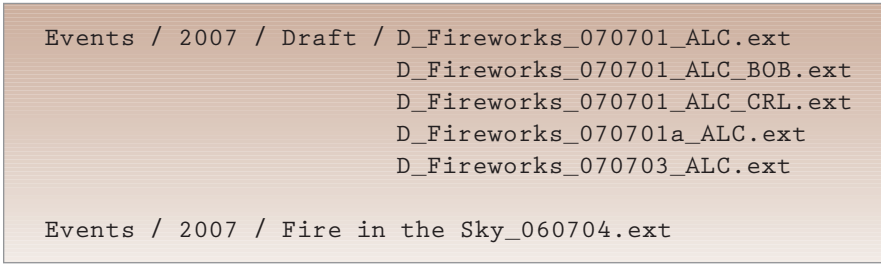


Figure 3. Directory structure. The three collaboration group members are Alice (ALC), Bob (BOB), and Carol (CRL).

dental or malicious insiders from modifying or deleting files they do not own. Collaborators and system administrators must pay special attention to policies in the host system that might attempt to propagate ACL entries from the directory to the new object. This might result in a less than adequate ACL implementation.

A read-only ACL entry for nonowners provides protection against malicious logic that might try to subvert the system. For example, a virus might be able to delete or encrypt files. If those involved follow the proposed ACL strategy, only the files the infected user owns are at risk. Proper use of ACL mitigates the effect of malicious code on the other documents in the group because the user is limited to read-only access.

Using the inherent capabilities of an OS file system can meet some common challenges that collaboration groups encounter.

Designing and implementing coherent file and directory structures will make it easier for collaborating members to locate relevant files as well as the latest version. Leveraging file system access-control capabilities will prevent unauthorized modifications to work the group’s members have developed. A directory-structure approach not only makes it easier for group members to find relevant information but also helps protect their work’s integrity.

Sean M. Price is an independent security consultant. Contact him at sean.price@sentinel-consulting.com.

Editor: Neville Holmes, School of Computing, University of Tasmania; neville.holmes@utas.edu.au. Links to further material are at www.comp.utas.edu.au/users/nholmes/prfsn.

Supporting Resource-Constrained Collaboration Environments

Sean M. Price, Sentinel Consulting



Using an approach that incorporates a directory structure helps collaborative group members find information while protecting their work's integrity.

The many collaboration and document-management products available today help establish controls over document workflows. Essentially, these applications automate information categorization, filing, version control, and historical tracking.

The advent of products that help find the necessary but long-forgotten documents needed to complete a task has spurred demand for these applications. Organizations with document collaboration requirements clearly benefit from this technology. Computing professionals, in turn, play an important role in identifying the most appropriate technology for the collaboration problem at hand.

Unfortunately, resources are not always available to acquire the needed product. Document-management and collaboration tools can be cost-prohibitive to many organizations. This does not bode well for resource-constrained organizations that need some form of information management. However, computing professionals can

propose nonautomated solutions to accommodate manual problems, such as file-naming conventions and document versioning, through the appropriate use of the technology at hand. Although automated tools are more attractive and exciting, making such acquisitions is not always feasible.

Organizations without processes and procedures in place for information management can end up with their documents scattered in a directory kludge. A lack of structure for information management can make it difficult to find relevant information. People do not always categorize their information efficiently, which impedes future rediscovery. This is evident by the inclusion of file system tools in modern operating systems that let users do keyword searches for documents within their control.

Computing professionals can bring order to file and directory chaos by proposing logical yet nontechnical solutions that meet the needs of a collaboration group. A well-thought-out directory structure can also afford

security benefits that might have been missing in the prior information conglomeration.

COLLABORATION DIRECTORY ALGORITHM

The collaborators can create the directory structure by recursively asking a few questions, with the idea of creating a structure of topics within topics. The highest point in the structure represents the most general or abstract view of a particular subject, while the bottom topic offers the most granular view. A topic within a topic is a *subtopic*. The pseudocode in Figure 1 shows how to create the directory structure.

The pseudocode logic is straightforward. First, create an initial topic. Next, identify any subtopics that might exist. Move to the first subtopic, if it exists, and ascertain more subtopics. Repeat this process until the current topic returns no immediate subtopics. In this case, create a directory within the current topic that includes the path/Year/Month/Draft, return to the immediate parent level, and move to the next subtopic if it exists.

The approach for creating the output, which appears in pseudocode, exemplifies the idea that a logical process can help solve a manual problem. It relies on properly identifying the subtopics, which in turn requires involving the collaboration of group members with strong analytical abilities. With their combined expertise, the collaborators can decompose the workflow's major components into a hierarchy structure. Primarily, they will need to correctly determine if a given subtopic best fits under the immediate parent topic. Computing professionals can facilitate this process—manually or through organization policy—to achieve the desired results.

DEFINITIONS AND DIRECTORY TYPES

Names for topic and subtopic directories should be kept as short as possible. This makes it easier for users to quickly navigate or explore deep directory trees. Acronyms well-known to

Continued on page 106

one hundred fifty years of content

fifteen leaders in science & technology research

three million documents

one gateway to it all

scitopia.org

Search scitopia.org to find quality content from leaders in science and technology research. Scitopia.org generates relevant and focused results – with no Internet noise. From peer-reviewed journal articles and technical conference papers to patents and more, scitopia.org is a researchers' heaven on earth.

search

scitopia.org

Integrating Trusted Science + Technology Research

Scitopia.org was founded by: Acoustical Society of America • American Geophysical Union • American Institute of Physics • American Physical Society • American Society of Civil Engineers • American Society of Mechanical Engineers • American Vacuum Society • ECS • IEEE • Institute of Aeronautics and Astronautics • Institute of Physics Publishing • Optical Society of America • Society of Automotive Engineers • Society for Industrial and Applied Mathematics • SPIE

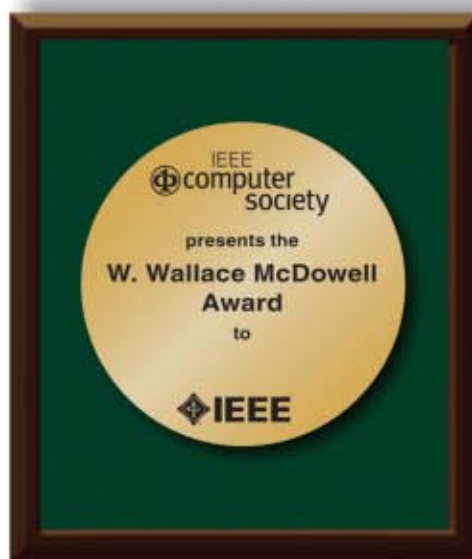
Nominations are solicited for the HARRY H. GOODE MEMORIAL AWARD and the W. WALLACE McDOWELL AWARD

Harry H. Goode Memorial Award



The Harry H. Goode Memorial Award was initiated by the American Federation of Information Processing Societies (AFIPS) in 1964 in recognition and appreciation of Mr. Goode's invaluable contributions to the information processing societies. Upon the dissolution of AFIPS in 1990, the IEEE Computer Society agreed to assume responsibility for the continuation of the award.

The award, which consists of a bronze medal and \$2,000 honorarium, is presented annually on the basis of achievements in the information processing field which are considered either a single contribution of theory, design, or technique of outstanding significance, or the accumulation of important contributions on theory or practice over an extended time period, the total of which represent an outstanding contribution.



W. Wallace McDowell Award

The W. Wallace McDowell Award was established through a grant by the International Business Machines Corporation in honor of its late vice president, W. Wallace McDowell, for his contributions to computer development and engineering.

The award, which consists of a bronze medal and \$2,000 honorarium, is presented annually by the IEEE Computer Society to individuals whose professional work has been outstanding in recent theoretical, design, educational, practical, or other similar innovative contribution that falls within the scope of IEEE Computer Society interest. The award may be given for a single contribution of great merit or a series of lesser contributions that have had or are expected to have an important influence on the computer field.



Nomination form and submission:
<http://www.computer.org/awards>

Deadline for both awards:
15 September 2007