Innovative Technology for Computer Professionals

Computer

August 2004

sensor networks

IEEE

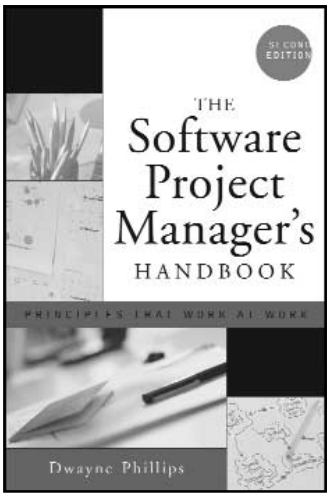
IEEE COMPUTER SOCIETY

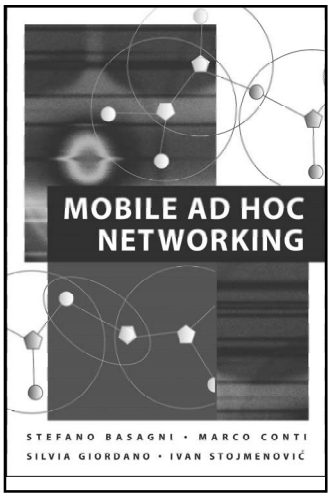
http://www.computer.org

Wiley and The IEEE Computer Society

Partners in Publishing

WILEY THE COMPUTER SOCIETY IEEE





The Software Project Manager's Handbook **Principles that Work at Work, 2nd edition**

Dwayne Phillips

This book guides both software project managers and their team members toward a common goal, emphasizing basic principles that are extremely effective at work. The author's unique approach stresses that success on software projects has more to do with how people think individually and in groups than with programming.

A Wiley-IEEE Computer Society Press Publication

ISBN: 0-471-67420-6 **Price: $59.95**
Pages: 504 (Paper) **Date: July 2004**

Mobile Ad Hoc Networking

Stefano Basagni, Marco Conti, Silvia Giordano, Ivan Stojmenovic´

Assimilating the most up-to-date information on research and development activities in this rapidly growing area, it covers physical, data link, network, and transport layers, as well as application, security, simulation, and power management issues in sensor, local area, personal, and mobile ad hoc networks.

A Wiley-IEEE Press Publication

ISBN: 0-471-37313-3 **Price: $ 79.95**
Pages: 480 **Date: Aug. 2004**

Great Software Debates

COMING SOON!

Alan M. Davis

The book is a collection of essays presenting proven ideas on how to approach the difficult task of constructing software. Software practitioners who are in desperate need of sound, palatable advice will find this book a valuable reference.

A Wiley-IEEE Computer Society Press Publication

ISBN: 0-471-67523-7 **Price: $ 45.00**
Pages: 288 **Date: November 2004**

Stuff You Don't Learn in Engineering School

COMING SOON!

Skills for Success in the 'Real World'
Carl Selinger

It gives the young engineer an easy-to-read "cribsheet" so they can comfortably face the complex professional world of dealing with people, processes, and yes, even meetings! The book covers non-technical "real world" leadership skills like decision-making, setting priorities, negotiating, teamwork, running meetings, and better writing and speaking.

A Wiley-IEEE Press Publication

ISBN: 0-471-65576-7 **Price: $ 39.95**
Pages: 192 **Date: Nov. 2004**

Setting the Standard

***John Wiley & Sons Inc.*, and the *IEEE Computer Society* are proud to present a new series of books based on IEEE Standard 12207 and other Software Standards consistent with Software Engineering Body of Knowledge (SWEBOK).**

Now Available:

Software Reuse: A Standards-Based Guide
Carma McClure
0-7695-0874-X 318 (Paper) 2001 $60.00

Coming Soon:

Roadmap to Software Engineering: A Standards-Based Guide
James Moore

Software Requirements: A Standards-Based Guide
Linda Shafer

Software Configuration Management: A Standards-Based Guide
Donald F. Shafer

www.wiley.com/ieeecs

1. CALL: NORTH AMERICA: 1-877-762-2974
ALL OTHERS: +44 (0) 1243 779 777

2.FAX: U.S.: 1-800-597-3299
ALL OTHERS: +44 (0) 1243 843 296

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

4. E-MAIL: U.S.: custserv@wiley.com
ALL OTHERS: cs-books@wiley.co.uk

ASPLOS XI Call for Participation

Park Plaza, Boston, Massachusetts, October 9 through 13, 2004

Eleventh International Conference on

Architectural Support for Programming Languages and Operating Systems

Saturday, Oct. 9th

Workshops

WASSA: Workshop on Architectural Support for Security and Anti-Virus

Operating System and Architectural Support for the on demand IT InfraStructure

Tutorials

Architectures, Languages, and Compilers for the Streaming Domain

Using the Click Modular Router for Routing, Measurement, Protocols, and the Classroom

Designing Instrumentation Tools With PIN

Building Experimental Networks Using Platform FPGA's

Sunday, Oct. 10th

Workshops

Second Value-Prediction and Value-Based Optimization Workshop

Workshop on Building Block Engines for Computers and Networks

Tutorials

Using The Liberty Simulation Environment withEmphasis on Hardware Validated OS-Level Simulation

Tutorial on Remote Direct Memory Access (RDMA) over IP Transports

Program locality models and Their Use in Memory Optimization

Computing In The Presence of Soft Errors

Reception, 6:30pm Park Plaza

Monday, Oct. 11th

Keynote Speech: *Building Dependable Software*
Jim R. Larus, Microsoft Research

Session 1: New Models and Architectures

Programming with Transactional Coherence and Consistency (TCC)
Lance Hammond, Brian Carlstrom, Vicky Wong, Ben Hertzberg, Mike Chen, Christos Kozyrakis, and Kunle Olukotun, Stanford University

Spatial Computation
Mihai Budiu, Girish Venkataramani, Tiberiu Chelcea, and Seth Copen Goldstein, Carnegie Mellon University

An Ultra-Low-Power Processor for Sensor Networks
Virantha Ekanayake, Clinton Kelly IV, and Rajit Manohar, Cornell University

Session 2: Storage

D-SPTF: Decentralized Request Distribution in Brick-Based Storage Systems
Christopher R. Lumb, Carnegie Mellon Univ., Richard Golding, IBM Almaden Research, and Gregory R. Ganger, Carnegie Mellon Univ.

XYZ: Building Reliable Enterprise Storage Systems on the Cheap
Yasushi Saito, Svend Frolund, Alistair Veitch, Arif Merchant, Susan Spence, Hewlett-Packard Labs

Deconstructing RAID Storage Arrays
Timothy E. Denehy, John Bent, Florentina I. Popovici, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau, Univ. of Wisconsin-Madison

Session 3: Security

HIDE: An Infrastructure for Efficiently Protecting Information Leakage on the Address Bus
Xiaotong Zhuang, Tao Zhang, and Santosh Pande, Georgia Institute of Technology

Secure Program Execution via Dynamic Information Flow Tracking
G. Edward Suh, Jae W. Lee, and Srinivas Devadas, MIT

Tuesday, Oct. 12th

Session 1: Architecture

Coherence Decoupling: Making Use of Incoherence
Jichuan Chang, Univ. of Wisconsin-Madison, Jaehyuk Huh, Doug Burger, Univ. of Texas at Austin, and Gurindar S. Sohi, Univ. of Wisconsin-Madison

Continual Flow Pipelines
Srikanth T Srinivasan, Ravi Rajwar, Haitham Akkary, Amit Gandhi, and Mike Upton, Intel

Scalable Selective Re-execution for Speculative Dataflow Architectures
Rajagopalan Desikan, Simha Sethumadhavan, Doug Burger, and Stephen W. Keckler, University of Texas at Austin

Session 2: Potpourri

HOIST: A System for Automatically Deriving Static Analyzers for Embedded Systems
John Regehr and Alastair Reid, University of Utah

Helper Threads via Virtual Multithreading On An Experimental Itanium 2 Machine
Perry H. Wang, Jamison D. Collins, Hong Wang, Dongkeun Kim, Bill Greene, Kai-Ming Chan, Aamir B. Yunus, Terry Sych, and John P. Shen, Intel

Low-Overhead Memory Leak Detection Using Adaptive Statistical Profiling
Trishul Chilimbi, Microsoft Research, Matthias Hauswirth, University of Colorado, Boulder

Session 3: Memory System Analysis and Optimization

Locality Phase Prediction
Xipeng Shen, Yutao Zhong, and Chen Ding, University of Rochester

Dynamic Tracking of Page Miss Ratio Curve for Memory Management
Pin Zhou, Vivek Pandey, Jagadeesan Sundaresan, University of Illinois, Urbana-Champaign, Anand Raghuraman, Google, Yuanyuan Zhou, University of Illinois, Urbana-Champaign, and Sanjeev Kumar, Intel

Compiler Orchestrated Prefetching via Speculation and Predication
Rodric M. Rabbah, MIT, Hariharan Sandanagobalane, NUS, Mongkol Ekpanyapong, Georgia Institute of Technology, and Weng-Fai Wong, NUS

Software Prefetching for Mark-Sweep Garbage Collection: Hardware Analysis and Software Redesign
Chen-Yong Cher, Antony L Hosking, and T N Vijaykumar, Purdue University

Wild/Crazy Ideas Session

Wednesday, Oct. 13th

Session 1: Reliability

Devirtualizable Virtual Machines Enabling General, Single-Node, Online Maintenance
David E. Lowell, Yasushi Saito, and Eileen J. Samberg, Hewlett-Packard

Architectural Fingerprinting: Bounding Soft Error Detection Latency and Bandwidth
Jared C. Smolens, Brian T. Gold, Jangwoo Kim, Andreas Nowatzyk, Babak Falsafi, and James C. Hoe, Carnegie Mellon University

Application-level Checkpointing for Shared Memory Programs
Greg Bronevetsky, Martin Schulz, Peter Szwed, S. Shafat Zaman, and Keshav Pingali, Cornell University

Session 2: Power

Formal Online Methods for Voltage/Frequency Control in Multiple Clock Domain Microprocessors
Qiang Wu, Philo Juang, Margaret Martonosi, and Douglas W. Clark, Princeton University

Leveraging SMT and CMP to Manage Power Density Through the Operating System
Mohamed Gomaa, Michael D. Powell, and T. N. Vijaykumar, Purdue University

Performance Directed Energy Management for Main Memory and Disk
Xiaodong Li, Zhenmin Li, Francis David, Pin Zhou, Yuanyuan Zhou, Sarita Adve, University of Illinois at Urbana-Champaign, and Sanjeev Kumar, Intel

Please check the following web site for up to date information:

http://www.eecg.toronto.edu/asplos2004

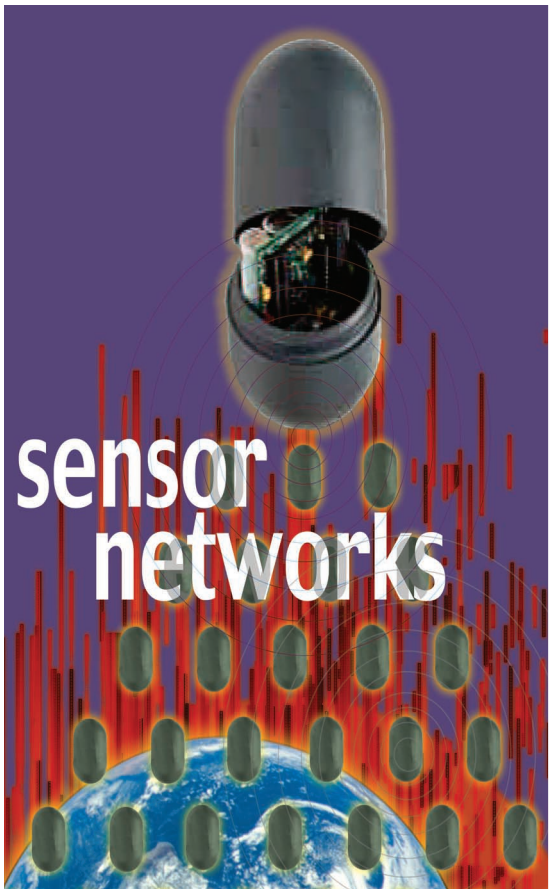
ASPLOS XI is sponsored by

SIGARCH
SIGPLAN
SIGOPS

Corporate Support

Innovative Technology for Computer Professionals
Computer

August 2004, Volume 37, Number 8



Cover design and artwork by Dirk Hagner

A shift to mass-produced intelligent sensors and the use of pervasive networking technology give wireless sensor networks a new kind of scope that can be applied to a wide range of uses. WSNs merge information technology that spans hardware, systems software, networking, and programming methodologies. In this issue, we look at three projects that deploy WSNs to monitor the natural environment, a platform for implementing wireless sensor networks that achieve low-power operation, and a program that integrates this emerging field into an undergraduate computer engineering curriculum.

COMPUTING PRACTICES

COVER FEATURES

GUEST EDITORS' INTRODUCTION

Computer

IEEE Computer Society: http://www.computer.org
Computer: http://www.computer.org/computer
computer@computer.org
IEEE Computer Society Publications Office: +1 714 821 8380

Membership Magazine of the

IEEE COMPUTER SOCIETY

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 $17 for a subscription to *Computer* magazine. Nonmember subscription rate available upon request. Single-copy prices: members $20.00; nonmembers $88.00.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 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 4960-2 Walker Road, Windsor, ON N9A 6J3. Printed in USA.

ARTICLE SUMMARIES

Modeling Complex Spoken Dialog

pp. 32-40

Hans Dybkjær and Laila Dybkjær

Commercial spoken dialog systems, which handle increasingly advanced tasks, must deal with three development challenges: how to model the dialog when the user can select from many tasks, how to communicate with customers about SDS design, and how to develop code for a large SDS when the design will likely require updates throughout development.

To develop a commercial phone-based SDS that supplies information to Danish employees about their holiday allowance, the authors had to meet all these challenges. Key to this system was their use of the Conceptual Dialog Language, which expresses patterns specific to the dialog model while still providing a clear picture of it to domain experts and supporting model updates.

CDL captures the specifics of an SDS application while remaining flexible enough to support a range of modeling styles and requirements. To facilitate effective communication with domain experts, modelers can translate the dialog model in CDL to an HTML document, or they can compile it to a programming language for executable code.

Sensor Network Applications

pp. 50-61

Kirk Martinez, Jane K. Hart, and Royan Ong

Sean M. Brennan, Angela M. Mielke, David C. Torney, and Arthur B. Maccabe

Miklos Maroti, Gyula Simon, Akos Ledeczi, and Janos Sztipanovits

Sensor networks can be used to monitor our environment, objects in that environment, and the interactions of objects with each other and their encompassing environment. Examples include environmental and habitat monitoring, structural monitoring and condition-based equipment maintenance, and disaster management and emergency response. Researchers working on three diverse projects have developed other novel applications for sensor network technology.

In "Environmental Sensor Networks," Kirk Martinez and his coauthors from the University of Southampton describe their GlacsWeb project, which involves ongoing research in subglacial bed deformation. They also discuss the challenges encountered in extracting data gathered by sensor nodes deployed in remote locations.

In "Radiation Detection with Distributed Sensor Networks," Sean M. Brennan and coauthors discuss a project being developed at Los Alamos National Laboratory in cooperation with the University of New Mexico to provide a distributed sensor network for detecting vehicles transporting radioactive isotopes that could potentially be detonated over a densely populated area.

Finally, in "Shooter Localization in Urban Terrain," Akos Ledeczi and coauthors describe PinPtr, a prototype system that provides a novel approach for detecting and locating a sniper in a challenging environment such as complex urban terrain.

These three projects demonstrate the promise of sensor network technology for monitoring our environment and our safety in that environment.

WiseNET: An Ultralow-Power Wireless Sensor Network Solution

pp. 62-70

Christian C. Enz, Amre El-Hoiydi, Jean-Dominique Decotignie, and Vincent Peiris

Since wireless sensor networks often are deployed in regions that are difficult to access, the nodes should not require maintenance. They must be energetically autonomous, using batteries that do not need to be replaced or recharged. In many application scenarios, the targeted node lifetime typically ranges from two to five years, imposing drastic constraints on power consumption. With a single 1.5-V AA alkaline battery, the average power consumption ranges from 100 to 10 microwatts, for a node lifetime ranging between two and seven years.

To conserve power, the nodes must sleep most of the time. Thus, the Swiss Center for Electronics and Microtechnology developed WiseNET to optimize power consumption. An ultralow-power platform for the implementation of wireless sensor networks, WiseNET achieves low-power operation through a careful codesign approach. The system combines a complex system-on-chip sensor node with a dedicated duty-cycled radio and WiseMAC, a low-power MAC protocol designed for low-duty-cycle wireless sensor networks. The WiseNET solution consumes about 100 times less power than comparable alternatives available today.

The Flock: Mote Sensors Sing in Undergraduate Curriculum

pp. 72-78

Bruce Hemingway, Waylon Brunette, Tom Anderl, and Gaetano Borriello

Educational excellence requires exposing students to the current edge of research. To ensure that student projects follow the same trajectory that industry is traveling, educators must continually introduce emerging techniques, practices, and applications into the curriculum.

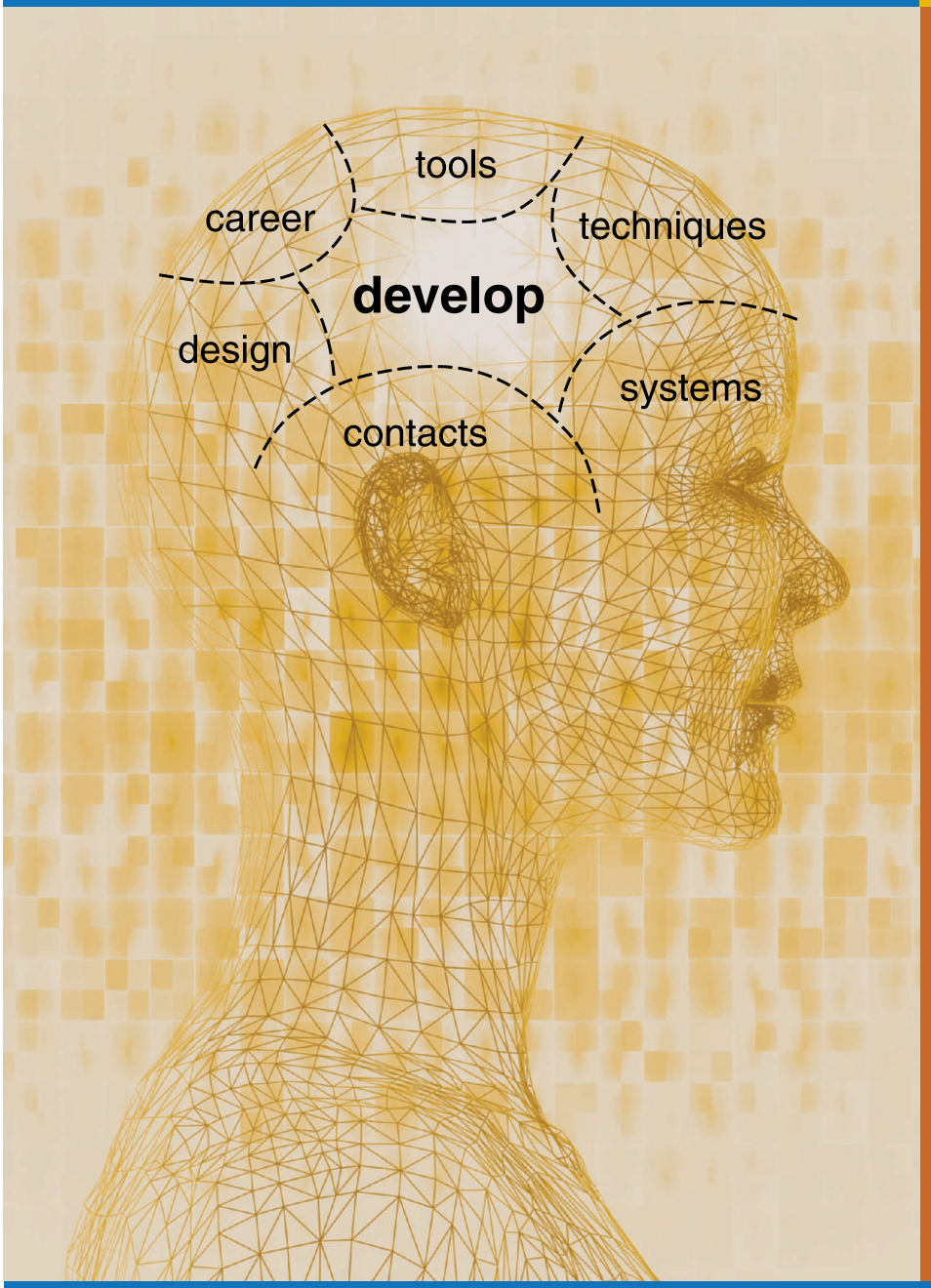
While many graduate-level classes focus on sensor networks, they do not serve as a good template for the undergraduate curriculum because they assume a much greater breadth of knowledge on the students' part as well as greater maturity to absorb new topics on their own.

The "Flock of Birds" project in the Department of Computer Science and Engineering at the University of Washington integrates the theory and practice of wireless sensor networks into the mainstream curriculum early enough to form a basis for all students' understanding of embedded computing—not just a short-lived application exercise for some of their capstone design projects.

Embedded Systems Conference Boston

Co-located with: Embedded Security seminar

September 13-16, 2004
Hynes Convention Center
Boston, MA



Develop Your Career

Attend the East Coast's largest embedded systems design educational opportunity

The Embedded Systems Conference Boston

Keep current with 78 classes and full-day tutorials taught by more than 40 industry experts addressing your toughest processor-based design challenges. The event features a special focus on Single Board Computers including classes, a panel, and a Single Board Computer Village on the exhibits floor.

The Embedded Security Seminar

Learn to apply security to your connected systems with this three-day technical program offering sessions on topics including encryption, standards, firewalls, protecting IP, and security for wireless and Internet protocol.

Maximize Your Knowledge. Attend both conferences in the same week and save!
Design your own Combo Pass conference package choosing from 92 technical sessions from the **Embedded Systems Conference Boston** and the **Embedded Security Seminar.**

Save up to $595, register by August 17th.

www.embedded.com/esc

Priority code: XMAT

Conference Sponsors:

EETIMES EmbeddedSystems PROGRAMMING

CMP United Business Media

COMPUTER VULNERABILITY

Although I enjoyed Butler Lampson's article in *Computer*'s June 2004 issue ("Computer Security in the Real World," pp. 37-46), I am concerned that part of the article might be misread, perpetuating a viewpoint that leaves our computers vulnerable to viruses.

Figure 1 in this article shows the guard authenticating the principal and making an access decision based on that authentication. Further, the authentication and authorization steps are described using the pronoun "who."

Lampson clearly states that the principal need not be a person, but readers might miss the significance of this point. Some could conclude that Lampson is saying that we can base access decisions on the identity of the person who started the program. Virus attacks show the folly of this approach. Access control based on a user's identity can keep people from doing what they are not allowed to do. Unfortunately, it does nothing to stop a process acting on a user's behalf from doing something the user is allowed to do but doesn't want done. Viruses exploit this flaw in the access control model of today's operating systems.

The solution is to enforce the Principle of Least Privilege at a granularity finer than that of the user without making a system that is secure but hard to use. The research prototype we have built at HP Labs for Microsoft Windows shows this can be done. Manipulating privileges based on the user actions that designate files and programs enhances security without adversely affecting usability.

Alan Karp
Palo Alto, Calif.
alan.karp@hp.com

The author responds:

Karp is quite right that it is unwise to base access control entirely on the identity of the user who starts a program. I discuss this point at length on p. 44 in the article, but I should have emphasized that treating a program as a principal is the most effective way to control viruses.

In "Computer Security in the Real World," Butler Lampson offers two suggestions: "… the system should classify all programs as trusted or untrusted based on how they are signed" and "… a trusted authority must sign all executable programs." It is important to caution readers against this kind of black-and-white thinking.

No entity can ever be classified as "trusted" in an absolute sense. The word "trusted" has no meaning without an accompanying context that answers two questions: "Trusted by whom?" and "Trusted to do what?"

Software signatures do not guarantee correct behavior. Moreover, "correct behavior" is not even a useful concept without an understanding of the software user's expectations. For example, spyware behaves correctly—just not in the user's interest. A signature is only useful if the user understands the particular assurance the signer intended to make and trusts the signer to make that assertion.

Requiring signatures on all programs also leads in a dangerous direction: It needlessly increases the barriers to acceptance of software from smaller companies or open source software teams.

We must acknowledge that users cannot be expected to determine the correctness of their software and place absolute trust in it. Instead, we should provide the means to run software while limiting its ability to do harm.

I encourage all readers to start asking questions whenever they see the words "trust" or "trusted" used without qualification.

Ka-Ping Yee
University of California, Berkeley
ping@zesty.ca

POWERPOINT COUNTERPOINT

Neville Holmes is indeed correct that negative stories about PowerPoint are "not a joke" ("In Defense of PowerPoint," The Profession, July 2004, pp. 100, 98-99). He is also correct in stating that the ultimate responsibility for the quality of a presentation lies with the presenter. However, he is dead wrong in his implication that PowerPoint is not a deeply flawed tool that encourages bad presentations and discourages good ones.

By deploying zillions of defaults and automated "corrections" that are hard to disengage, PowerPoint's producers assume significant responsibility for the stylistic wasteland all too familiar to audiences around the world. Metaphorically, I believe the responsibility of PowerPoint's producers would fall under the legal heading "attractive nuisance."

Articles by and about Edward Tufte, author of the *Wired* article, "PowerPoint Is Evil" (www.wired.com/wired/archive/11.09/ppt2.html), provide information about the negative aspects of PowerPoint. Oddly, Holmes never names Tufte. Just Google "tufte powerpoint," and you'll find more than you care to read.

A balanced starting point for the interested reader is the *Wikipedia* entry on PowerPoint: http://en.wikipedia.org/wiki/PowerPoint.

Richard (DJ) L. Waddell Jr.
Laurel, Md.
richard.waddell@jhuapl.edu

Neville Holmes responds:

I had not intended to imply that I thought PowerPoint a good thing. Indeed, I imagined that stating on the opening page that I found it very difficult to make PowerPoint do what I

wanted and offering lengthy suggestions about when computing professionals should not use PowerPoint would have made this plain.

My main points, emphasized in the conclusion, were that computing professionals should be trained in presentation techniques and that they should resist professional and public attempts to blame digital technology for the ills consequent on its misuse.

By this, I do not mean to imply that I disagree with the points Richard Waddell makes about PowerPoint's qualities and effects. However, my strongly held view is that PowerPoint should not be the main target of criticism. Rather, the criticism should be directed at the educators and other professionals who neglect their responsibilities in favor of letting technology do the driving.

On Edward Rolf Tufte, I suggest that readers go straight to http://artenumerica.com/inspiration/tufte.en.html for an introduction to Tufte and links to his more appropriate writings. Computing professionals should ignore the chip Tufte has on his shoulder about PowerPoint, and they should take his splendid work on data presentation to heart.

In his essay about PowerPoint, Neville Holmes offers the shopworn argument that technology is benign, and it is only the use of technology that is for good or ill. The same argument is also used for handguns, and, in both that use and in the defense of PowerPoint, it remains unconvincing.

It might be useful, instead, to consider the statement (by Edsger Djikstra, I believe) that the tools we use affect not only the way we think, but our ability to think. Since PowerPoint is primarily a management tool, what is its effect on the way that managers receive, present, and act upon information? At what point does the need to put the argument in a set of bullet points and box-and-line diagrams suppress consideration of details that all too often are key to project success of failure? How does PowerPoint affect a manager's ability to think about detail?

John Boddie
Landenberg, Pa.
jboddie@verizon.net

Neville Holmes responds:

To use a rather shopworn cliché, John Boddie is putting words into my mouth. My point was not that technology is benign, but that it is neutral—neither benign nor malignant.

Technology—tools and techniques—is something we use, professionally or domestically. If our use of technology results in good or bad, then the credit or blame belongs to us. Although society might perhaps be considered indirectly responsible, the blame for running amok with a handgun belongs directly to the person who used the handgun in that way.

When we use technology professionally, either as computing professionals or as managers, it is our professional responsibility to use it properly. If we don't know how to do that, it is our direct fault for using the technology at all. If it is important technology, failing to use it properly is indirectly the fault of the educators in our professions for not having equipped us with the appropriate skills.

Certainly our tools and techniques can influence the way we solve problems, but we are being unprofessional if we analyze problems to suit the technology we have or if we design our solutions primarily to suit the technology we plan to use.

As professionals, our focus must be on the people whose problems we are tackling, not on our own convenience. If we allow our tools to dictate how we think, then more fools we, domestically or professionally.

I fully agree with Neville Holmes's comments about PowerPoint.

Every user of this useful tool is responsible for the developed slides and the manner in which they are presented. Frequent poor use of PowerPoint doesn't mean that it can't be used intelligently. I recommend using a combination of PowerPoint and Microsoft Equation 3.0 for presentations that include mathematical symbols and equations.

Janusz Kowalik
Seattle, Wash.
j.kowalik@comcast.net

UNIVAC OVERDRIVE MODIFICATION

I am trying to locate information about a hardware modification called "Overdrive" that was designed for the Univac I. I have (with the bemused permission of Unisys) written a simulator for Univac I and Univac II, and I would like to incorporate "Overdrive" in it as an option if I can find authenticated details of how it worked.

I am not particularly interested in the mechanics of the hardware modification itself, but rather in the changes it made to a Univac I from a programmer's point of view. I believe the modification allowed for three instructions per Univac word, rather than the "normal" two, but I have no idea what it did for instructions that required two characters (most were one character) and no address or for those that required two characters and an address.

I would be delighted to hear from any readers who have any information—or are even merely curious about—Univac I.

Peter Zilahy Ingerman
Willingboro, N.J.
pzi@ingerman.org

We welcome your letters. Send them to computer@computer.org. Letters are subject to editing for style, clarity, and length.

The Art of the Possible

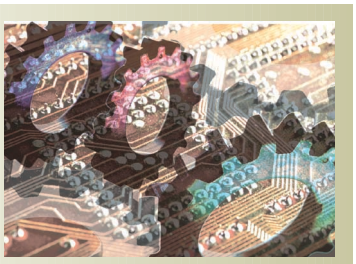
Bob Colwell

Engineering is the art and science of compromise, striving for some desired outcome while giving up as little as necessary—and not more than you can afford—on other aspects of the design. This process defines the "art of the possible." Occasionally, a design team will create a product that is so much better than anything else that it wins on all of its important metrics. When that happens, the company will usually recognize the product's superiority by raising the price so much that they reopen the door for their competitors, thus simultaneously restoring the cosmic balance and driving their own designers crazy.

But that situation is rare. By far the most common case is that the design team at Company A juggles 10 major aspects of its conceived product and arrives at some set of tradeoffs that nobody is completely happy with: Management thinks it costs too much; marketing is afraid the competition will come up with something better and they'll have to save the day with an innovative advertising campaign; and the designers wish they had more time, money, and engineers, without which they know they must back down from certain targets that would otherwise be technically achievable.

All the while, Company B's design team is doing the same thing, starting with essentially the same technology and aiming for the same market, but coming up with a different set of tradeoffs.

The one thing these two teams have in common is that whatever they design will embody the compromises they thought best—assuming neither has succumbed to the ever-present danger of design-by-committee, in which case the pathetic, misbegotten horror they produce has a good chance of representing the worst of every possible choice available.



Often, a product prevails because it embodies the best available set of compromises.

If both companies do their jobs well, the marketplace now has a choice between two solutions to some need, and—all else being equal, which sometimes happens—the product the market likes best will ultimately prevail. This will occur not because that product embodies *fewer* compromises, but because it embodies the *best available* set of compromises.

CONSTRAINTS ARE GOOD

So far, I've referred to compromises as though they're the unfortunate by-product of an imperfect world: If only we could get rid of the necessity to make tradeoffs, we'd all be so much better off. I don't actually believe that, however. In a subtle way, I believe real-world constraints actually benefit the design process. They help set boundaries that we unconsciously use to rein in our thought processes and keep them from chaotically running open-loop.

A few engineers—and artists and musicians, for that matter—will react usefully when someone puts a blank sheet of paper in front of them and tells them to create something. But, unlike Leonardo, Kelly Johnson, or Seymour Cray, when most engineers face this situation, we must first fight off a feeling of panic as we feel our mental machinery over-revving and spinning apart in all directions.

Constraints are good. They focus our attention on areas where real innovation can occur. They steer us away from a breadth-only subconscious search of the possibility space, which is open-ended and may never converge, and toward a deep consideration of ideas that are either promising or that the constraints themselves necessitate.

Better yet, while we can't enumerate the breadth-search prospects and have difficulty grasping how much of that space we may already have searched mentally, we can directly list the areas that our constraints steer us toward, and we can employ a thorough, logical, guaranteed-to-complete-sometime method for checking them.

I sometimes wonder if one reason that software and hardware products exhibit such vastly different design errata rates is that the hardware universe is constrained by nature itself, while the software universe is constructed by humans who believe that removing as many constraints as possible is the right thing to do. The resulting "possibility spaces" are several orders of magnitude apart in size.

ONE HUNDRED MILLION CLOSEST FRIENDS

Then we come to the end user. This is the very important person who exchanges her hard-earned money for your product. She's the reason we design things in the first place. In the end, hers is the only vote that counts as to whether a design is a success or failure. Her purchase of your product and not a competitor's is at least comforting circumstantial evidence that she preferred your combination of compromises over the alternatives.

Any given buyer can purchase a product for reasons quite unrelated to anything a design engineer can directly influence, such as store location, ad campaigns, word of mouth, pet rock fads, and so on, but here we're considering a representative buyer and her hundred million closest friends.

There was a time when end users were technically sophisticated persons who understood that nothing is perfect and that high-tech products are not exceptions to that rule. This historical epoch was called the 1960s, 1970s, and 1980s. When these antediluvian computer users found shortcomings in our industry's products, they weren't surprised, and they found and implemented workarounds that still let them accomplish their goals.

That time is long gone. The computing industry now sells much more complex equipment directly to people who not only don't know how any of it works, they don't even have reasonable ways to think about it. Most people don't know how an internal combustion engine works either, but their intuitions are good enough to mostly keep them out of trouble: Spark plugs fire a combustible fluid, and the resulting controlled explosion is translated into a rotational motion of a shaft, which turns the wheels. What is their equivalent understanding of how a single transistor works, let alone how billions of them work together in a computer? In some sense, when things are working properly, users don't have to understand them. It's when they fail that these intuitive models become important.

Without a working understanding, buyers aren't well equipped to judge the compromises built into high-tech products. Judging from the various Monday-morning-quarterbacking Web sites, even techie savants fall prey to the "better means better at everything" syndrome when comparing different products.

When engineers do stupid things, the technology itself amplifies the effects.

Buyers don't necessarily see how one product compromise relates to any others. If performance is obtained by jacking up thermal power, they'll complain about the power and take the performance as a given. Buyers aren't in a good position to fairly evaluate alternative offerings. I think that's why side-by-side comparison tables are perennially popular in trade magazines. Buyers have only one product, with that product's intrinsic set of tradeoffs, and their tendency will therefore be to notice what's "wrong" with it and take the rest for granted.

Engineers are worse than anyone else at this—we're always looking for ways to improve everything we encounter. I can reliably report that this causes occasional exasperation in people around us, such as spouses. As Nickel Creek says, "Others have excuses … I have my reasons why."

COMPROMISES AND GAFFES

Not all of our technical age's shortcomings are ascribable to unavoidable design compromises. Sometimes even engineers do stupid things. It's just that when nonengineers do stupid things, the effects are usually localized, but when engineers do stupid things, the technology itself amplifies the effects.

Gaffes are classic engineering blunders, the kinds of choices that look bad no matter how you look at them. A clue that you're seeing a gaffe is hearing yourself say, "That's ridiculous" or "That's never going to work" or "What kind of certified, fully licensed professional moron designed this piece of doodoo?" (This is a particularly painful observation if you happen to be the licensee in question.)

There's a crucial difference between a well-considered engineering tradeoff and a gaffe. For example, a traffic light was installed at an intersection close to my house a few years ago. There are sensors in the pavement, presumably intended to smooth the flow of traffic: If a lot of cars are flowing through the intersection, the green light is longer. If yours is the only car at the intersection, you should get the next green light.

At least, those are the considerations I would have thought important. But that's not what I observe.

Imagine I'm heading for the airport at 4:30 a.m.—unfortunately, not a hypothetical example. I approach the intersection to make a left turn. But before I get to the turn lane, I must first drive over the sensor that detects long, straight flows. This causes the light to immediately switch to a straight-ahead green. This is teeth-grindingly stupid because only one sequence is possible: straightahead green for a minimum of 15 seconds if no cars detected, then yellow and red; turn-lane green for opposite direction for a minimum of 15 seconds even if no cars detected; straightahead green for opposite direction for a minimum of 15 seconds if no cars detected.

In other words, my approach to the intersection causes the lights to go green for every other possible source of cars, even though the system's own sensors are telling it that no cars are there. It feels almost personal, like the light controller has some kind of vendetta against those of us who live uphill of this design atrocity.

I've discovered that if I veer out of my lane enough to miss the initial sensor, thus making my turn lane the first sensor the controller detects, I get a

green light in about 4 seconds. I consider the rigid light-sequencing algorithm in this design to be an outright design gaffe.

Ill-conceived features

My sister used to drive a Fiat 124 with a four-cylinder engine that had a really disturbing design feature: The distributor cap was mounted under the engine, pointing forward, and down at the ground. If the weather was dry, or the spark plug wires had no minute cracks in their insulation, this was a workable arrangement of engine parts. Under any other circumstances, the engine would sputter and cough or run on *N* cylinders for *N* considerably less than 4.

It reminded me of the Ford Pinto, in which the gas tank was placed in such a way that a commonplace rear-end collision could and did result in ghastly explosions and fires. I don't know why the Pinto designers thought this tank placement was reasonable, but (for what it's worth) the US court system considered this a design gaffe, not a reasonable compromise. (On the other hand, the US court system thinks driving around with a boiling hot cup of coffee between the driver's legs is a reasonable scenario. Don't get me started.)

Questionable design choices

Computer user interfaces are rife with questionable design choices. Donald Norman has written several books exposing some of the more egregious ones. For example, while using a Windows computer system, if you take out the removable media before letting the application close, you get a dialog box that reminds you the media is missing. You mutter to yourself, "So what? I'm finished with that app. Go away." But pressing cancel doesn't help. The box momentarily disappears, but it immediately comes back. The computer is longing for its missing floppy, and it won't be mollified until you feed it one. It doesn't even have to be the right one; any floppy will do. The only way out is to insert another floppy in the drive and then hit cancel. Every time this happens to me, I feel like smashing the machine into the back of a Ford Pinto and hoping for the best.

Don't confuse legitimate design compromises with outright errors.

Computers aren't the only electronic devices with ill-conceived features, though. My relatively high-end DVD player is crippled with the "regionality" antifeature intended to combat global DVD piracy. I don't fault the intentions of the policy makers who wanted to do something about piracy, but I do fault their "solution." A few months after I got the device, I sat down to watch a movie and discovered my region 1 player now mysteriously thought it was a region 3 player. I had to return the unit to the store that sold it to me, and they fixed it.

When I pressed the service technician to disclose what had caused this nuisance so that I could avoid it in the future, he eventually told me he had fixed it with a secret sequence of button pushes on the standard remote. Instantly, the problem was clear: If he could reset the region via the infrared remote, then perhaps I had caused the same sequence inadvertently by operating other infrared-controlled electronics in the vicinity.

Perhaps setting region designations needs to be done via infrared for manufacturing reasons. But if it's done that way, the designers must make sure that the odds of a user inadvertently causing that sequence are vanishingly low, which was clearly not achieved in the case of my DVD player. Infrared control was probably a design compromise; not providing the proper safety margins on it was a gaffe.

There are no easy answers to complex design issues. You might not like my design compromises; perhaps I can't live with yours. This isn't necessarily bad—the market is usually big enough for both of our products. Just don't confuse legitimate design compromises with outright errors. Design teams are required to make compromises, and they must strive mightily to avoid gaffes. And if you ever design a traffic-light control system for deployment in my neighborhood, please don't single me out for abuse, or I'll park an old Pinto next to the controller and remove its floppy. ■

Bob Colwell was Intel's chief IA32 architect through the Pentium II, III, and 4 microprocessors. He is now an independent consultant. Contact him at bob.colwell@comcast.net.

Poznan University Team Wins $18,000 at CSIDC 2004

With a system design that promises to revolutionize safety and rescue protocols in wilderness areas, a team of students from Poland's Poznan University of Technology won the 2004 IEEE Computer Society International Design Competition (CSIDC) in Washington, D.C.

Their "Lifetch" system is a GPS-based unit for hikers or other outdoor enthusiasts to carry into areas beyond the reach of ordinary communications. An on-board RF transceiver periodically transmits to a central monitoring station data on the hiker's position, temperature, acceleration, and light levels. The units would also use ad hoc networking to communicate with one another and the monitoring station.

The members of the Poznan team, Wojciech Jaskowski, Krzysztof Jedrzejek, Bartosz Nyczkowski, and Stanislaw Skowronek, will share the $15,000 first-place team prize. The "Lifetch" project also garnered the $3,000 Microsoft Multimedia Award, which recognizes the most interesting, innovative, exciting, and appropriate use of multimedia.

Teams from Poznan University have finished in the top three at all but one of the CSIDC's past four competitions, including a first-place finish in 2001. Photos of these winning teams are posted on the Poznan campus.

Competition at this year's CSIDC finals was especially tough. Said judge Paul Maj of Edith Cowan University in Australia, "While the Poznan team was really a clear winner, it was very difficult to place the other projects. The quality of the entries was uniformly high."

CSIDC 2004

More than 200 teams entered the initial phase of CSIDC 2004. An overarching theme, "Making the World a Safer Place," guided the entrants, though few other restrictions were imposed beyond a $400 spending limit for hardware.

Unlike other computer design competitions, which focus on software programming or similar tasks that can be completed in a single day, the CSIDC is the only long-term, project-based challenge open to undergraduates. Student teams competing in the CSIDC spend more than six months preparing for the final round of competition.

Said CSIDC chair Alan Clements of the University of Teesside in England, "In the CSIDC, student teams function like industry. This is giving students a chance to show teamwork. In the real world, ideas aren't created and marketed by one person."

Each year, competing teams go through three steps to reach the CSIDC finals. Shortly after beginning work on their projects, teams submit an interim report for judges to review. Projects not meeting minimum requirements are dropped from further consideration. As the live finals round draws near, teams submit a 20-page report providing a prospectus of their projects. A panel of judges then invites the 10 most promising teams to compete at the CSIDC finals.

Humboldt University team mentor Nikola Milanovic, a past student competitor at CSIDC, discussed how the Computer Society event compares to

CSIDC Chair Alan Clements (left) presents top honors to the team from Poland's Poznan University including mentor Jan Knigt, Wojciech Jaskowski, Krzysztof Jedrzejek, Bartosz Nyczkowski, and Stanislaw Skowronek.

Editor: Bob Ward, *Computer,* 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; bnward@computer.org

other student projects in computing. Said Milanovic, "There is another computer programming competition that gives you about five hours to complete an assigned task. This one takes [several] months. At the university, there are no resources for this type of thing. We really get a chance here to learn from our mistakes. There is no other competition like it."

CSIDC DEMONSTRATIONS

During the two-day world finals event, competitors have two opportunities to present their projects to the judging panel. An informal day of poster displays, including time for interviews and Q&A, preceeds an intensive day of scheduled presentations by each team. Winners are announced at an awards dinner on the second night.

Exhibit floor

On day one of CSIDC 2004, the finalist teams showcased their projects in displays that featured models, charts, brochures, and demonstrations of both hardware and software. The judges had the opportunity to circulate in small groups, questioning the competitors and informally evaluating the conception and execution of the prototypes. Teams promoted their projects to judges and other observers by handing out pamphlets and press releases.

The Lahore University of Management Sciences, for example, provided promotional materials that described their "SensUS Structure Security System" as being a "one-box solution to ... safety and security concerns" in high-rise steel-framed structures. The system consists of a wireless network of sensors that track the structural health of a building. Team member Tashfeen Suleman noted that, "it's a bit complex here, but when it goes to market, it will be packaged to be more appealing." One goal of the CSIDC is for teams to produce prototypes with commercial development in mind.

The Iowa State team produced a four-color brochure that briefly detailed the capabilities and intent of their "Spatial Cue" device. The brochure also included a graphical representation of how the system would function in the field. Said Douglas Houghton of the Iowa State team, "The primary focus is urban search and rescue. Our system will let you put the message where you need it, when you need it."

Formal Presentations

On day two of the CSIDC finals, teams gave formal presentations before a panel of judges. This year Major Fernando Maymi, who teaches software engineering at the US Military Academy at West Point, served as chair of the judging panel.

Said Maymi, who has been involved with the CSIDC for three years, "The quality of the teams' efforts continues to improve. I was pleased last year. But this year, I am floored! The competition is getting tougher every year.

"Not only is it getting tougher, but the diversity is increasing. There are more countries involved, and the projects are becoming unique," Maymi continued. "Some have come with strong industry support, and others have been very good at scrounging. If they don't have what they need, they'll find a clever way to do it."

Competitors at CSIDC reported some of the same sentiments. Said Kirill Orlov of the University of Virginia team, "If we found that something we wanted to do was outside the spending limit, we tried to find a way

CSIDC 2005 and into the Future

Organizers of the IEEE Computer Society International Design Competition (CSIDC) are envisioning several changes to future competition cycles. In particular, planners are seeking ways to expand the competition without further taxing Society resources and volunteer commitment. For example, in the first stages of CSIDC 2004, Microsoft staged a preliminary competition among universities with which it maintains academic liaisons. Of the 29 teams competing in the Microsoft ChallengE (so named because of the Windows CE platform used by competitors), one advanced to CSIDC 2004's top 20.

CSIDC chair Alan Clements and other organizers would like for most, if not all, finalist teams to advance through similar region-based competitions.

Said Clements, a computing professor at the UK's University of Teesside, "We are actively promoting internal competitions at individual schools or within countries or regions where several schools would take part. This would expand the number of teams entering the competition without requiring additional time commitments from our volunteer judges."

For next year's competition, organizers have also chosen to drop the idea of an annual theme in favor of opening up the competition to interdisciplinary teams. Teams will now be allowed to have members who are majoring in fields outside of computer science and engineering. For example, a team that is entering a tsunami warning system would be encouraged to include an oceanographer or a seismologist on the project. The only requirement will be that systems submitted for judging have a demonstrable benefit to society in general. The 2005 theme is simply, "Going Beyond the Boundaries."

Clements noted, "These are really applications of existing technology. The challenge at CSIDC is to create something of value to society. In that spirit, we are opening up the competition even further next year by simply reiterating that theme. We're asking only that the systems provide some type of social benefit. We're not going to be restricting them to safety, or security, or what have you."

Teams wishing to enter CSIDC 2005 should apply by **1 December**. To learn more about the future of CSIDC, visit www.computer.org/CSIDC/.

Judges Paul Maj, Mike Lutz, Ann Gates, and Simon Ellis; Lahore team.

2005 Computer Society president Gerald Engel; Georgia Southern University's Bob Cook.

Lead judge Fernando Maymi; Microsoft's Janie Schwark.

CSIDC Chair Alan Clements, of the University of Teesside.

Instituto Militar de Engenharia team.

Lahore University team.

Parth Thaker, University of Virginia; Judges Shakeel Mahate and Marnie Salisbury.

Politehnica University team with Microsoft's Schwark.

CRA executive director Andy Bernat; Poznan University team.

to approximate that functionality with inexpensive, off-the-shelf technology."

Orlov, whose team contributed a water quality monitoring system called "The Polluter Must Pay," also commented on how real-world applicability played a central role in their development strategy. "If something would be too expensive for developing countries," he stated, that would contradict the competition's goal of creating devices that provide a benefit to society.

JUDGING CRITERIA

Judges evaluate entries based not only on their technical merit but also on their fidelity to the year's theme. Entries at CSIDC 2004 were judged on the basis of the following criteria:

- originality, innovation, and social usefulness of the project (30 percent);
- system specifications, algorithms, and implementation, including the design and construction of any tools that were developed in the course of the project (20 percent);
- achieving the design objective, including compensating for any known limitations (20 percent);
- creativity and ingenuity in the design and implementation (15 percent); and
- usability, manufacturability, marketability, and maintainability, including validation testing, performance measurements and evaluations, and their thoroughness (15 percent).

The judging panel for the CSIDC 2004 finals included judging panel chair Fernando Maymi, the US Military Academy at West Point; Andy Bernat, the Computing Research Association; Elizabeth Burd, the UK's University of Durham; Bob Cook, Georgia Southern University; Simon Ellis, Intel; Ann Gates. the University of Texas; Robert Graham, Toshiba; Shakeel Mahate, CSIDC sponsor ABB; Paul Maj, Australia's Edith Cowan University; Mike Lutz, the Rochester Institute of Technology; and Marnie Salisbury of MITRE. Other judges served as reviewers of the reports submitted at two points earlier in the project year.

THE TOP TEN

After a day of watching 35-minute presentations from each team, the judges had only two hours to decide on how to place the top 10 teams. At the awards dinner, CSIDC chair Clements announced that the Poznan University team had taken the first-place honors.

"We're very happy about winning," remarked Poznan presenter Krzysztof Jedrzejek. "The other teams were very competitive."

In second place was Politehnica University of Bucharest, with "eXpress! Help," an emergency locator system intended as an add-on to existing mobile phones. The system would automatically relay the location of a subject to neighboring devices via a low-range Bluetooth module. Map coordinates relayed to rescue services then direct responders to the subject. Bucharest team members Andrei Mihai, Marian Mihailescu, and Monica Toma will share a $10,000 cash prize.

Third place honors went to a team from Iowa State University for "Spatial Cue." This PDA-based system would allow users to place messages in three-dimensional space, based on a series of GPS coordinates. When a user enters the proximity of these coordinates, a visual or auditory cue would be transmitted to his or her PDA. Applications could include warning messages at disaster sites or more prosaic communications like shopping reminders linked to stores.

Iowa State University team members Shahzaib Younis, Douglas Houghton, and Melanie Davis will share a $6,000 prize. The Iowa State team also received the $3,000 Microsoft Software Engineering Award, which recognizes the team whose project exemplifies the best application of proper software engineering principles.

Clements presented the seven remaining teams with certificates of honorable mention and team prizes of $2,000. Those teams were Humboldt University, Germany, for the "Person Loss Avoidance System for Mobile Applications (PLAS.MA);" the Military Institute of Engineering, Brazil, for "Dangerous Load Monitoring Alarm System;" Lahore University of Management Sciences, Pakistan, for "SensUS Structure Security System;" National Taiwan University, Taiwan, for "Adaptable VIsionary System for Earthquake Resolution (ADVISER);" Tribhuvan University, Nepal, for "TremorFlash;" University of Pretoria, South Africa, for "Intelligent Elderly Care (iEC);" and the University of Virginia, for "The Polluter Must Pay."

INTERNATIONAL SCOPE

CSIDC organizers intend the competition to be an opportunity for undergraduate students from around the world to engage in direct competition, underscoring the global nature of the Computer Society's mission. Of the ten finalist teams in Washington, D.C., this year, only two were from the United States. This provided an additional challenge to many of the competition's finalists, who had to give both oral and written presentations of highly technical work in a second language.

"CSIDC is exciting and fun," said IEEE Computer Society 2005 president-elect Gerald Engel. "This is really a flagship event for the Society. It's one of the best things we do, and it generates enthusiasm for the profession. In the end, it's one of the things that makes us want to be volunteers."

GAINING SUPPORTERS

Primary financial support for CSIDC is provided by Microsoft, which has committed funding through the 2006 competition year.

According to Ivan Joseph, one of two Microsoft observers at this year's event, his company supports the competition as part of its efforts to reach

out to both practicing computer professionals and emerging student leaders.

"This is a 'warm touch' for us," said Joseph, "It allows us to meet one on one with those who are the future of the profession. It's a tribute to the Computer Society and the efforts of its volunteers. Our investment in academia is something that keeps the computing ecosystem healthy by emphasizing a broad set of skills."

Janie Schwark, also from Microsoft, remarked, "Microsoft supports CSIDC because it's something we truly believe in. It's an international platform for raising the profile of the computing profession."

Further support for CSIDC 2004 was provided by the IEEE Foundation and Zurich-based engineering firm ABB.

CSIDC 2004 volunteer judges and organizers (from left to right): Marnie Salisbury, Elizabeth Burd, judging panel chair Fernando Maymi, Simon Ellis, Mike Lutz, Bob Cook, Robert Graham, CSIDC Chair Alan Clements, Paul Maj, Andy Bernat, Ann Gates, Shakeel Mahate, and Microsoft representative Janie Schwark.

CSIDC 2004 JUDGES

The annual IEEE Computer Society International Design Competition relies heavily upon the efforts of volunteer judges throughout all stages of the contest. For further information regarding CSIDC, including instructions for entering the contest or volunteering as a judge, visit www.computer.org/CSIDC/. ■

Not A Member Yet?

Here Is What You're Missing...

Distance Learning

Members of the IEEE Computer Society enjoy FREE access to a comprehensive distance learning program for computer professionals. Join today and take any or all of 100 online course titles through our Distance Learning Campus. Subjects include Java, project management, Cisco networks, UNIX, Windows, XML, Oracle, SQL, and more.

Magazines and Journals

A FREE subscription to *Computer* magazine is included with your membership. You may also subscribe to other publications in your area of expertise at member discounts. Or subscribe to the *IEEE Computer Society Digital Library*—21 periodicals and 1200+ conference proceedings—for a complete online resource.

Conferences and Workshops

Enhance your knowledge and share practical experiences at more than 150 conferences, workshops, and symposia held each year worldwide. Members save at least 25% on registration fees and get advance notice of the meetings.

Other Valuable Benefits

- FREE membership in your local chapter
- FREE e-mail alias of Your.Name@computer.org
- FREE membership in up to four of 40+ Technical Committees
- FREE membership in 160+ Standards Working Groups
- Member discounts on hundreds of books and conference proceedings

Join the IEEE Computer Society Today!

Complete the adjacent membership application today. For fastest service, apply online now at

http://computer.org/join

IEEE

IEEE COMPUTER SOCIETY

2004 IEEE Computer Society Professional Membership/Subscription Application

Membership and periodical subscriptions are annualized to and expire on 31 December 2004.
Pay full or half-year rate depending upon the date of receipt by the IEEE Computer Society as indicated below.

Membership Options*

All prices are quoted in U.S. dollars

	FULL YEAR Applications received 16 Aug 03 - 29 Feb 04	HALF YEAR Applications received 1 Mar 04 - 15 Aug 04
1 I do not belong to the IEEE, and I want to join **just** the Computer Society	$ 99 ❑	$50 ❑
2 I want to join **both** the Computer Society and the IEEE:		
I reside in the United States	$189 ❑	$95 ❑
I reside in Canada	$170 ❑	$85 ❑
I reside in Africa/Europe/Middle East	$166 ❑	$83 ❑
I reside in Latin America	$159 ❑	$80 ❑
I reside in Asia/Pacific	$160 ❑	$80 ❑
3 I already belong to the IEEE, and I want to join the Computer Society. (IEEE members need only furnish name, address, and IEEE number with payment.)	$ 42 ❑	$21 ❑

Are you now or were you ever a member of the IEEE?
Yes ❑ No ❑ If yes, provide member number if known: _____

Add Periodicals**

	ISSUES PER YEAR	FULL YEAR Applications received 16 Aug 03 - 29 Feb 04 PRINT	ELECTRONIC	COMBO	HALF YEAR Applications received 1 Mar 04 - 15 Aug 04 PRINT	ELECTRONIC	COMBO
IEEE Computer Society Digital Library **BEST DEAL**	n/a	n/a	$109 ❑	n/a	n/a	$55 ❑	n/a
Computing in Science and Engineering	6	$42 ❑	$34 ❑	$55 ❑	$21 ❑	$17 ❑	$28 ❑
IEEE Computer Graphics and Applications	6	$39 ❑	$31 ❑	$51 ❑	$20 ❑	$16 ❑	$26 ❑
IEEE Design & Test of Computers	6	$37 ❑	$30 ❑	$48 ❑	$19 ❑	$15 ❑	$24 ❑
IEEE Intelligent Systems	6	$37 ❑	$30 ❑	$48 ❑	$19 ❑	$15 ❑	$24 ❑
IEEE Internet Computing	6	$39 ❑	$31 ❑	$51 ❑	$20 ❑	$16 ❑	$26 ❑
IT Professional	6	$40 ❑	$32 ❑	$52 ❑	$20 ❑	$16 ❑	$26 ❑
IEEE Micro	6	$37 ❑	$30 ❑	$48 ❑	$19 ❑	$15 ❑	$24 ❑
IEEE MultiMedia	4	$35 ❑	$28 ❑	$46 ❑	$18 ❑	$14 ❑	$23 ❑
IEEE Pervasive Computing	4	$41 ❑	$33 ❑	$53 ❑	$21 ❑	$17 ❑	$27 ❑
IEEE Security & Privacy	6	$41 ❑	$33 ❑	$53 ❑	$21 ❑	$17 ❑	$27 ❑
IEEE Software	6	$44 ❑	$35 ❑	$57 ❑	$22 ❑	$18 ❑	$29 ❑
IEEE/ACM Transactions on Computational Biology and Bioinformatics **NEW**	4	$35 ❑	$28 ❑	$46 ❑	$18 ❑	$14 ❑	$23 ❑
IEEE/ACM Transactions on Networking†	6	$44 ❑	$33 ❑	$55 ❑	$22 ❑	$17 ❑	$28 ❑
IEEE Transactions on:							
Computers	12	$41 ❑	$33 ❑	$53 ❑	$21 ❑	$17 ❑	$27 ❑
Dependable and Secure Computing **NEW**	4	$31 ❑	$25 ❑	$40 ❑	$16 ❑	$13 ❑	$20 ❑
Information Technology in Biomedicine†	4	$35 ❑	$25 ❑	$44 ❑	$18 ❑	n/a	$22 ❑
Knowledge and Data Engineering	12	$43 ❑	$34 ❑	$56 ❑	$22 ❑	$17 ❑	$28 ❑
Mobile Computing	4	$30 ❑	$24 ❑	$39 ❑	$15 ❑	$12 ❑	$20 ❑
Multimedia†	6	n/a	n/a	$38 ❑	n/a	n/a	n/a
NanoBioscience†	4	$30 ❑	$24 ❑	$38 ❑	$15 ❑	n/a	$19 ❑
Parallel and Distributed Systems	12	$40 ❑	$32 ❑	$52 ❑	$20 ❑	$16 ❑	$26 ❑
Pattern Analysis and Machine Intelligence	12	$44 ❑	$35 ❑	$57 ❑	$22 ❑	$18 ❑	$29 ❑
Software Engineering	12	$38 ❑	$30 ❑	$49 ❑	$19 ❑	$15 ❑	$25 ❑
Visualization and Computer Graphics	6	$34 ❑	$27 ❑	$44 ❑	$17 ❑	$14 ❑	$22 ❑
VLSI Systems†	12	n/a	n/a	$28 ❑	n/a	n/a	$14 ❑
IEEE Annals of the History of Computing	4	$31 ❑	$25 ❑	$40 ❑	$16 ❑	$13 ❑	$20 ❑

Choose PRINT for paper issues delivered via normal postal channels.
Choose ELECTRONIC for 2004 online access to all issues published from 1988 forward.
Choose COMBO for both print and electronic.

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

_____
Expiration date (month/year)

_____
Signature

USA-only include 5-digit billing zip code
☐☐☐☐☐

* Member dues include $17 for a 12-month subscription to *Computer*.

** Periodicals purchased at member prices are for the member's personal use only.

*** Canadian residents add 15% HST or 7% GST to total. AL, AZ, CO, DC, NM, and WV add sales tax to all periodicals. GA, IN, KY, MD, and MO add sales tax to print and combo periodicals. NY add sales tax to electronic periodicals. European Union residents add VAT tax to electronic periodicals.

**** Payable to the IEEE in U.S. dollars drawn on a U.S. bank account. Please include member name and number (if known) on your check.

† Not part of the IEEE Computer Society Digital Library. Electronic access is through www.ieee.org/ieeexplore.

For fastest service, apply online at http://computer.org/join

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
PO Box 3014
Los Alamitos, CA 90720-1314 USA
Phone: +1 714 821 8380
Fax: +1 714 821 4641
E-mail: help@computer.org

Allow up to 8 weeks to complete application processing. Allow a minimum of 6 to 10 weeks for delivery of print periodicals.

IF4H

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

APPLICATION MUST BE SIGNED

NOTE: In order for us to process your application, you must complete and return both sides of this form.

IEEE

IEEE COMPUTER SOCIETY

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 _____

E-Micropayments Sweat the Small Stuff

David Geer

E-commerce is certainly nothing new. Many people buy books, CDs, clothes, travel, and other products and services over the Internet every day. However, after years of online purchasing, one long-promised aspect of e-commerce has remained largely unfulfilled: electronic micropayments for items that cost $5 or less.

Now, though, there appears to be a growing demand for e-micropayment technologies that would enable the purchase of items and services such as MP3 music files, magazine and newspaper articles, Web-based comics, pay-per-view videos, database access, and driving directions.

Highlighting the demand for micropayments, Apple Computer says its iTunes Music Store sold about 70 million songs, at 99 cents each, during its first year of operation, ending April 2004, and 95 million songs as of July 2004.

In addition, personal publishing has become pervasive during the past few years, with many Web sites specializing in comics, music, and art. Content creators would like to make money from this type of material and thus are interested in e-micropayment technology, said Professor Andrew Odlyzko, director of the University of Minnesota's Digital Technology Center. Without such a technology, these providers generally give their content away.

Also, said Nitesh Patel, senior wireless analyst for Strategy Analytics, a market research and consulting firm, the boom in mobile phone use among young people has increased the demand for ways to buy low-cost ring tones, icons, and games.

The ability to buy inexpensive items conveniently would eliminate the need for buyers to pay large subscription fees for entire sets of material when they only want selected pieces of content. It could also reduce the incentive to share files illegally.

Providers of inexpensive content see micropayment technology as a way to generate revenue, said Kurt Huang, founder, chief product officer, and president of e-micropayment-technology vendor BitPass.

Using credit cards for each purchase, as is usually done with e-commerce, is impractical for inexpensive items because the credit card fees and associated costs eat up sellers' profits. In addition, online merchants must spend time and money checking a credit card holder's identity and available credit before approving a purchase, which is also impractical for small transactions.

Because of this, several companies are developing e-micropayment products, including BitPass's Core BitPass System and BitPass Studios; Firstgate's Click & Buy; Payloadz's PayLoadz.com system; Paystone's personal, merchant, and group-pay accounts (for handling high-volume payments such as commissions, rebates, or paychecks); and Peppercoin's Peppercoin 2.0.

Despite the promise, not all industry observers say the approach has a bright future. In addition, the technology faces several obstacles, including the need to convince sellers and users to trust and work with e-micropayments.

THE TECHNOLOGY IN DETAIL

Most credit card processors charge merchants a minimum fee of between 15 and 35 cents per transaction, said Dave McClure, director of the developer network for PayPal, a service typically used for larger online purchases. This would be impractical for providers of inexpensive content because it represents all or much of the profit on many sales.

Strategy Analytics' Patel said that in the late 1990s, micropayment providers like Beenz, DigiCash, and Flooz didn't succeed because the vast majority of online vendors were not interested in selling inexpensive items. In addition, at that time, consumers expected most online content to be free.

A challenging development task

Typically, buyers acquire low-cost digital content in one of two ways: by either giving a single content provider large prepayments that would cover multiple purchases or receiving a bill from a vendor after making numerous small purchases.

However, this requires individual content providers to track each customer's purchases, which may not be worthwhile for small transactions. Also, these relationships lock a buyer in to an individual provider.

Other key challenges for e-micropayment technology include security, ease of use, the ability to handle high transaction volumes, and record keeping related to banking regulations.

General approaches

In meeting these challenges, micropayment vendors have taken a number of general approaches.

Usually, the payment-service vendors' Web sites enable buyers to access, via a link or button that opens a second browser window, e-micropayment applications that can combine purchases from multiple online sellers.

In many cases, users of a single e-micropayment technology can make prepayments to an account, which they can then use for purchases from multiple content providers.

Some e-micropayment technologies don't process a user's small purchases until they total perhaps $20, noted Rob Carney, Peppercoin's vice president of marketing. The multiple purchases then incur only a single credit card processing fee, which is relatively low on a per-purchase basis.

With some e-micropayment systems, the money goes first from the buyer to the payment-service vendor, which then pays the merchants. With other systems, buyers pay sellers directly, via the e-micropayment application. E-micropayment providers get their revenue from charging fees to content sellers.

In general, vendors use passwords and encryption to provide security for e-micropayments. Also, e-micropayment applications no longer require that users install plug-ins or other software downloads, which used to be sources of security holes.

Payment-service vendors sometimes have Web-hosting companies operate their systems to provide a stable, reliable platform. Other reliability approaches include uninterruptible power supplies, backup power generators, secondary sources of Internet access in case the primary connection crashes, and other types of redundancy.

Micropayment systems also work with error-checking technology to reduce the chances of transaction-handling mistakes, noted Ossip Kaehr, Firstgate's chief technology officer.

Most vendors offer ease of use by making their e-micropayment interfaces look like the credit card interfaces that customers are used to.

There appears to be a growing demand for e-micropayment technologies.

MAJOR E-MICROPAYMENT VENDORS

The various commercial e-micropayment technologies are similar in many respects but differ in others.

BitPass

BitPass deducts the cost of small purchases from a buyer's account that has been prepaid via a credit card or some other payment system such as PayPal.

The approximately 1,000 providers that work with BitPass register their content and services with the company and install gateways on their Web-servers to the e-micropayment technology.

When buyers click on BitPass-enabled content on a seller's site, the e-micropayment system prompts them to enter their password in a new browser window. Once the system confirms that buyers have enough funds in their account, they can access the content.

BitPass pays merchants via PayPal or by using the Automated Clearing House electronic-funds-transfer system to deposit money directly to their bank accounts.

Firstgate Internet

Firstgate works with about 3,000 providers of articles, information, games, and other content and has 2.5 million customers, mostly in Europe.

For security, Firstgate uses passwords and secure sockets layer (SSL) encryption and also records the IP addresses of buyers' machines for reference in case problems occur.

The company charges content providers an up-front fee—which varies based on the complexity of the project—for set-up, integration, and consulting. Firstgate then tailors its systems with business rules for each participating merchant.

The company bills buyers' credit cards, debit cards, or phone accounts, once they have accrued a few dollars in charges. North American customers can buy material via a charge to their bank accounts.

Firstgate aggregates consumer micropurchases across merchants to enable a single processing of multiple transactions, explained Chief Operating Officer Ed Burrell. "This reduces processing costs, per-purchase credit card fees, and administration," he explained. Firstgate then remits money to merchants via checks or bank transfers.

PayLoadz

PayLoadz works with about 1,000 merchants and generates about $3 million per year in revenue, explained founder and lead developer Shannon Sofield.

Rather than having prepaid accounts with PayLoadz, users pay sellers directly through PayPal, which also verifies that the buyers have enough money in their accounts to make the purchase. Merchants pay PayLoadz a flat monthly fee based on their sales levels, rather than a percentage of each transaction.

PayLoadz delivers files via a secure server to the customer using PayPal's Instant Payment Notification system, which lets vendors integrate PayPal with their back-end operations. Buyers never see a PayLoadz interface.

The PayLoadz system consists of an application server that hosts the Web site and handles transaction processing, a file storage and delivery server, and a database server. Separating the components into servers tuned for specific roles improves performance and

reliability, Sofield said.

For security, PayLoadz uses SSL with 128-bit RC4 encryption.

Paystone Technologies

Customers in North America, Australia, New Zealand, and parts of Europe can access Web content after setting up prepaid Paystone accounts by mailing funds directly to the company using their bank's bill payment service or by depositing cash at any Bank of America branch.

Merchants that sell content via Paystone create links to the e-micropayment system. Customers follow the links, enter their Paystone password, and are redirected back to the content they want to purchase.

For security, Paystone uses 128-bit SSL encryption.

Peppercoin

Massachusetts Institute of Technology cryptography expert Ron Rivest and fellow MIT computer scientist Silvio Micali founded Peppercoin.

Buyers at participating Web sites who want to use Peppercoin see the e-micropayment technology's interface, which looks like a common credit card interface, noted Bob Nix, the company's vice president of engineering.

The application's new version, Peppercoin 2.0, doesn't require preenrollment or predeposit of funds. Consumers just enter their credit or debit card information with the merchant online as usual, and Peppercoin processes the transaction.

As Figure 1 shows, the application uses a *universal aggregation* technique to efficiently process many small interactions between multiple consumers and merchants as a few large transactions.

The company uses RSA BSAFE software to provide encryption and digital signing capabilities for the security of a buyer's information and the transaction's integrity. Digital signatures can authenticate a message's sender's identity and verify that no one has altered the original content.

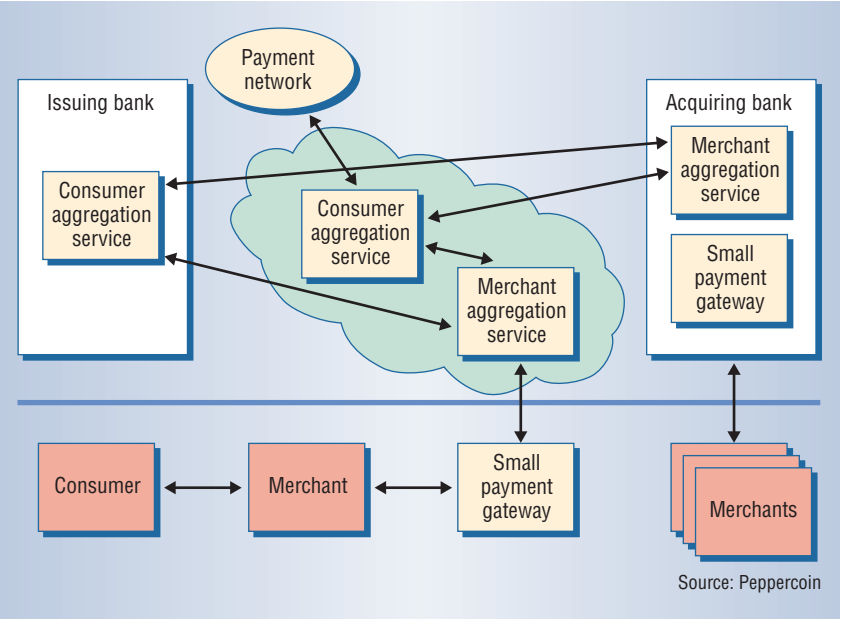


Figure 1. Peppercoin e-micropayment technology. When a consumer buys an item from a merchant, the transaction passes through a small-payment gateway to the payment service's system. Among other functions, the gateway feeds payment information into the system's service for aggregating multiple payments by consumers to merchants. The consumer aggregation service authorizes consumer spending, integrates with credit cards or other payment instruments, and presents a summary aggregated bill. The merchant service handles the sellers' end of the transaction. The banks that handle the money transfer have their own aggregation services.

To reduce overhead on small transactions, Peppercoin uses an online self-service customer-service module.

MACROCHALLENGES FOR E-MICROPAYMENTS

E-micropayment technology faces a number of obstacles to success. For example, said the University of Minnesota's Odlyzko, although e-micropayment technologies may be reliable, they run on the Internet, PCs, and browsers, which are not always dependable.

Economic factors

E-micropayment products deal in very small transactions and thus potentially yield low profit margins, noted PayLoadz's Sofield. Therefore, it may not prove to be economical for many companies to invest in and distribute the expensive e-commerce systems only to get thin profits dependent on high transaction volumes that might not materialize.

Also, Odlyzko said, content providers eventually may find that they can economically and successfully sell small items bundled into larger packages on their own, thereby eliminating the need for third-party e-micropayment technology.

Insufficient demand

Perhaps the key marketplace challenge for e-micropayment vendors is making content providers aware of their applications and selling them on the idea that their services are effective and worth paying for.

Currently, said Strategy Analytics' Patel, many merchants are not interested in using e-micropayment technology until they see one or two products prove themselves over time as being reliable and attractive to buyers.

Also, digital content is scattered all over the Internet. There are relatively

few sites where potential buyers can see large amounts of digital content grouped together. This makes it less convenient to buy the material, thereby reducing the number of online micropurchases and thus the demand for e-micropayment software.

Micropayment use is beginning to grow in Japan and with European news and publishing companies. If the technology becomes widely used, e-micropayment vendors may have to deal more effectively with such potentially thorny issues as refunds and customer support, explained BitPass's Huang.

In the future, said Odlyzko, "There will be progress in usability and, in particular, in incorporating micropayments into devices like cell phones and PDAs." However, researchers must figure out how to run e-micropayment applications on handheld devices, which have performance, memory, and battery-life limitations.

For example, Odlyzko noted, the cryptography required for e-micropayment security uses considerable processing power, which can run down mobile devices' batteries. Researchers are working to solve this problem.

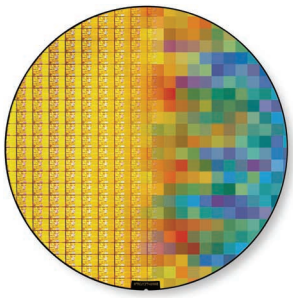
Technical improvements notwithstanding, Odlyzko said, micropayments are likely to make slow progress in the marketplace during the next five years simply because of low demand. "It is a technology with grand ambitions and limited prospects," he said.

Not all industry observers agree. "Micropayment technology will become more pervasive as more consumers pay for personalized content and services," said BitPass's Huang. Over time, said PayLoadz's Sofield, this will cause the use of micropayments to grow "at an incredible rate."

As this occurs, said Erik Michielsen, an analyst with ABI Research, the sale of low-cost content will increase, competition will grow, and the price of the material will decrease, which will encourage more use of micropayment technology. ■

David Geer is a freelance technology writer based in Ashtabula, Ohio. Contact him at geercom@alltel.net.

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



Intel Developer
FORUM

Exercise for the neck up

Systems Conference • September 7–9, 2004
Moscone South Convention Center • San Francisco, CA

The Intel Developer Forum is here — the invigorating biannual workout for the mind. Stretch the left side of your brain through in-depth training sessions. Develop the right side by building new and reaffirming existing relationships. As a whole, you'll have a stronger grip on the future of technology.

Register now for IDF. Your brain will thank you.

Register now: intel.com/idf/us
> or call 1-888-261-0971

Early Bird rates (until August 19, 2004)
> **Alumni:** $1,195 (save $500)
(Priority Code: ALMMTCTS)
> **New Attendee:** $1,495 (save $200)
(Priority Code: ATTMTCTS)

Gold Sponsors*

Silver Sponsors*

Sponsoring Community*

intel®

3D Searching Starts to Take Shape

Sixto Ortiz Jr.

Today, both the Internet and corporate data warehouses are full of all types of digital information, from simple text documents to complex applications. One type of information gaining in prominence is the three-dimensional object. From computer-aided design (CAD) drawings of complex engineering parts to digital representations of proteins and complex molecules, an increasing amount of 3D information is making its way onto the Web and into corporate databases.

Because of this, users need ways to store, index, and search this information. Typical Web-searching approaches, such as Google's, can't do this. Even for 2D images, they generally search only the textual parts of a file, noted Greg Notess, editor of the online *Search Engine Showdown* newsletter.

However, researchers at universities such as Purdue and Princeton have begun developing search engines that can mine catalogs of 3D objects, such as airplane parts, by looking for physical, not textual, attributes. Users formulate a query by using a drawing application to sketch what they are looking for or by selecting a similar object from a catalog of images. The search engine then finds the items they want.

Susan Feldman, research vice president of content technologies at IDC, a market research firm, said 3D image recognition and searching would be important for intelligence agencies examining photos, corporate marketing departments looking for one of many product images, or any individual or organization trying to find an

item in a collection of objects.

3D search engines could also help big companies quickly find whether they have certain parts in their inventories. For example, many big engineering and manufacturing companies have huge databases of parts they search for components to use in new products. Frequently, engineers spend a lot of time looking for such parts either manually or by older automated methods. If a part exists but engineers can't find it, the company must make it again, wasting valuable time and money.

3D search engines could eliminate these problems and also let users mine for other important parts-related information, such as cost or manufacturing method.

However, the technology faces several technical and marketplace hurdles before it can become popular and commercially successful.

SEARCHING IN THREE DIMENSIONS

Advances in computing power combined with interactive modeling software, which lets users create images as queries for searches, have made 3D-search technology possible.

3D searching has several important elements.

The voxel

The key to the way computer programs look for 3D objects is the *voxel* (volume pixel). A voxel is a set of graphical data—such as position, color, and density—that defines the smallest cube-shaped building block of a 3D image.

Computers can display 3D images only in two dimensions. To do this, 3D rendering software takes an object and slices it into 2D cross sections. The cross sections consist of *pixels* (picture elements), which are single points in a 2D image.

To render the 3D image on a 2D screen, the computer determines how to display the 2D cross sections stacked on top of each other, using the applicable interpixel and interslice distances to position them properly. The computer interpolates data to fill in interslice gaps and create a solid image.

Query formulation

True 3D search systems offer two principal ways to formulate a query: Users can select objects from a catalog of images based on product groupings, such as gears or sofas; or they can utilize a drawing program to create a picture of the object they are looking for. For example, Princeton's 3D search engine uses an application to let users draw a 2D or 3D representation of the object they want to find.

The 3D search process

The 3D-search system uses algorithms to convert the selected or drawn image-based query into a mathematical model that describes the features of the object being sought. This converts drawings and objects into a form that computers can work with.

The search system then compares the mathematical description of the drawn or selected object to those of 3D objects stored in a database, looking for similarities in the described features.

3D SEARCH ENGINES

Purdue and Princeton researchers have developed two major examples of 3D search engines.

Purdue University

Scientists at the Purdue Research and Education Center for Information Systems in Engineering, led by Professor Karthik Ramani, created a 3D shape search technology called 3DESS (3D Engineering Search System; http://tools.ecn.purdue.edu/~cise/dess.html). The engine is designed primarily to find computer-designed industrial parts.

For example, said Ramani, the system would be useful for any design and manufacturing company that deals with many CAD models.

3DESS gives users three options for searching industry databases: using a drawing application to sketch a part from scratch, penciling in modifications to an existing part to create a new search object, or selecting a part from a catalog of choices.

3DESS begins by converting query drawings into voxels via an algorithm that uses mathematical representations describing the solid shape.

Another algorithm uses *thinning*, which extracts only the voxels that represent the most important parts of the shape, to create a *skeleton* of the object. This is a reduced graphical representation of the object that displays its basic outline and topology, as Figure 1 shows.

3DESS then develops a *skeletal graph*, using an algorithm that analyzes and renders the skeleton in terms of three common topological constructs: loops; edges; and nodes, which are connecting points between loops and/or edges. Expressing an object in terms of these common constructs, plotted on a graph, reduces the overall amount of data in the object representation and makes it easier to store and index descriptions in the database.

According to Ramani, 3DESS also describes objects in terms of feature vectors, a set of mathematical representations of various aspects of the item's shape, such as its volume, surface area, and the number of loops, edges, and nodes.

The system can analyze a query by comparing either its feature-vector set or its skeletal graph with those of objects stored in a database.

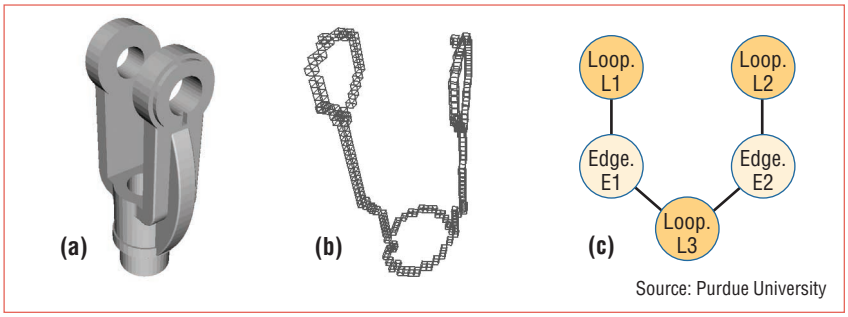


Figure 1. Purdue University's 3DESS 3D search technology starts with (a) a drawing of an object that a user wants to find in a database. The system uses algorithms to extract the most important parts of the shape to create (b) a skeleton, a reduced graphical representation of the object. 3DESS then develops (c) a skeletal graph, using an algorithm that renders the skeleton in terms of three common topological constructs: loops, edges, and nodes. This further reduces the amount of data in the object's representation and makes it easier to store, index, and search for descriptions within the database.

When the system retrieves models in response to a query, users can input which models more closely resemble the object they're seeking. 3DESS then uses neural-network technology to analyze users' feedback, learn more about what they want, and fine-tune both the current search and future similar searches.

Princeton University

Professor Thomas Funkhouser and his colleagues at the Princeton Shape Retrieval and Analysis Group (www.cs.princeton.edu/gfx/proj/shape) developed their Web-based 3D Model Search Engine. The engine works with a Java-based application that lets users produce, via their mouse, 2D renderings of objects to search for in a database.

Funkhouser and his team were not available for comment. However, *Computer* was able to speak with Michael M. Kahzdan, an assistant professor at Johns Hopkins University who, while a Princeton PhD student, worked on the 3D search project with Funkhouser.

The Princeton search engine, like its Purdue counterpart, uses mathematical representations to store the salient characteristics of 3D shapes in its database and to describe drawings that users submit as queries. The system compares queries' shape representations with those stored in the database to find the ones that are most similar.

Princeton's 3D search engine first renders a 3D object or drawing as a voxel grid and then maps this to a sphere, yielding spherical-based mathematical functions. The system then breaks the spherical functions into component parts that make up the ultimate *shape descriptor*, a graphlike abstraction that represents the item's overall shape.

An advantage of the spherical approach, Kahzdan explained, is that it lets the search engine work with shape descriptors regardless of the object's position. Thus, the system will match a query with the appropriate result even if the query drawing is positioned differently than the object in the database.

Searchers can refine their queries by adding a text description.

The Web-based 3D Model Search Engine has not yet been released commercially, according to Kahzdan.

CONCERNS

3D search technology faces several potential obstacles. For example, researchers are trying to improve accuracy so that search results more closely match queries.

In addition, researchers must improve search speed, said Nainesh Rathod, president and CEO of software vendor Imaginestics, which is licensing Purdue's engine for use in its products this fall. He noted that researchers are currently trying to improve speed by developing new search algorithms.

Ease of use is another important issue. For 3D search engines, ease of use will boil down to the manner in which people can enter queries, according to *Search Engine Showdown*'s Notess. Drawing shapes is more difficult than entering text queries. Thus, Notess said, 3D search engines will be most helpful for users with computer-aided drawing skills, such as those who work with CAD and graphic design.

Also, he noted, the quality of models drawn by users can be a factor if drawing is the main or only query approach. This would not be such a problem if users can select predrawn objects from catalogs, he added.

"Advances in systems allowing users to efficiently and effectively generate models would play an important role," Johns Hopkins' Kahzdan said.

Currently, said Kahzdan, 3D search focuses on submitting an image of an entire object as a query and receiving an entire object as a matching result. He said it would be important if, for example, researchers could give 3D search engines the capability to take a query that is a part of an object or scene and either return the entire object or scene as a result or find the part within a larger object or scene.

Developing this approach, he noted, is a much more challenging technical problem but could be useful for such purposes as recognizing health problems in the results of 3D medical imaging. For example, a physician could search a hospital's database of already-diagnosed magnetic-resonance scans to accurately diagnose a vexing abnormality found in a patient's scan.

Most of today's individual and corporate computer users don't conduct the type of searches that would justify paying for and learning to use sophisticated 3D search engines. Thus, said IDC's Feldman, the technology is likely to find only small, niche markets.

However, Notess said, within those markets, 3D search could prove useful and successful. ■

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

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

IEEE COMPUTER SOCIETY

IEEE

CALL FOR PAPERS

IEEE Real-Time and Embedded Technology and Applications Symposium

San Francisco, CA, March 7-10, 2005
Submission Deadline: September 27, 2004
Web site: http://www.cis.upenn.edu/rtas05/

RTAS 2005 seeks papers describing significant contributions both to state of the art and state of the practice in the broad field of embedded and open real-time computing, control, and communication. Contributions can cover timing or QoS issues in computation and networking, systems integration, scheduling, operating systems, middleware, software engineering, dependability, databases, programming languages, system development tools, performance modeling, and performance control. Special focus is on real-time and embedded applications ranging from industrial embedded applications such s aeronautics and automotive systems to open multimedia, telecommunication and mobile computing systems. Of particular interest are papers detailing experiments, implementations, and experiences in application domains that present new model problems or identify significant temporal or QoS constraints.

In order to maintain a close connection to the practice of embedded and real-time computing, RTAS 2005 is co-located with the Embedded Systems Conference, the leading conference and trade show in the embedded systems industry.

The scope of RTAS consists of the core area of real-time infrastructure and development and three additional areas broadly outlined below.

Core Area. Real-time Infrastructure and Development: This thrust continues from previous years with focus on embedded and real-time systems that exhibit significant timing constraints. Papers should describe significant contributions to the fundamental infrastructure, system support, or theoretic foundations for real-time computing. Topics include all of those associated with real-time computing platforms and development tools and techniques, such as real-time resource management, real-time operating systems, security, real-time Java, middleware, real-time CORBA, secure real-time systems, support for QoS, novel kernel-level mechanisms, power-aware real-time systems, real-time software component models, QoS-aware design, scheduling, and performance control.

Area B. Model-driven Real-time and Embedded Systems: This track solicits papers that increase our understanding of how complex large-scale real-time and embedded systems operating in heterogeneous and time-varying environments can be modeled, configured, composed, analyzed, checked, secured, certified, and controlled so that crucial system properties can be assured using techniques from Model Driven Architectures, Model Integrated Computing, Aspect-based Composition, Hybrid Control and other model-based techniques. Topics of interest for this track include, but are not limited to the following: empirical profiling and modeling of system properties; standardization efforts such as MDA and QoS-CCM; frameworks and tools for composition of multiple QoS properties; analysis, modeling and generation tools; applications of control theory to adaptive QoS management; techniques for representation and analysis of system properties; open research issues for model-driven composition of embedded systems; application scenarios and use cases for model-driven embedded systems; industry experience with modeling, analysis and control; architecture description languages and tools; model-based checking and certification of embedded systems; performance/efficiency of model-driven embedded systems; experiences implementing embedded systems with stringent QoS requirements; domain-specific requirements; integrating components, tools, and techniques from multiple sources.

Area A. Embedded Applications: We invite papers on industrial and other real-time and embedded applications. The focus of this track is on contributions associated with systems that are actually deployed in commercial industry, military, or other production environments, including automotive, avionics, telecommunications, industrial control, aerospace, consumer electronics, and sensors. Papers in this area include, but are not limited to challenges, requirements, model problems, and constraints associated with various application domains, use of real-time and embedded technologies in meeting particular system requirements, performance, scalability, reliability, security, or other assessments of real-time and embedded technologies for particular application domains, mining of architectural and design patterns from applications, and technology transition lessons learned.

Experience papers are especially encouraged within this topic, which may be less formal than traditional research papers, as well as proposals for panels which may offer a broader view of industrial activity on a particular subject.

Area C. QoS in Open Systems: The domain of real-time computing has broadened from primarily hard real-time closed embedded systems such as avionics and automotive applications to new open environments with other types of performance constraints such as the Internet and mobile computing systems. In such open environments independently developed system components and applications share common resources (often across a network) and need some form of performance assurances. Papers submitted to this track should address or extend the broad spectrum of performance assurance problems, QoS constraints, and quality metrics in open systems. Topics include but are not limited to interoperability of open QoS-aware application components, performance guarantees under uncertainty, combining/trading-off time or quality with other dimensions such as dependability, mobility, and security, QoS-aware communication, including Internet and Web-based applications, QoS in wireless and mobile computing, ad hoc networks, sensor networks, peer-to-peer computing, novel quality and performance metrics, user studies, and user-perceived QoS.

Conference Committee:

General Chairs: Tarek Abdelzaher, *University of Virginia*, Greg Bollella, *Sun Microsystems*, David Sharp, *Boeing*.
Program Chairs: Seongsoo Hong, *Seoul National University*, Oleg Sokolsky, *University of Pennsylvania*, Douglas Stuart, *Boeing*.
Finance Chair: Wei Zhao, *Texas A&M University*.
Publicity Chaits: Gerhard Fohler, *Malardalen University*, Tei-Wei Kuo, *National Taiwan University*.
Ex-Officio: Insup Lee (IEEE TC-RTS Chair), *University of Pennsylvania*.
Area Chairs: Core Area: Oleg Sokolsky, *University of Pennsylvania*. Area A: Seongsoo Hong, *Seoul National University*. Area B: Christopher Gill, *Washington University*. Area C: Douglas Stuart, *Boeing*.

Important dates:

Submission Deadline: Monday, September 27, 2004
Acceptance Decisions: Monday, November 22, 2004
Final Manuscript: Monday, December 13, 2004

For more information, including submission details, conference events, accommodations, area attractions, etc., visit the RTAS web site:
http://www.cis.upenn.edu/rtas05/

Companies Develop Nanotech RAM Chips

A new company has used nanotechnology to develop tiny, high-capacity memory chips that maintain stored data when a host device's power is shut off. The chips promise to be faster and more energy efficient than many comparable systems.

Nantero has created NRAM—the "N" stands for both nanotechnology and nonvolatile—a high-density RAM chip that saves its contents when a device is turned off or loses external power. The company will make the chips, while LSI Logic, a vendor of custom, high-performance semiconductors, will embed them in application-specific integrated circuits.

Nantero will manufacture its chip with carbon nanotubes, cylindrical structures consisting of hexagonal graphite molecules. Carbon nanotubes are up to 100 times stronger but one-sixth the weight of a steel object the same size.

Because of nanotubes' strength, NRAM could undergo many more read-write processes than other types of nonvolatile memory, said Nantero CEO Greg Schmergel.

The nanotubes also offer better electrical conductivity than many other materials frequently used in chips and thus enable faster performance. In addition, they are not affected by magnetic fields, which makes them particularly useful in harsh, magnetized environments such as outer space, Schmergel explained.

Nantero's design applies electrical charges to groups of nanotubes suspended over an electrode. Applying the opposite type of charge to the tubes and the electrode causes the tubes to bend down, touch, and bind to the electrode, thereby creating an electrical connection. Applying the same type of charge to the tubes and electrode causes the tubes to bend upward, thereby creating no electrical connection. The memory system reads these two states as the ones and zeros of binary data.

Because NRAM systems can place nanotubes so closely together, Schmergel said, they can offer high storage density.

NRAM promises to improve the performance of electronic devices that use nonvolatile memory. For example, the technology would be more energy efficient than flash memory—an inexpensive technology commonly used in digital cell phones and cameras, LAN switches, and other devices—because NRAM uses lower voltages and requires very little energy to move the tiny nanotubes.

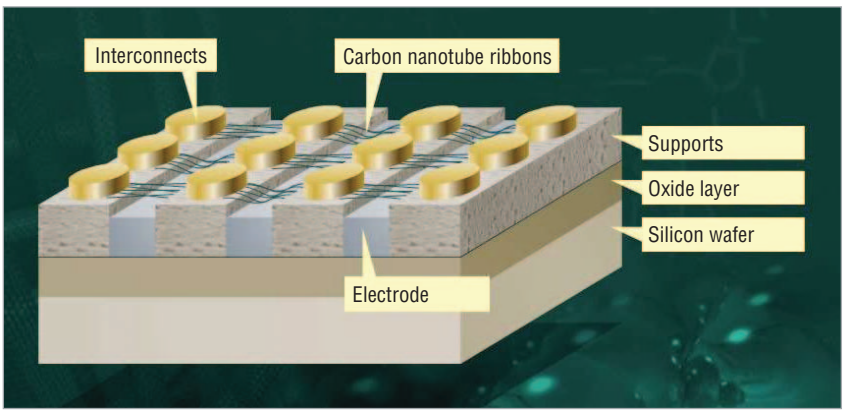
Because it is so much smaller, NRAM would be less expensive than static RAM, commonly used in cache memory and in digital-to-analog converters in video cards' RAM.

To succeed, NRAM must survive stiff market competition, said Richard Gordon, research vice president for Gartner Inc., a market research firm. Besides flash and SRAM, there are about 30 types of memory in varying stages of development and market readiness.

According to Gordon, NRAM is still a research project and won't succeed unless it gets even less expensive and more reliable and offers more features and functions.

Schmergel admitted NRAM needs work, such as reducing the overall chip's size to that of flash memory and increasing manufacturing yield to help lower costs. He said these improvements should occur by late 2005. ■

—Linda Dailey Paulson



Nantero's nanotech RAM chip applies electrical charges to nanotubes suspended over an electrode. Applying the opposite type of charge to the tubes and the electrode causes the tubes to bend down, touch, and bind to the electrode, creating an electrical connection. Applying the same type of charge to the tubes and electrode causes the tubes to bend upward, creating no electrical connection. The system reads these two states as binary data's ones and zeros. The high-capacity NRAM chip maintains stored data when a host device's power is shut off and promises to be fast and energy efficient.

First Smart-Phone Virus is Discovered

Security companies have discovered the first virus that targets smart phones. They say the Cabir worm did not contain malicious code but instead was a proof-of-concept program for possible future viruses with harmful content.

Cabir demonstrates that users should begin thinking of their cellular phones as computing devices, with all the potential risks, said Marty Lindner, team leader for the CERT Coordination Center, an Internet security organization.

Vincent Weafer, senior director with vendor Symantec's Security Response Unit, said Cabir is probably an ego-stroking stunt for its authors.

However, said Jimmy Kuo, director of antivirus research for vendor McAfee's Antivirus Emergency Response Team, Cabir could become dangerous when the basic code leaks into the wild, where other virus writers could adapt it.

Smart phones, combinations of cell phones and PDAs, are often used for more than just making calls. This creates multiple paths for malicious-code infection.

Cabir affects Nokia's Series 60 phones running both Symbian, the most popular smart-phone operating system, and Bluetooth, a short-range, wireless connectivity technology.

The virus scans for other phones running Bluetooth within the technology's 10-meter transmission range. The worm arrives on a recipient's phone as a Bluetooth message. Users must respond to a dialog box that asks if they want to accept a file someone is trying to send them. They would then have to download the file to activate it, similar to the steps that have allowed many viruses to spread in desktop systems.

After infecting a phone, the worm writes the word "Caribe" on the host's screen and creates and transmits a file containing the worm and instructions on how it should replicate. However, Cabir doesn't contain code that would damage a phone or its contents.

Two minor Cabir variants have appeared already, both from Cabir's creators, the 29a Eastern European group, which specializes in creating proof-of-concept viruses such as the recent 64-bit Rugrat. The group sent copies of Cabir to several antivirus firms, which say they have not found the worm in the wild.

According to Weafer, smart phones will become targets for viruses with malicious code in two to five years. Therefore, he said, phone users and service providers should begin upgrading security now.

"We've always known that as the wireless phone became a computing platform, a virus could happen," said Keith Nowak, spokesperson for Nokia Americas. Thus, he said, Nokia phones display a warning panel telling users they are about to download a file from an unverified source. They thus have the option not to accept the file.

To improve security in phones that use its OS, Symbian has started a Symbian Signed program that would require developers to digitally sign all applications for the platform. According to Symbian, its signature is a tamper-proof certificate that identifies the developer and says a given program is safe to download. Users could then refuse to install unsigned applications.

Experts also say that user education is critical. ■

—Linda Dailey Paulson

Users Feel Flush with New Digital Toilet

A Japanese firm has taken computer technology into new areas of human activity by developing a complex, computer-controlled toilet system.

The $5,000 Neorest toilet from Toto automatically raises and lowers the seat as needed, flushes the bowl, deodorizes the air, and cleans itself. The system also provides controls for heating the toilet seat and even replaces toilet paper with a bidet-like water sprayer and dryer, called a *personal cleansing system*.

The toilet is powered by a 128-kilobyte ROM chip with separate RAM chips, offering a total of 512 kilobytes of memory, located in a motherboard in the seat, noted Toto spokesperson Lenora Campos.

When used, she noted, a battery-operated wireless remote system controls all the features by bouncing infrared beams off the bathroom ceiling and onto a receiver on top of the toilet. The toilet also uses infrared technology to enable sensors to detect a user's presence.

"This is the first time technology of this magnitude has been used in a toilet," Campos noted.

The system has a manual override so that in case of a power outage, it can continue to function as a traditional toilet.

While the initial cost seems expensive, Campos said, the system saves money by lowering water use and eliminating the need for toilet paper.

In addition, Neorest utilizes fuzzy logic and past usage patterns to recognize when the toilet hasn't been used for a while and to automatically turn it off, like a PC's energy-saving sleep mode.

Otherwise, she said, the system uses as much energy as a traditional household appliance.

—Linda Dailey Paulson

Getting Electric Power via Ethernet Cables

Many offices use Ethernet cables for network connections, Internet access, and, with the advent of IP telephony, phone communications. Now, a number of companies are beginning to use the cables for something different: to provide electrical power.

Ethernet cables, using the IEEE 802.3af standard, carry 13 watts of power. This is enough for most battery-operated devices except laptops but considerably less than most household appliances, said Geoffrey O. Thompson, Nortel Networks' manager of standards development and recent vice chair of the IEEE 802 Working Group.

Thus, power over Ethernet (PoE) will work for low-wattage networked devices such as telephones, wireless network access points, and digital security cameras.

According to Thompson, the technology works with most common flavors of basic 10Base-T Ethernet over twisted-pair copper cabling. However, he added, it remains to be seen whether PoE will work with faster Ethernet versions.

PoE offers several advantages. For example, generic data cabling is considerably less expensive to install than electrical wiring. This is because, for electrical purposes, data cables are considered safety-extra-low-voltage cabling and thus have fewer regulatory requirements than electrical wiring.

PoE would also help companies avoid the expense of installing multiple electrical outlets just for wireless LAN access points, which require both a wired network connection and a power source.

By using Ethernet networking, PoE would enable improved remote troubleshooting and rebooting. Moreover, in the event of a power failure, networked devices could remain running.

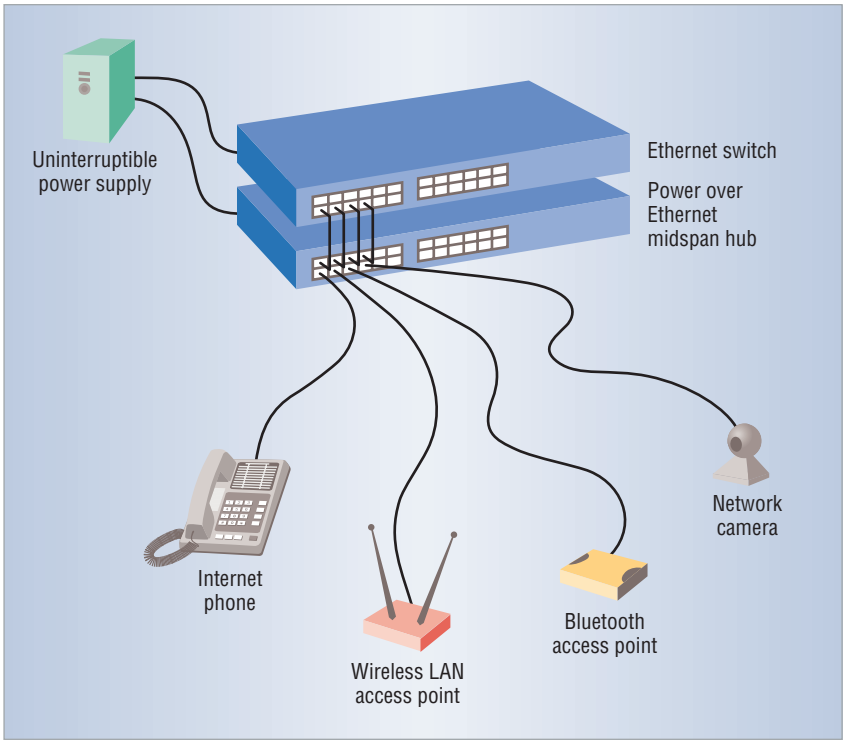
However, PoE switches cost 30 percent more than those used in standard Ethernet networks, noted David Willis, vice president of technology research services for the Meta Group, a market research firm. Users thus must determine whether the extra cost is worth the benefits for the size of network they operate.

Because of the advantages PoE offers, Willis said, companies are particularly interested in using the technology with their deployments of the increasingly popular IP telephony.

Data and electricity can travel over the same wires without interference because they operate at different frequencies. This phenomenon also lets service providers offer broadband Internet service over power lines ("Is Broadband over Power Lines About to Take Off?" *Computer*, June 2004, p. 18).

Devices that need more power than the new technology offers could still connect to PoE networks but get electricity from standard sources. ■

—Linda Dailey Paulson



Power over Ethernet technology lets users obtain low-wattage electricity directly through their network connections. In a typical implementation, power applied to an Ethernet switch is sent out through a PoE hub to various objects that are connected to the network.

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

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
The IEEE Computer Society's Web site, at **www.computer.org**, offers information and samples from the society's publications and conferences, as well as a broad range of information about technical committees, standards, student activities, and more.

OMBUDSMAN Members experiencing problems—magazine delivery, membership status, or unresolved complaints—may write to the ombudsman at the Publications Office or send an e-mail 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 the Publications Office:

- 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)

To check membership status or report a change of address, call the IEEE toll-free number, +1 800 678 4333. Direct all other Computer Society-related questions to the Publications Office.

PUBLICATIONS AND ACTIVITIES

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, industry trends, and product reviews.

Periodicals. The society publishes 12 magazines and 10 research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 160 titles every year.

Standards Working Groups. More than 200 groups produce IEEE standards used throughout the industrial world.

Technical Committees. Thirty TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

COMPUTER SOCIETY OFFICES

Headquarters Office
1730 Massachusetts Ave. NW
Washington, DC 20036-1992
Phone: +1 202 371 0101 • Fax: +1 202 728 9614
E-mail: hq.ofc@computer.org

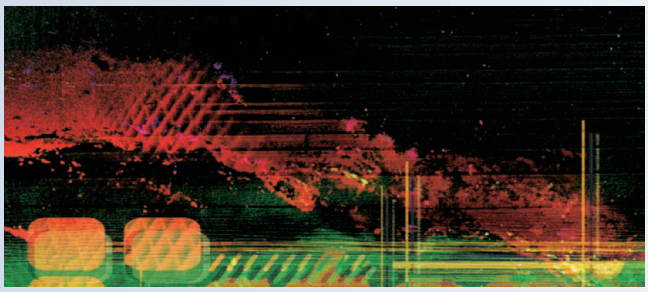
Publications Office
10662 Los Vaqueros Cir., PO Box 3014
Los Alamitos, CA 90720-1314
Phone:+1 714 821 8380
E-mail: help@computer.org
Membership and Publication Orders:
Phone: +1 800 272 6657 Fax: +1 714 821 4641
E-mail: help@computer.org

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
E-mail: tokyo.ofc@computer.org

IEEE

Modeling Complex Spoken Dialog

Increasing task complexity is challenging the models that underlie spoken dialog systems. A customized dialog language, evolved from building an actual system, uses core patterns that modelers can adapt to most applications.



Hans Dybkjær
Prolog Development Center

Laila Dybkjær
NISLab, University of Southern Denmark

Although not as complex as their research counterparts, commercial spoken dialog systems (SDSs) can handle increasingly advanced tasks. This relatively new complexity poses three immediate development challenges. The first is how to model the dialog when the user can select from many tasks. How can the system present the available choices and anticipate what the user will talk about next?

The second challenge is how to communicate with customers about SDS design. When the system is fairly simple, developers can easily learn how to handle the domain to be covered and have relatively little need to communicate with domain experts. When the domain is large and complex, however, developers are less likely to understand its nuances and thus must collaborate more closely with those who do. Thus, choosing the best communication method becomes another development task—a concern SDS developers have not had to deal with traditionally.

The final challenge is how to develop code efficiently for a large SDS when the design is likely to require updates throughout development. A complex application domain poses many language and flow problems, as the "Dealing with Complexity" sidebar describes.

We had to meet all these challenges in our effort to develop a commercial phone-based SDS that supplies information to employees about their holiday allowance. Key to this frequently-asked-questions (FAQ) system was our use of the Conceptual Dialog Language (CDL), a customized language that lets us express patterns specific to the system's dialog model while still providing a clear picture of the dialog model to domain experts and supporting model updates. CDL's power stems from its ability to capture the specifics of an SDS application while remaining flexible enough to support a range of modeling styles and requirements. To facilitate effective communication with domain experts, modelers can translate the dialog model in CDL to an HTML document, or they can compile it to a programming language for executable code.

FAQ SYSTEM

From September 2001 to December 2002, Prolog Development Center and NISLab developed a FAQ SDS that supplies holiday-allowance information on behalf of FerieKonto (www.ferie konto.dk), an institution that administers holiday allowances to Danish employees.[1]

As the right side of Figure 1 shows, the system is a phone-based SDS that uses the High-Level Dialog Description Language (HDDL)[2] and runs on ScanSoft's SpeechMania platform (www.scansoft.com). HDDL, a grammar and dialog description language that Philips developed in 1994, describes dialog in automatic inquiry systems that involve speech recognition. SpeechMania, a platform Philips developed and then sold to Scansoft, includes components for speech recognition, dialog and lexicon management, language modeling, and speech output. The package runs on a Windows PC with a telephone interface board.

Published by the IEEE Computer Society
0018-9162/04/$20.00 © 2004 IEEE

Dealing with Complexity

A well-known software-engineering challenge is how to deal with modeling complexity. Most practitioners use a divide-and-conquer strategy based on object orientation and other encapsulation. Others use communication abstractions in the form of patterns.

Spoken dialog system developers have similar solutions. Some have observed that concrete tasks, such as obtaining a date or a zip code, are common enough to be candidates for reuse as standard library subdialogs.[1] Commercial products like SpeechMania use similar ideas. The subdialogs encapsulate well-proven behaviors as long as the structured subdialog remains simple.

Unfortunately, a subdialog typically requires complex division into parameters, which is problematic because the subdialog is highly context sensitive. In Figure A, for example, the task is to turn "get a date value" into a subdialog and then call it twice to obtain a time interval. The subdialog would require at least three parameters: prompt, input grammar, and input interpretation. This is apart from the added problem of dialog flow.

At every user input, several error situations should give rise to different continuations. Including this as a parameter is difficult, but leaving it at the calling place could greatly increase its complexity.

Language

A common approach has been to divide grammars into subgrammars that correspond to subtasks and have active only the grammars that correspond to the tasks in focus. The drawback is that users and designers never work with the same division of the world. Some have attempted to add keywords to the active grammars to trigger tasks outside the focus. However, because speech recognizers become better, the trend is to have all grammars active all the time. Researchers are currently investigating how to combine subgrammars.[2]

```
[Assume that today is Monday.]
        S1.1: When do you leave?
        U1.1 Friday. [The user means the
first Friday after today]
        S1.2: Leaving Friday, when do you
return?
        U1.2: Wednesday. [The user means
the first Wednesday after Friday, not the
first Wednesday after today.]
```

Figure A. Example of how semantics depend on context, both on the immediate prompt and on the wider discourse and world state. Bracketed text is analysis, not dialog.

Flow

As Figure A shows, a period—in this case the time between Friday and Wednesday—is not simply a combination of two date subdialogs, even if you view the prompt text as a parameter. A full object model would require at least two dates (today, and a relative date), and the model still does not account for repair (what if the user had responded in U1.2 "No, Thursday," as a correction to "Friday"?). Clearly, using objects requires some level of sophistication.

Flow patterns are also important. Natural dialog tends to follow standard patterns that dialog management should support, not least to ensure that the system behaves uniformly so that the user can easily recognize what is going on.

References

1. D.G. Novick and S. Sutton, "Building on Experience: Managing Spoken Interaction through Library Subdialogues," *Proc. 11th Twente Workshop on Language Technology* (TWLT 11), S. Luperfoy, A. Nijholt, and G. Veldhuijzen, eds., University of Twente, 1996, pp. 51-60.
2. M. Rayner et al., "Plug and Play Speech Understanding," *Proc. 2nd SIGdial Workshop Discourse and Dialogue*, J. van Kuppevelt and Ronnie Smith, eds., 2001; www.sigdial.org/workshops/.

Information users need

In Denmark, all full-time employees have at least five weeks of vacation or holiday per year, but only those who have remained full-time workers the previous year can receive a salary during their vacation or holiday. Each month, employers contribute 12.5 percent of an employee's salary to FerieKonto, which saves it as holiday allowance to pay for that person's vacation the next year. Employees who continue in a permanent position receive their usual salary during holidays, and FerieKonto reimburses their employer. Employees who change to another position or have a temporary position, however, must fill in a holiday allowance certificate and submit it to FerieKonto. They then receive their saved holiday allowance instead of a salary.

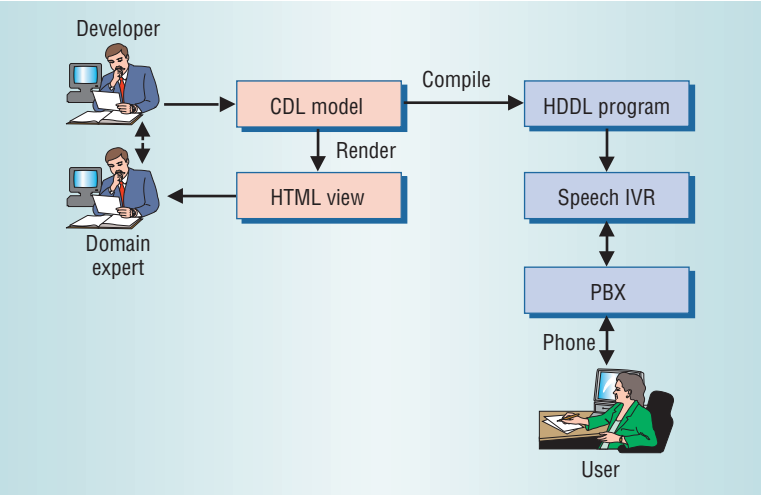


Figure 1. FAQ spoken dialog system (SDS) architecture. Danish employees call the SDS to get information about their holiday allowance. During system design, developers manipulated a CDL model, which they can translate to HTML for communication with domain experts or compile to a dialog program in HDDL, which the speech interactive voice response (IVR) then interprets through the switchboard (PBX).

```
                [...]
System   13.2   - Do you want information about payment of
                "small amounts" or "error in amount"?
                - PAUSE1000
User     13.1   error in amount
Recognised(error in amount)
Concepts(amount/score 736, error/score 771)
System   14.1   - Please remember that tax and labor market
                payments have been deducted from your holiday
                allowance before it is paid to us.
                - If you think the amount is wrong, please
                contact your employer.
                - Also contact your employer if a payment is
                totally missing.
System   14.2   - If you think you have received a wrong amount,
                please call again and talk to a case officer.
System   14.3   - Do you want information about "payment of
                small amounts," "problems with the employer," or
                to get a "phone number"?
                - PAUSE1000
User     14.1   get a phone number
Recognised (got phone number)
Concepts(phone/score 1000)
System   15.1   - Our phone number is: 48 20 49 10
System   15.2   - You may also say "address," "email," "fax,"
                and "web,"
                - PAUSE1000
Event    10     hangup
```

Figure 2. Dialog fragment, translated from Danish. System pauses are in milliseconds. After each user utterance, the system records what it recognized and to what degree using a recognition score of 0 to 1000. The score represents the strength of a match to certain concepts that the developer specifies.

Because the rules governing holiday allowance earning and distribution are complicated, many people need information about it, and the FAQ SDS must model all possible concerns. Questions to FerieKonto can relate to a specific person, such as, "How much money (or how many days) do I have in my account?" The questions might also be more general, such as, "Is Saturday considered a holiday?" or "Can I transfer a holiday allowance to next year?"

Although the FAQ SDS recognizes personal questions, it deals strictly with nonpersonal questions, referring other inquiries to an operator or a Web page. Figure 2 illustrates the kind of dialog a person can have with the system. Typically, callers fall into some combination of four categories:

- They have a problem and want an answer to their question now.
- They don't know exactly how to describe their problem.
- They lack information about the holiday allowance domain in particular.
- They've never interacted with an SDS before.

Dialog model design

Because the holiday-allowance domain is large and complex, we knew that the dialog model design would be a central issue in developing the FAQ SDS. Consequently, we devoted much effort to structuring the large information space and presenting it to the user, as well as to designing the user introduction, which had to convey both the system's capabilities and how users could get information.

Although design issues for standard SDSs are fairly well understood from both general and concrete perspectives,[3,4] the FAQ SDS is far from a standard commercial system. The most obvious difference is that it covers a large domain comprising many tasks, any one of which might be in focus at a given time. It is thus impossible to know in advance which part of the system knowledge a specific user will find relevant. These numerous tasks naturally give rise to many domain concepts.

As the "Choosing the Right Dialog Model" sidebar describes, application domain and user ability largely determine the selection of dialog model or combination of models. The application can have a range of tasks, concepts, and values, and that range is part of what determines the most appropriate model. The FAQ SDS, for example, has many tasks and many concepts, but each concept has few values, and the users are primarily novices.

For the FAQ SDS dialog model design, our options were to use a natural language model or a combination of the menu and natural language models. A menu model alone would have been infeasible because the FAQ SDS domain has too many concepts.

Using the natural language model,[5] which in essence invites people to speak their minds after the

system asks, "How may I help you?" was tempting, but we decided against it because users have only a limited understanding of what an SDS can do. Moreover, user tests during development indicated that giving the user too much freedom to initiate dialog was not compatible with how a FAQ system should work.

We confirmed these results later when we analyzed dialogs from the production system. Most users expected the FAQ SDS to guide them through the dialog. A few users took the initiative, using their own formulations to try to obtain answers. Still others seemed to ignore that they were speaking to a computer and provided too many details of the events leading to their current situation. Many users probably expected to speak to a human, so when they found themselves connected to a computer, they either hung up or did not speak but listened to the system's introduction, perhaps to extract some general information.

These results convinced us that too many users wouldn't know what to do, and only a few would take the initiative to use natural language and be able to focus their queries at the right detail level.

We ultimately decided to use a combination of menus and natural language to target the largest user group—people who didn't know how to describe the problem but needed an answer—and still accommodate the second group—people with direct questions about their holiday allowance. The menu points cover only the most frequently asked questions and serve as examples of what the system covers and how the user should ask questions.

CUSTOMER COMMUNICATIONS

Like any system, an SDS requires software engineering. As such, communication with the customer and domain expert is an important concern. For SDS development, the amount of required communication can differ considerably, depending on the system's size and complexity.

Previously, the target user group got information from a simple voice-response system, a Web site, or a case officer, whom they must call during office hours. Consequently, FerieKonto and its domain experts had been using diagrams as dialog models.

For a small system, diagrams are useful in showing prompts and dialog flow, and they provide an adequate overview for nondevelopers. For a large, complex system like the FAQ SDS, however, diagrams do not work as well.

First, the domain experts must easily understand the dialog structure and be able to provide feedback on the correctness and appropriateness of holiday-allowance rule formulations and combinations. To allow detailed feedback, the domain experts should be able to move around in the dialog model in a way that corresponds to what the implemented system would allow and to see all output, including rules. Second, the model presentation must be easy to update, since changes to the system must be consistent with what nondevelopers see.

```
Story: Amount, Error
- error in amount

r04_04      Amount, Error
- Please remember that tax and labor market payments have been
deducted from your holiday allowance before it is paid to us.
- If you think the amount is wrong, please contact your employer.
- Also contact your employer if a payment is totally missing.

r07_14      Payment, Error
- If you think you have received a wrong amount, please call again
and talk to a case officer.

Qo: Offer   - Do you want information about "payment of small
              amounts," "problems with the employer," or to get a
              "phone number"?
Small=Amount+Small | Amount=Amount+Small |
Problems=Problems+Employer |
Employer=Problems+Employer
Repeat=Amount+Error | Silence=Q: Instruction | (global)
```

Figure 3. HTML rendering of the dialog model. The links refer to approximately correct states. Fully correct transitions depend on dynamic information and need an interpreter. Overall, however, the HTML approximation is much simpler to read and explain to domain experts. A list of specialized links precedes the set of globally defined links.

We found that representing the dialog model as an HTML document, an excerpt of which is shown in Figure 3, satisfied both these requirements. The HTML links let domain experts follow the essential dialog flow; dialog content remains clearly visible; and with support from CDL, experts can easily update the HTML model.

CONCEPTUAL DIALOG LANGUAGE

A key concern was how to create efficient code that would capture the large, complex domain in which the FAQ SDS had to operate. The dialog model and the grammar would likely require several iterations, and the evolving specification had to be consistent with the envisioned implementation. These constraints were our primary motivation for creating CDL. Our aim was to tailor it to the FAQ SDS and to base it on general notions so that modelers of other SDSs could use it.

Figure 4 gives part of the dialog model in CDL with an XML syntax. CDL represents input as concepts and output as phrases concatenated into prompts. The dialog structure is partly static, modeled as collections of prompts or stories, and partly dynamic, expressed through conditions and links that control dialog flow. Using the XML syntax is purely a matter of convenience. CDL's combina-

Choosing the Right Dialog Model

The most common dialog models for commercial spoken dialog systems (SDSs) are command and control, menu, form-filling, and natural language—often in combination. As Table A shows, the exact model or combination of model elements depends on application complexity and the user's expertise with a particular SDS. Most current commercial SDSs belong to one of the first four rows in the table. Systems in the last two rows are still fairly rare.

Some researchers describe application complexity primarily in terms of language, but we view it in terms of tasks, concepts, and values. Language becomes abstract using domain concepts (station, departure and arrival times, and seat availability) that are definable as elements of meaning that the system needs to perform one or more tasks.

Linguistically, concepts correspond to subgrammars. A single task, such as obtaining information about train travel, can have a few concepts, and those concepts can have a few values (Is the seat available or not?) or many values (station names, times).

User expertise in the particular SDS under development seems to matter quite a bit. An expert user can be familiar with the domain and at least one SDS type, but even that user might have a hard time with a different SDS type. The user of a command-and-control system, for example, might use her background in the application domain and in using a natural language SDS to guess some of the functionality and some of the command words. But without knowing how this particular SDS works, she cannot fully exploit it. Thus, as a rule, the more novice the user, the more directive the dialog model.

Command and control

In this model, accepted system input is a set of predefined words and phrases, and the user initiates the interaction. Thus users must be familiar with the sys-

Table A. How application complexity and user expertise drive dialog classification.

Application complexity			User's domain and SDS expertise		
Tasks	Concepts	Values	Novice	Expert	Systems with these characteristics
One	Few	Few	Menu	Menu	Collect calling
One	Few	Many	Form filling	Form filling +*	Train information; airline ticket reservation
Many	Few	Few	Menu	Command and control	Simple car control, operating room
Many	Few	Many	Menu, form filling	Command and control, form filling +	Car control with other features, such as route planning
Many	Many	Few	Menu, natural language	Menu +, natural language	Simple FAQ
Many	Many	Many	Menu, form filling, natural language	Menu +, form filling, natural language	Complex FAQ, with accounts and names, for example

*A "+" indicates the use of speak ahead, in which users can provide the needed information in the order they choose, rather than waiting for the system to ask.

tion of abstraction and customizable compilations is what makes it work, not the use of XML.

Concepts as an input abstraction

The immediate description of user input is a grammar. Grammars are concrete and large—complexity we handle using concepts. In the FAQ SDS implementation, a *concept* is a semantic abstraction that HDDL maps to a concrete input grammar fragment.

Modelers can typically indicate to the speech recognizer what input they expect at a given point in the dialog by defining certain concepts to be active at that point. Having a small set of active concepts increases recognition quality, but only if the user says something within the set. If the user says something outside the set, the recognizer will not recognize it or will mistakenly view it as an active concept. In the FAQ SDS, we decided that the user will probably want to ask for everything at any time, so we made all concepts active all the time.

The FAQ SDS comprises 75 domain-related concepts, six user-provided metaconcepts—yes, no, help, repeat, bye, and other options—and four system-detected metaconcepts—barge-in, silence, too-long input, and not-understood input.

Structure from prompts and stories

After analyzing the domain and reviewing early drafts of the dialog structure, we identified four overall prompt types:

- *rule:* formal regulation governing holiday allowance;
- *information:* more details about holiday allowance or dialog continuation;

tem before they can really control it. The more command words the system includes, the more difficult it is to remember them all. Sample applications are voice control of operating rooms, such as Stryker's Endosuite, and in-car systems, such as those in some automobiles, in which drivers can control the CD player, radio, navigation system, and so on with voice commands.

Menu

In menu-driven systems, the user chooses from a list of items. This model is well suited for users who don't know the task and aren't familiar with the system. The system dictates to the user exactly what he can say, such as "Please say one of the following: pay bills, account balance, or transfer funds." Some systems offer skilled users the option of interrupting menu prompts (barge-in).

Menus appear in interactive voice-response or touch-tone systems and in their SDS equivalents. In these systems, dialog is always based on menus. Designers might also opt to use menus with other models, say, to let a user select a task from a voice portal. The selected task might then follow a form-filling structure.

Form-filling

In this model, the system has slots that the user must fill. A primitive form-filling model has only one slot, requesting a yes or no answer or a waybill or account number. More complex versions can require several pieces of information, such as the departure and arrival station and the arrival and departure times and dates for train travel.

This type of system typically guides the dialog, and the user needs no system or task knowledge. If the system dialog is poorly crafted, however, the user might not know exactly what to say and still have the system understand.

To accommodate users familiar with the task, SDSs using the form-filling approach often provide a speak-ahead option, where users can provide the needed information in the order they choose. In the train-travel example, a user could provide the departure and arrival stations, dates, and times in the same turn even if the system asked only for the departure and arrival stations.

Natural language interfaces

Interfaces that accept free natural language do not exist, but many systems allow limited natural language input, which the designer specifies. This kind of input fits well with the form-filling model because free input formulation can be a natural user choice. However, it is not a good fit with command-and-control systems, which are more restrictive.

Researchers are investigating conversational dialog that is not task oriented, but their efforts have yet to solve the problem of how to tell the user what the system can and cannot understand. The vast majority of changes to fine-tune a deployed system that allows some kind of natural language involve expanding the grammar. In the train example, the system must understand many formulations: "When is the next connection between Munich and Frankfurt today?" and "I need to be in Frankfurt at five pm tomorrow."

When the application domain has too many concepts, neither the menu nor the form-filling model is viable. Using full natural language queries would be optimal if SDS developers can overcome the limited subset of natural language that current systems are restricted to. This limitation might be acceptable for experts, but novice system users will always need help in determining what they can say and do.

- *question:* clarification of user issues; and
- *list:* options that the user can choose from.

These prompts are in essence reusable core patterns. Representing these patterns as a built-in syntax—basic building blocks—ensures that modelers can easily express them as needed and that the user sees a consistent structure.

We supplemented core patterns with patterns for specific situation types. A fixed repair pattern addresses any miscommunication: The SDS offers first instruction, then help, then the option of speaking with a case officer, and finally, if all else fails, politely closes the dialog. Another standard pattern is that after it gives the information, the system offers to provide more details related to that topic.

To allow for larger patterns and variations over prompt combinations, we introduced the notion of stories. A *story* combines several prompts into a larger structure. Figure 3, for example, is a partial story about errors in the amount paid. Stories are based on both rules and meta-issues, such as the user asking for help or other options or the system detecting silence or overly long input.

To compose stories, we used these informal design rules:

- Each single domain concept must have an associated story.
- If a concept concerns only one or two rules, the system provides these rules to the user immediately.
- If a concept concerns more than two rules, the system provides the most general rule and simultaneously indicates that more information is available on that topic.

```
<story topics="amount error" implicit="error in amount">
        <ruleRef rule="r04_04"/>
        <ruleRef rule="r07_14"/>
        <prompt type="Q" subtype="o">
                <phrase text="Do you want information about 'payment of small amounts,'
'problems with the employer,' or to get a 'phone number'?"/>
                <links type="meta">
                   <link ref="@story" cond="(got again)"/>
                   <link ref="QhI" cond="silence"/>
                </links>
                <links type="topic">
                   <link ref="amount_small" cond="(got small)"/>
                   <link ref="amount_small" cond="(got amount)"/>
                   <link ref="problems_employer" cond="(got problems)"/>
                   <link ref="problems_employer" cond="(got employer)"/>
                </links>
        </prompt>
        […]
</story>
[…]
<rule id="r07_14" topics="payment error">
        <phrase text="If you think you have received a wrong amount, please call again and talk
to a case officer."/>
</rule>
```

Figure 4. Fragment of the dialog model in CDL, translated from Danish. Coding CDL in XML was purely a matter of convenience. The power of CDL lies in its ability to be at once application-specific and extendable.

- If no rule is the most general, the system asks the user for more information.
- If related rules exist, the system offers them to the user after providing the information about the current story.

Flow via conditions and links

CDL captures the flow of prompts both within and between stories through links and uses conditions to control the flow of prompts. Every prompt has a list of links to other assigned prompts. We opted to have conditions on both links and stories and prompts to have greater flexibility in modeling style.

Link conditions are local in the sense that the system knows which prompt the dialog is in and what input it just received. Conditions on links enable a state-machine or flow-chart style—a traditional way to describe simple dialogs, such as those in voice-response systems. Link conditions answer the question, "Where should I go from here?"

Prompt and story conditions are global in that they are independent of local context. Conditions on stories and prompts enable a rule-based style more akin to AI systems. Prompt conditions answer the question, "Now that I'm here, should I enter?" The "talk to operator" story, for example, has two variants. If the operator is available, the system will tell the user that she is now being redirected to an operator. If an operator is not available, the system will tell the user when FerieKonto is open and explain how to consult a Web site for information.

Additional design support

In addition to typical dialog modeling and structuring tasks, CDL lets the modeler experiment with ideas that the programming language (HDDL in our case) might not support directly—ideas that could lead to a more varied and fluent dialog. For the FAQ SDS, we experimented with several ideas.

One is *reductive simplification*. Some stories have a partial overlap that is unavoidable because the same piece of information might be relevant in quite different contexts. To avoid redundancy, the system tracks which stories the user has heard. If the user has asked about payment of holiday allowance, the system will tell how this happens, but it will tell when the money will be paid only if the user did not already get this information from another story.

Another idea we experimented with was *barge-in control*. Barge-in is a technique that lets users interrupt system prompts. The system then stops talking and reacts to user input. Although barge-in is standard in SpeechMania, the platform does not include the ability to inform the system's dialog manager that a user has interrupted prompts. Consequently, the system never knows how much output a user actually heard before the interruption. Because part of our dialog design builds on knowledge about what information the user has already received, the FAQ SDS lets designers set barge-in as an event condition if desired.

The ability to *randomly switch among variations of frequent phrases*, such as "Are there other topics you want to ask about?" helps make the dialog less monotonous. Some variations that the FAQ SDS randomly switched were "Is there any other information you want?" and "Is there anything else you would like to know?" We generally have three variants of a frequent phrase.

The last idea we experimented with is *confidence control*. The speech recognizer provides a confidence measure between 0 and 1,000, which the developer can divide into three score ranges: low, medium, and high. We used 0-100, 101-200, and

```
1. COND ((V_currentName == MSN_amount_error__Qo)) {
2.        MO_comment("#COND ((V_currentName == MSN_amount_error__Qo))");
3.        M_setStory(amount_error);
4.        M_statPoint
5.        MC_preCond(MSN_amount_error__QhI, MSN_help__QhH, MSN_bargeIn__Qd);
6.        MC_preCond_Qo
7.        QUESTION (MH_allConcepts) {
8.              INIT {
9.                    MC_utterance_Qo
10.                   MP("-Do you want information about 'payment of small amounts,''problems
                      with the employer,' or to get a 'phone number'?");
11.                   MP_bargeIn_Qo
12.             }
13.             MC_events(MH_allConcepts);
14.       }
15.       MC_heard(V_amount_error__QoState);
16.       MC_postQuestion(3157, V_amount_error__QoState);
17.       MC_nextState(V_menu__more__QdState);
18.             IF ((^smallConcept && smallConcept.positive)) {
19.                   MO_comment("#(^smallConcept && smallConcept.positive)");
20.                   V_topicFound := TRUE;
21.                   MC_setState(V_amount_small__r07_03State);
22.                   V_catchLevel := V_catchLevel + 1;
23.                   MS_used(small, smallConcept);
24.                   BREAK;
25.             }
26. [...]
27. }
```

Figure 5. Machine-generated code for part of the fragment in Figure 4. The code, which constitutes the runtime version of the dialog model, is in the HDDL programming language. Macros (conventionally named with an initial M, MC, or MO) implemented much of the FAQ semantics. State names start with MSN. Variable names start with V.

201-1,000. The score value for any actual input depends on how well the acoustic input signal from the user matches the acoustic models of the concepts in the system's vocabulary.

A bad match can occur for several reasons, such as heavy background noise, a foreign accent or strong dialect, and out-of-vocabulary words. A low score causes the system to ask the user for confirmation of what she said: "Did you say maternity leave?" A medium score causes the system to repeat the key concept(s) the user mentioned before providing an explanation: "*You asked about maternity leave*. You cannot get your holiday allowance while you are on leave, but ...." The user then becomes aware of any misunderstanding as early as possible and can make a correction. A high score causes the system to give the user implicit feedback by mentioning in the information it outputs the key concept(s) it is responding to: "You cannot get your holiday allowance while you are on *leave*, but...."

IMPLEMENTED DIALOG MODEL

The final FAQ SDS dialog model comprises 85 concepts and 233 prompts, including 66 holiday allowance rules, structured into 102 stories. The dialog model is compiled either into human-readable form, the HTML in Figure 3, or into executable code, as in Figure 5. Although our executable code was in HDDL, it could also have been in VoiceXML, an upcoming standard for spoken dialog.[6] The model is about 2,400 lines of CDL (XML), which translates to nearly 50 pages of HTML description and 12,000 lines of HDDL code (compact, full of macro calls). The model also has 2,700 lines of grammar.

We could not have dealt with the SDS complexity or communicated the model to the domain experts without the CDL abstractions. So far, FerieKonto has not formally investigated users' satisfaction with the system and will find it difficult to do so, since the company is not allowed to log which users call the system. According to the FerieKonto case officer responsible for the system, which receives roughly 12,000 calls per year, they have received no negative user comments in the year and a half the system has been in use. The case officer also stated that the HTML version of the dialog model was easy for them to follow. Making changes and additions only in the CDL model guaranteed consistency between the HTML and HDDL versions.

The ideas of a customized language and core patterns apply well to SDS design. Widespread use of CDL would require a dedicated editor and integrated development environment, but for limited use, the solution we have described seems quite sufficient. ■

Acknowledgments

This project was supported by The National Agency for Enterprise and Housing and the Danish Ministry of Economic and Business Affairs. We thank the staff at FerieKonto for kind, competent, and enthusiastic cooperation.

References

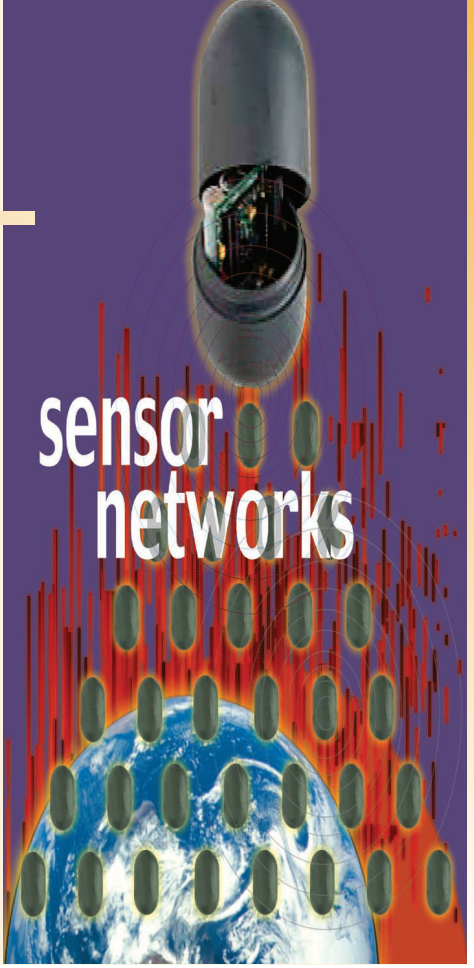
1. H. Dybkjær and L. Dybkjær, "Experiences from a Danish Spoken Dialogue System," *Proc. 2nd Danish*

HCI Research Symp., DIKU tech. report 02/19, Univ. of Copenhagen, 2002, pp. 15-18.
2. H. Aust et al., "The Philips Automatic Train Timetable Information System," *Speech Comm.*, vol. 17, 1995, pp. 249-262.
3. N.O. Bernsen, H. Dybkjær, and L. Dybkjær, "What Should Your Speech System Say?" *Computer*, Dec. 1997, pp. 25-31; http://www.disc2.dk/tools/codial/index.html.
4. B. Balentine and D.P. Morgan, *How to Build a Speech Recognition System. A Style Guide for Telephony Dialogues*, 2nd ed., EIG Press, 2001.
5. A.L. Gorin et al., "Automated Natural Spoken Dialog," *Computer*, Apr. 2002, pp. 51-56.
6. N. Leavitt, "Two Technologies Vie for Recognition in Speech Market," *Computer*, June 2003, pp. 13-16.

Hans Dybkjær *is a software developer at Prolog Development Center in Denmark, where he heads the speech group and works on interface specifications and the improvement of test procedures. He also led the development of the FAQ SDS described in this article. He received a PhD in computer science from the University of Copenhagen. Contact him at dybkjaer@pdc.dk.*

Laila Dybkjær *is a professor at the University of Southern Denmark's Natural Interactive Systems Laboratory (NISLab), where her interests include development and evaluation of interactive speech systems, usability aspects in interface design, and dialog modeling and theory. She received a PhD in computer science from the University of Copenhagen. Contact her at laila@nis.sdu.dk.*

Overview of Sensor Networks



Wireless sensor networks could advance many scientific pursuits while providing a vehicle for enhancing various forms of productivity, including manufacturing, agriculture, construction, and transportation.

David Culler
University of California, Berkeley

Deborah Estrin

Mani Srivastava
University of California, Los Angeles

The advances in science and technology are deeply intertwined. The telescope enables a deeper understanding of astronomy, the microscope brings bacteria into view, and satellites survey the Earth's surface, expanding what we can perceive and measure. Now, we can use computers to visualize, through numerical simulation, physical phenomena we cannot observe through empirical means.

This trend has advanced with the prolonged exponential growth in the underlying semiconductor technology. The number of transistors on a cost-effective chip and, therefore, the processing or storage capacity of that chip, doubles every year or two, following Moore's law. While it has provided ever more computing power, researchers are now applying this technology in ways that enable a new role for computing in science.

A given computing capacity becomes exponentially smaller and cheaper with each passing year. Researchers can use the semiconductor manufacturing techniques that underlie this miniaturization to build radios and exceptionally small mechanical structures that sense fields and forces in the physical world. These inexpensive, low-power communication devices can be deployed throughout a physical space, providing dense sensing close to physical phenomena, processing and communicating this information, and coordinating actions with other nodes. Combining these capabilities with the system software technology that forms the Internet makes it possible to instrument the world with increasing fidelity.

To realize this opportunity, information technology must address a new collection of challenges. The individual devices in a wireless sensor network (WSN) are inherently resource constrained: They have limited processing speed, storage capacity, and communication bandwidth. These devices have substantial processing capability in the aggregate, but not individually, so we must combine their many vantage points on the physical phenomena within the network itself.

In most settings, the network must operate for long periods of time and the nodes are wireless, so the available energy resources—whether batteries, energy harvesting, or both—limit their overall operation. To minimize energy consumption, most of the device's components, including the radio, will likely be turned off most of the time. Because they are so closely coupled to a changing physical world, the nodes forming the network will experience wide variations in connectivity and will be subject to potentially harsh environmental conditions. Their dense deployment generally means that there will be a high degree of interaction between nodes, both positive and negative. Each of these factors further complicates the networking protocols.

Despite these operational factors, deploying and maintaining the nodes must remain inexpensive. Because manually configuring large networks of

0018-9162/04/$20.00 © 2004 IEEE

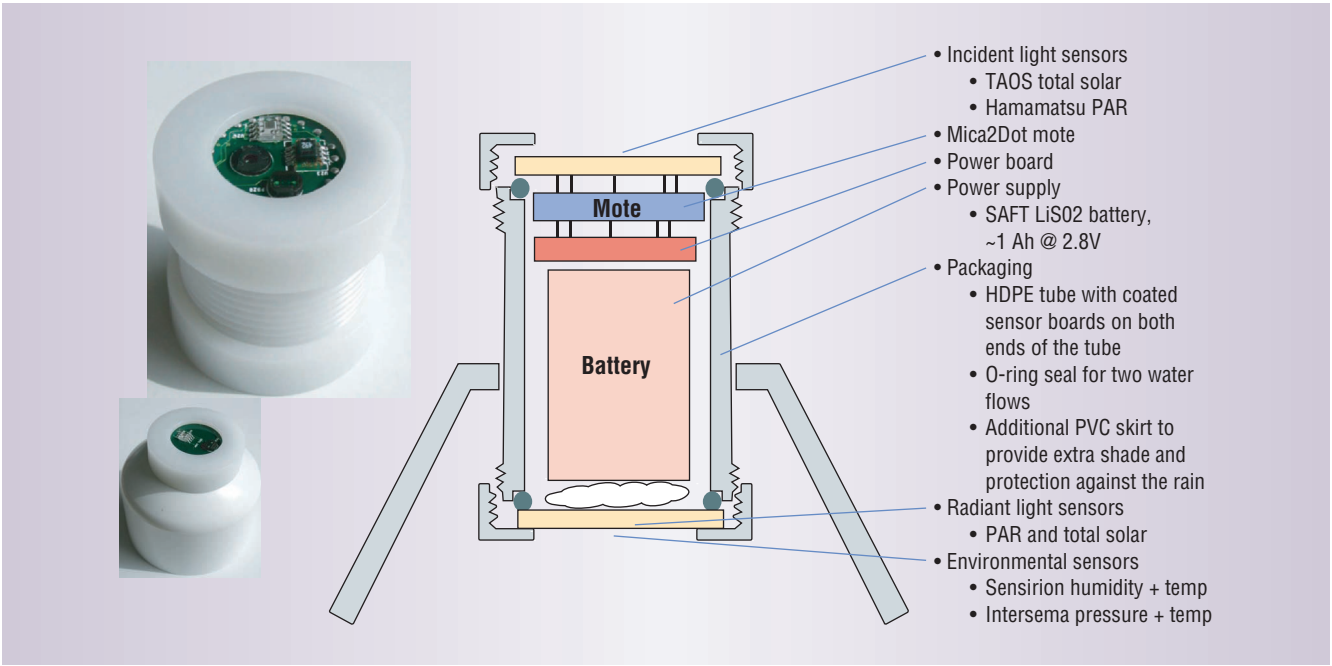


Figure 1. Wireless sensor node for environmental monitoring. An entire wireless microweather station fits in a tube about the size of a film canister.

small devices is impractical, the nodes must organize themselves and provide a means of programming and managing the network as an ensemble, rather than administering individual devices. Overcoming these challenges will let computer technology fill a new role in the progress of science.

SENSOR NETWORK APPLICATIONS

Although computer-based instrumentation has existed for a long time, the density of instrumentation made possible by a shift to mass-produced intelligent sensors and the use of pervasive networking technology gives WSNs a new kind of scope that can be applied to a wide range of uses. These can be roughly differentiated into

- monitoring space,
- monitoring things, and
- monitoring the interactions of things with each other and the encompassing space.

The first category includes environmental and habitat monitoring, precision agriculture, indoor climate control, surveillance, treaty verification, and intelligent alarms. The second includes structural monitoring, ecophysiology, condition-based equipment maintenance, medical diagnostics, and urban terrain mapping. The most dramatic applications involve monitoring complex interactions, including wildlife habitats, disaster management, emergency response, ubiquitous computing environments, asset tracking, healthcare, and manufacturing process flow.

Environmental monitoring

Many initial WSNs have been deployed for environmental monitoring, which involves collecting readings over time across a volume of space large enough to exhibit significant internal variation. Researchers are using WSNs to monitor nesting seabird habitats and microclimate chaparral transects and to conduct analogous studies of contaminant propagation, building comfort, and intrusion detection. One example, monitoring the microclimate throughout the volume of redwood trees, helps form a sample of entire forests.

Redwood trees are so large that entire ecosystems exist within their physical envelope. Climatic factors determine the rate of photosynthesis, water and nutrient transport, and growth patterns. Substantial variations are known to exist over the volume of an individual specimen, and researchers believe that the microclimatic structure varies over regions of the forest. In addition, water transport rates and the scale of respiration may influence the microclimate around a tree, effectively creating its own weather. All these factors influence the habitat dynamics of species existing in and on the tree.

Researchers traditionally have performed ecosystem measurements by hauling a suite of instruments, weighing perhaps 30 lbs, up a tree by a winch attached high in the canopy. Serial cables then hang down to the forest floor, where a data logger collects measurements. The instruments obtain measurements at various elevations at distinct points in time over relatively short intervals.

Figure 1 shows a modern WSN used for environmental monitoring in collaboration with biologist Todd Dawson. An entire wireless weather station fits in a tube about the size of a film canister. On top, two incident-light sensors measure total solar radiation, specifically light and photosynthetically active radiation, the bands at which chlorophyll are sensitive. An identical pair of sensors on the bottom,

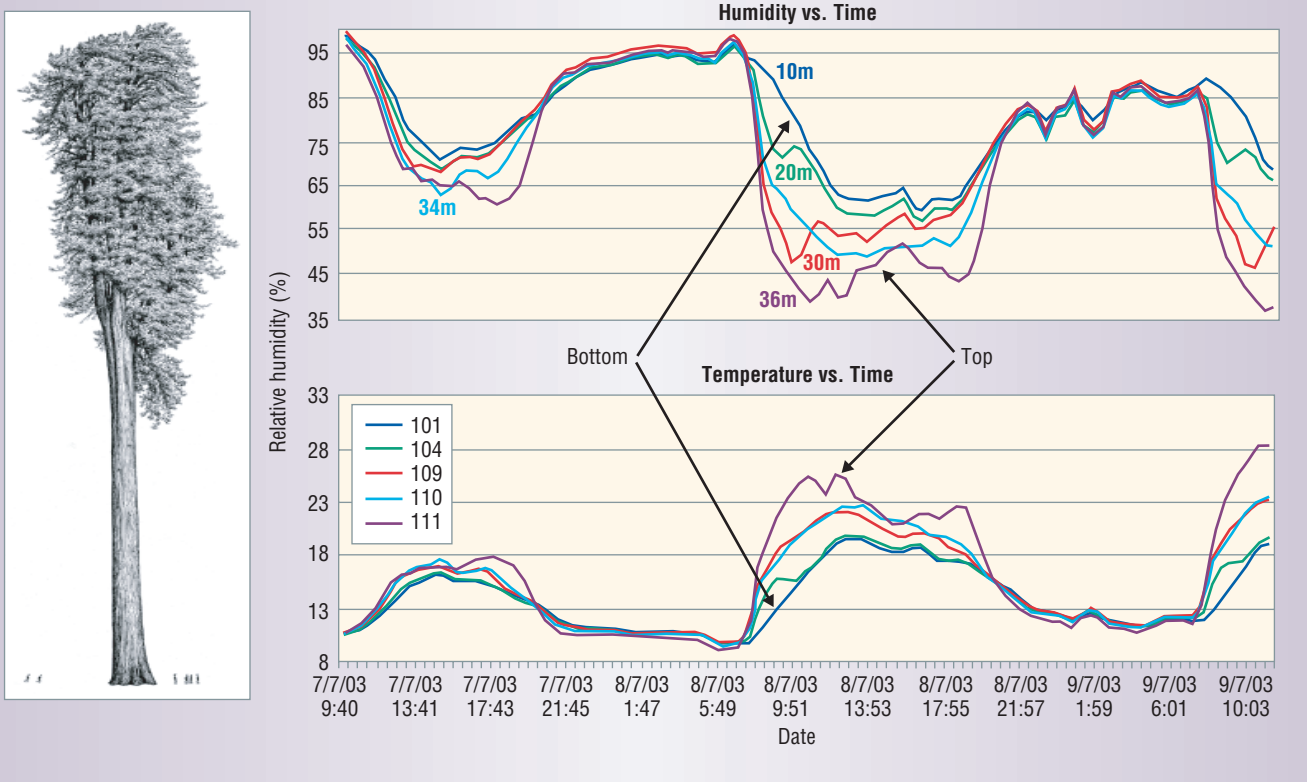


Figure 2. WSN climate data. The WSN samples climate data every five minutes and computes an average temperature at each elevation.

underneath a shade, measure radiant light. In addition, on the bottom there are environmental sensors to monitor relative humidity, barometric pressure, and temperature. The shade keeps rain off these sensors and prevents warming.

The weather-protected center of the tube contains a small computer, data storage, battery, and low-power radio to collect data, process it, and route information among the nodes and to the outside world. This provides a cost-effective means of obtaining simultaneous measurements at many points in the tree, spanning elevation and radial direction over a prolonged period.

Various facets of the forest, such as the center versus the windward, leeward, or sunward edges, can be similarly instrumented using small router nodes or long-distance uplinks placed in stretches of intervening forest to provide connectivity.

Figure 2 shows a temperature profile over three days, collected from 16 nodes at four elevations in a 35-meter study tree. The WSN samples climate data every five minutes and computes an average temperature at each elevation. The measurements show that within the expected daily cycle, the top of the tree experiences much wider climatic variation than the forest floor.

The network data also shows how weather fronts move up and down the tree. The top of the tree warms rapidly as the sun rises. This thermal front moves down the tree as the day warms, decreasing as it progresses. At nightfall, the situation reverses, with the top cooling below the base.

Variations in humidity are even more pronounced than climatic variations, further illustrating the importance of dense instrumentation because the tree moves tremendous volumes of water, which increases the humidity within the canopy. These weather fronts create powerful temperature and moisture gradients that could be instrumental in understanding growth dynamics, water intake, and nutrient transport over such large structures, yet they cannot be observed with sparse instrumentation.

In an initial study, researchers will likely extract all the environmental data from such a network and use it to identify biologically relevant features, such as a temperature front. Later, the network will likely perform the front-tracking algorithm within the tree itself, communicating only relevant statistical summaries, thereby increasing the network lifetime. As the ecophysiological model evolves, the researchers will likely refine the detection and tracking features.

Motion monitoring

Monitoring objects presents different challenges and opportunities. Many applications can be viewed as a form of condition-based maintenance. A physical structure, such as a machine, motor, airplane wing, bridge, or building has typical modes of vibration, acoustic emissions, and response to stimuli. Variations in these behaviors indicate wear, fatigue, or other mechanical changes. For example, a bearing often will squeak and shudder before it seizes up.

Using sensor nodes to perform structured analysis requires establishing a highly accurate time frame.

In addition to the sophisticated instrumentation of the actual wafer processing, a modern semiconductor fabrication plant can have several thousand vibration sensors attached to various pieces of routine machinery. A team of electricians tours the plant with a computing device that attaches to a sensor and logs a sample for a short period. The team then carries these logs back to a central computer, which analyzes them for signs of wear. Months elapse between visits to a particular machine. This scenario applies to a wide range of manufacturing and power generation plants.

WSNs offer an alternative approach: performing local processing at each device and transporting the data continuously to operations staff. Sampling rates are much higher than with environmental monitoring—typically about 100 Hz for vibration analysis and several kilohertz for acoustic analysis. This requires more energetic sampling activity and greater care in how the network performs sampling. Buffering the data requires greater amounts of storage, and the system potentially performs more extensive local processing on data chunks sampled intermittently.

Rather than transmitting large amounts of raw data, the sensor nodes can perform signal analysis, communicating only the modes of vibration or detected anomalies. Sensor nodes can monitor control networks to establish what is active when a sample is taken or even to determine when to sample. Because reducing the cost of obtaining and processing data reduces costs overall, increasing the timeliness of analysis can improve plant performance.

The analysis of structural response in, for example, bridges, buildings, and airframes, places a further requirement to use data collected at different points in the structure in spatial-temporal analysis. This requires establishing a common, highly accurate time frame across nodes. Nodes share time-correlated raw or processed data to perform the structural analysis. For example, the sensor data from one instrument can be used as an input to a model-based analysis at each of several other points in a structure and compared to sensor data at those points. Researchers refine these models by using them iteratively in normal circumstances to detect anomalies.

EMBEDDED NETWORK TECHNOLOGY

WSNs merge a wide range of information technology that spans hardware, systems software, networking, and programming methodologies.

Microprocessors, power, and storage

A sensor network node's hardware consists of a microprocessor, data storage, sensors, analog-to-digital converters (ADCs), a data transceiver, controllers that tie the pieces together, and an energy source. Recently, a new operating point has emerged that suits all these components. As semiconductor circuits become smaller, they consume less power for a given clock frequency and fit in a smaller area. In simple microcontrollers, miniaturization increases efficiency rather than adding functionality, allowing them to operate near one milliwatt while running at about 10 MHz. Most of the circuits can be powered off, so the standby power can be about one microwatt. If such a device is active 1 percent of the time, its average power consumption is just a few microwatts.

This scale of power can be obtained in many ways. Solar cells generate about 10 milliwatts per square centimeter outdoors and 10 to 100 microwatts per square centimeter indoors. Mechanical sources of energy, such as the vibration of windows and air conditioning ducts, can generate about 100 microwatts.

A typical cubic-centimeter battery stores about 1,000 milliamp-hours, so centimeter-scale devices can run almost indefinitely in many environments. However, low-power microprocessors have limited storage, typically less than 10 Kbytes of RAM for data and less than 100 Kbytes of ROM for program storage—or about 10,000 times less storage capacity than a PC has. This limited amount of memory consumes most of the chip area and much of the power budget. Designers typically incorporate larger amounts of flash storage, perhaps a megabyte, on a separate chip.

Microsensors

Sensors give these nodes their eyes and ears. Many materials change their electrical characteristics when subjected to varying environmental conditions. Sensors are manufactured so these changes are predictable over a certain range. For example, a thermistor is a variable resistor that changes smoothly with temperature. An ADC converts the voltage drop into a binary number that a microcontroller can store or process. Photocells and fog detectors work similarly, but they consist of finely interleaved combs separated by a material that uses incident photons or moisture to change resistance.

Many more sophisticated structures have been developed to detect other phenomena. These structures consume a few milliwatts and only need to be turned on a fraction of the time. Extremely efficient

ADCs have been developed so that the sensor subsystem has an energy profile similar to the processor.

Microelectromechanical systems (MEMS) can sense a wide variety of physical phenomena cheaply and efficiently. Researchers can use the processes for etching transistors on silicon to carve out tiny mechanical structures, such as a microscopic springboard within an open cavity. Gravitational forces or acceleration can deflect this cantilevered mass, causing powerful internal forces that cause changes in material properties or delicate alignments, which can be amplified and digitized. Manufacturers used the first major commercial MEMS sensor, the accelerometer, to trigger automotive airbag release.

Whereas high-precision piezoelectric accelerometers cost hundreds of dollars, MEMS provided sufficient precision for a few dollars. Once the devices entered mass production, they could ride the CMOS technology growth of modern chips to become increasingly accurate while remaining inexpensive. A wide variety of MEMS devices can sense various forces, chemical concentrations, and environmental factors.

Microradios

For some time now, manufacturers have added sensors to many appliances, vehicles, and gadgets. The breakthrough comes from communicating sensor readings to other devices, translating the physical world into information—bits—that the devices can transport, store, and process.

Radio components can now be manufactured using conventional CMOS technology, enabling wireless entry devices, pagers, walkie-talkies, cell phones, and wireless local area networks for mobile laptops. However, the amount of energy required to communicate wirelessly increases rapidly with distance. Obstructions—such as people or walls—and interference further attenuate the signal.

Wireless LANs and cell phones consume hundreds of milliwatts and rely on a powerful infrastructure. WSN radios consume about 20 milliwatts, and their range typically is measured in tens of meters.

For small devices to cover long distances, the network must route the information hop by hop through nodes, much as routers move information across the Internet. Even so, communication remains one of the most energy-consuming operations, with each bit costing as much energy as about 1,000 instructions. Thus, WSNs process data within the network wherever possible.

SYSTEMS CHALLENGE

Bridging the gap between the hardware technology's raw potential and the broad range of applications presents a systems challenge. The network must allocate limited hardware to multiple concurrent activities, such as sampling sensors, processing, and streaming data. The potential interconnections between devices must be discovered and information routed effectively from where it is produced to where it is used. There must also be a means of programming the ensemble.

To reduce energy consumption, WSNs process data within the network wherever possible.

TinyOS

Conventional operating systems such as Unix run well on a 32-bit microprocessor at 50 to 100 MHz, with several megabytes of RAM and a gigabyte or more of secondary storage. Today, this can be achieved in a handheld device that runs for several hours on a single charge.

A more typical operating point for WSNs is one year on a pair of AA batteries with a small fraction of these resources. Further, this application focuses on structured interaction with the physical world, rather than on complex human interactivity. The developers of the open source TinyOS tailored it for this application.

The TinyOS provides a framework for assembling application-specific systems that can handle substantial concurrency within limited physical resources. The software components and the underlying operating system support specific event-driven functionality. The lowest-level components abstract the physical hardware and deliver physical interrupts as sanitized, asynchronous events. Each component handles certain events and signals actions in other components, but they never consume processor cycles while waiting for future events.

Each application includes only the components it requires. For example, a small stack of components process sensor readings. The lowest components handle the ADC to obtain raw readings, whereas the higher ones filter and distill data streams for the application.

The network involves a more complex stack in which the lower levels deal with acquiring the radio channel, framing data streams into packets that receiving nodes can recognize. These components also perform error coding and channel scheduling, as well as detecting the arrival of incoming packets and processing them into input buffers.

Higher levels deal with buffer management, authentication, and multiplexing the network

The link layer transmits a structured series of bits that form a packet encoded in the radio signal.

across application components. A typical top-level application might receive and process a stream of filtered sensor readings, then deliver important notifications to the network. A second component would receive such notification messages, maintain a routing structure, and retransmit them along the next hop in a route to a data collection gateway.

Network sensor platforms

Berkeley motes and TinyOS are widely used for exploring systems issues and deploying pilot applications. The microcontroller provides a modest amount of RAM and program storage and contains an internal ADC. A simple frequency-agile radio with roughly the bandwidth of a modem provides the connectivity that developers can use to construct a network. Off-chip flash memory provides storage to hold both the program while it transfers through the network and the data buffering beyond the on-chip RAM. Several sensor boards have been designed for this platform.

The Intel iMote is a recent integrated design that uses a commercial chip with a powerful ARM microprocessor, storage, and radio integrated into a single package. The radio implements the Bluetooth standard, which is becoming widely used in laptops and cell phones. The radio operates at higher bandwidth and has a sophisticated frequency-hopping protocol.

Normally, the ARM processor would be devoted to managing the Bluetooth radio and transferring packets to and from a serial port. In the iMote, however, TinyOS runs directly on the ARM processor, providing a stand-alone system that services various sensors and routes, processes high-level information streams, and manages power consumption.

Most of the lower-level TinyOS components are implemented directly in hardware. These include an extremely low-power ADC and a very efficient radio developed in the Smart Dust project (http://robotics.eecs.berkeley.edu/~pister/SmartDust/). The entire design occupies only 5 square millimeters. It is estimated that at 1 percent active, the chip could run a hundred years on the energy stored in a pair of AA batteries.

On the other end of the spectrum of in situ nodes lie 32-bit processor-based devices such as Stargate, which run traditional operating systems such as Linux that are equipped with longer-range radios such as IEEE 802.11 or with cell phone modems.

This node class will play a critical role in most deployed systems. At a minimum, these nodes will act as gateway points both to retrieve data off the sensor network and to monitor, configure, and task the system. In more sophisticated heterogeneous systems, these nodes will be distributed more widely and used as points of aggregation, data storage, data fusion, and hosts for higher-end sensors. Because they are so much more energy intensive, these nodes operate along with a large battery and some form of recharge such as a solar panel. Alternatively, when wall outlets are available, the nodes can draw power from them.

SELF-ORGANIZED NETWORKS

Wireless communication and instrumentation have long been associated with remote sensing from satellites and missile telemetry—prime examples of wireless links. A network consists of many nodes, each with multiple links connecting to other nodes. Information moves hop by hop along a route from the point of production to the point of use.

In a wired network like the Internet, each router connects to a specific set of other routers, forming a routing graph. In WSNs, each node has a radio that provides a set of communication links to nearby nodes. By exchanging information, nodes can discover their neighbors and perform a distributed algorithm to determine how to route data according to the application's needs. Although physical placement primarily determines connectivity, variables such as obstructions, interference, environmental factors, antenna orientation, and mobility make determining connectivity a priori difficult. Instead, the network discovers and adapts to whatever connectivity is present.

Connectivity

The networking capability of WSNs is built up in layers. The lowest layer controls the physical radio device. Radios are by nature a broadcast medium: When one node transmits, a collection of others can receive the signal unless it is garbled by other transmissions at the same time. To avoid contending for the radio channel, the link layer listens on the channel and transmits only when the channel is clear. It transmits a structured series of bits that form a packet encoded in the radio signal.

When not transmitting, nodes sample the channel and scan for a special symbol at the start of a packet that also lets the receiver align itself with the sender's time. The packet layer manages buffers, schedules packets onto the radio, detects or even corrects errors, handles packet losses, and dispatches packets to system or application components.

Dissemination and data collection

Developers use this basic communication capacity to implement protocols that let the collection of nodes transport and process information and coordinate their activities. A basic capability in such networks involves disseminating information over many nodes. This can be achieved by a flooding protocol in which a root node broadcasts a packet with some identifying information. Receiving nodes retransmit the packet so that more distant nodes can receive it. However, a node can receive different versions of the same message from several neighboring nodes, so the network uses the identifying information to detect and suppress duplicates. Flooding protocols use various techniques to avoid contention and minimize redundant transmissions.

The network uses dissemination to issue commands, convey alarms, and configure and task the network. But it also uses dissemination to establish routes. Each packet identifies the transmitter and its distance from the root. To form a distributed tree, nodes record the identity of a node closer to the root. The network can use this reverse communication tree for data collection by routing data back to the root or for data aggregation by processing data at each level of the tree.

The root can be a gateway to a more powerful network or an aggregation point within the sensor network, as determined by some higher-level task. Often, tree formation and data collection are interwoven. Data can begin following up the tree as soon as a parent node is discovered. Nodes may learn of potential parents by overhearing data messages. The network continually collects statistics to reinforce the best routes.

These communication patterns differ significantly from those found on the Internet, where many client computers open connections to named servers and transfer large streams of data back and forth. In sensor networks, communication is usually performed in the aggregate, and participants are identified by attributes such as physical location or sensor value range. This style of routing has been formulated as *directed diffusion*, a process in which nodes express interest in data by attribute. The nodes flood interest outward to form a routing gradient, and they collect data up the gradient by reinforcing the associated subtree.

Reliability also follows a different pattern. Increasingly, sensor networks will deploy *disruption-tolerant networking* approaches in which they transfer bundles of data reliably, hop by hop, in contrast to the Internet, which sets up an end-to-end connection using byte or packet matching between the original source and destination to determine reliability. The DTN model better suits the variable connectivity that results from dynamic environments and the need to duty-cycle.

In sensor networks, disruption-tolerant networking reliably transfers data bundles hop by hop.

CONSERVING POWER AND BANDWIDTH

Communication, usually the most energy-intensive operation a node performs, must contend for a share of limited bandwidth. The network stack attempts to minimize energy usage, either by eliminating communication or by turning off the radio when no communication needs to occur.

Several approaches are possible. For example, nodes could process data locally and only communicate when they detect an interesting event. This approach would be employed in an intelligent alarm system or an environmental monitoring system that focuses data collection on time or areas of interest.

In many cases, crude low-power sensors trigger higher-powered sensor devices, such as cameras. Performing aggregation within the network can reduce communication. For example, an application might need to determine the average temperature at shaded nodes in a certain geographic region. Selecting the subset of readings of interest could be performed at the tree's leaves, routing their aggregation as data upward, so that each node transmits at most a single packet to provide a statistical summary of its subtree. More sophisticated aggregation could involve detecting distributed regions of interest.

Compression and scheduling also can conserve energy at lower layers. Some protocol overhead is associated with data communication to maintain routing structures, manage contention, and enhance reliability. Sensor networks can avoid explicit protocol messages by piggybacking control information on data messages and by overhearing packets destined for other nodes. They can use prescheduled time to reduce contention and the time the radio remains live. This can be coordinated with the high-level application behavior by, for example, periodic low-rate data sampling. Alternatively, the network could implement energy conservation generically within lower layers by, for example, time division multiple access.

In the spatial dimension, the network can assign specific responsibilities to certain nodes, such as retransmission or aggregation. Finally, the network can reject uninteresting packets by turning off the radio after receiving only a portion. However, because these many optimizations can be mutually

In settings in which general human activity occurs, many potentially interested parties can have varying uses for the data.

conflicting, a rich and growing body of literature employs different combinations of techniques under different application and platform assumptions.

PRIVACY

Dense instrumentation, real-time access, and in-network processing make a qualitative difference in our ability to perceive what is happening throughout large physical structures. In environmental monitoring and condition-based maintenance, the purpose of data collection, the parties responsible for using the data, and the scope of dissemination are clear. The situation becomes much less clear in more casual settings in which more general human activity occurs, such as the home, the workplace, a transportation terminal, or a shopping venue. In these cases, many potentially interested parties can have varying uses for the data.

More detailed sensing—such as occupancy, motion, and even physiological state—further amplifies concerns over proper use and dissemination. Indeed, image data, such as that obtained by the surveillance cameras in pervasive use today, can be viewed as an extremely powerful sensor, but network access and automated analysis are limited. These social factors are an inherent concern with sensor network technology. Fortunately, this area has become an active focus of research while the technology is still in its early stages.

IN THIS ISSUE

The articles in this special issue span novel sensor network applications, embedded network technology, and systems design challenges.

The three projects described in "Sensor Network Applications" demonstrate that deploying sensor networks to monitor the natural environment requires an understanding of earth science combined with sensor, communications, and computer technology.

In "Environmental Sensor Networks," Kirk Martinez and his coauthors from the University of Southampton describe their GlacsWeb project focusing on ongoing research in subglacial bed deformation and discuss the challenges encountered in extracting data gathered by sensor nodes deployed in remote locations.

As a final defense against the potential detonation of a radiological dispersion device (RDD) capable of broadcasting nonfissile but highly radioactive particles over a densely populated area, select traffic choke points in the US have large portal monitoring systems to help detect vehicles transporting illicit isotopes. In "Radiation Detection with Distributed Sensor Networks," Sean M. Brennan and coauthors discuss a project that is being developed at Los Alamos National Laboratory in cooperation with the University of New Mexico to provide a hand-carried DSN system that is less visible, uses less power per detector, and is more suitable for deployment in an urban environment than a portal monitoring system.

Detecting and accurately locating snipers has long been an elusive goal of the armed forces and law enforcement agencies. In "Shooter Localization in Urban Terrain," Akos Ledeczi and coauthors describe PinPtr, a prototype system that provides a novel approach to solving this difficult problem even in a challenging environment such as complex urban terrain. In PinPtr, an ad hoc wireless network of tiny inexpensive acoustic sensors measures both the muzzle blast and the shock wave to accurately determine the shooter's location and the bullet's trajectory. The system automatically classifies the measurements and eliminates those that are erroneous or the result of multipath effects.

"WiseNET: An Ultralow-Power Wireless Sensor Network Solution" by Christian C. Enz and his colleagues from the Swiss Center for Electronics and Microtechnology describes a platform for implementing wireless sensor networks that achieve low-power operation. To optimize overall power consumption, WiseNET uses a codesign approach that exploits the relationship between media access control layer performance and radio transceiver parameters. The system combines a complex system-on-chip sensor node with a dedicated duty-cycled radio and WiseMAC, a MAC-layer protocol designed for low-duty-cycle wireless-sensor networks.

In "The Flock: Wireless Mote Networks in an Undergraduate Curriculum," Bruce Hemingway and his colleagues from the University of Washington explain how they integrated the emerging field of wireless sensor networks into an undergraduate computer engineering curriculum to provide an introduction to embedded systems development. Using TinyOS sensor motes provided a convenient platform for applying concepts taught in a program consisting of three classes that supplement a core curriculum shared with computer science majors: advanced digital design, embedded software, and a capstone design experience.

As the technology that is commercially available today becomes established enough to warrant greater investment, straightforward engineering efforts will yield complete devices with processing, storage, sensing, and communication functions that fit in much less than a cubic centimeter of space and cost just a few dollars.

Looking forward, the technology will likely evolve into a much less distinct and visible form. Instead of being housed in many small devices, these elements will likely become part of the manufacturing process for various materials and objects. These sensors will tend to operate within the ambient energy sources of their intended environment and be placed at key junctures where analysis is most critical. As this vision evolves, so will the need for fundamentally new information technology architectures, from programming languages to signal-processing algorithms.

Over the 50 years of modern computing, we have seen a new class of computer emerge about once a decade, progressing through mainframes, minicomputers, personal computers, and mobile computers. Each successive model relies upon technical advances, especially integration, to make computing available in a form factor not previously possible. Each has ushered in new uses for computer technology. Each succeeding generation is smaller, more plentiful, and more intimately associated with personal activity than the generation that preceded it.

WSNs appear to represent a new class. They follow the trends of size, number, and cost, but have a markedly different function. Rather than being devoted to personal productivity tasks, WSNs make it possible to perceive what takes place in the physical world in ways not previously possible. In addition to offering the potential to advance many scientific pursuits, they also provide a vehicle for enhancing larger forms of productivity, such as manufacturing, agriculture, construction, and transportation. ■

David Culler *is a professor of computer science at the University of California, Berkeley, and founding director of Intel Research, Berkeley. His research interests include sensor networks, parallel computer architectures, and high-performance communication. Culler received a PhD from the Massachusetts Institute of Technology. He is a Senior Member of the IEEE and a Fellow of the ACM. Contact him at culler@eecs.berkeley.edu.*

Deborah Estrin *is the director of the Center for Embedded Networked Sensing and a professor in the Department of Computer Science at the University of California, Los Angeles. Her research interests focus on developing protocols and systems architectures needed to deploy sensor networks for environmental monitoring. Estrin received a PhD in computer science from the Massachusetts Institute of Technology. She is a Fellow of the IEEE, the ACM, and the AAAS. Contact her at destrin@cs.ucla.edu.*

Mani Srivastava *is an associate professor in the Department of Electrical Engineering at the University of California, Los Angeles. His research focuses on architectures, protocols, and algorithms for networked computing, communication, and DSP systems and VLSIs, and on CAD techniques to optimize and synthesize them. Srivastava received a PhD in electrical engineering from the University of California, Berkeley. He is a member of the IEEE and the ACM. Contact him at mbs@ee.ucla.edu.*

Sensor Network Applications

Mass-produced sensors and the use of pervasive networking technology give wireless sensor networks a new kind of scope that can be applied to a wide range of uses.

Sensor networks can be used to monitor our environment, objects in that environment, and the interactions of objects with each other and their encompassing environment. Uses for these networks include environmental and habitat monitoring; structural monitoring and condition-based equipment maintenance; and disaster management and emergency response. Researchers working on three diverse projects have developed other novel applications for sensor network technology.

In "Environmental Sensor Networks," Kirk Martinez and his coauthors from the University of Southampton describe their GlacsWeb project focusing on ongoing research in subglacial bed deformation and discuss the challenges encountered in extracting data gathered by sensor nodes deployed in remote locations.

In "Radiation Detection with Distributed Sensor Networks," Sean M. Brennan and coauthors discuss a project that is being developed at Los Alamos National Laboratory in cooperation with the University of New Mexico to provide a distributed sensor network for detecting vehicles transporting radioactive isotopes that could potentially be detonated over a densely populated area.

Finally, in "Shooter Localization in Urban Terrain," Akos Ledeczi and coauthors describe PinPtr, a prototype system that provides a novel approach for detecting and locating a sniper in a challenging environment such as complex urban terrain.

By no means exhaustive, the three projects described here demonstrate the promise of sensor network technology for monitoring our environment and our safety in that environment.

—*Computer*

Environmental Sensor Networks

Kirk Martinez, Jane K. Hart, and Royan Ong
University of Southampton

Recent developments in wireless network technology and miniaturization now make it possible to realistically monitor the natural environment. These systems can provide new data for environmental science, such as climate models, as well as vital hazard warnings such as flood alerts. This capability particularly benefits research in remote or dangerous locations, where many fundamental processes have rarely been studied due to inaccessibility.

Sensor networks incorporate technologies from three different research areas: sensing, communication, and computing.[1] Within the field of environmental sensor networks, domain knowledge is an essential fourth component. Before designing and installing any system, it is necessary to understand its physical environment and deployment in detail. The system must be able to withstand specific conditions such as temperature, pressure, or vibration. In addition, the collection and interpretation

Published by the IEEE Computer Society

of data can dramatically affect the design of communications and security mechanisms.

Sensor networks are designed to transmit data from an array of sensors to a server data repository.[2] They do not necessarily use a simple one-way data stream over a communications network; rather, elements of the system decide what data to pass on, using local area summaries and filtering to minimize power use while maximizing information content.

GENERIC ARCHITECTURE

Figure 1 shows a simple environmental sensor network. Sensor nodes gather data autonomously, and the network passes this data to one or more base stations, which forward the data to a sensor network server (SNS). We envision such servers making available a wide range of data from different networks in the future. A satellite image or map may be required to help interpret or visualize raw data. Web publishing through Web services and other interfaces gives researchers seamless access to information.

Moving from sensor nodes to the SNS, the systems generally increase in compute power, data storage, and power availability. Sensor nodes may only have power to survive a few months or years, and even base stations will have a limited life due to power supply restrictions and harsh environmental conditions. Highly mobile sensor nodes or base stations may require location-positioning systems. If a base station supports a large number of sensor nodes, they would typically be organized as an ad hoc set of clusters with representative nodes communicating a group's data to the base.

SENSOR NETWORK EVOLUTION

Environmental monitoring has a long history that includes analog loggers such as early paper plotters that measure barometric pressure and record specific environmental parameters. Loggers record data at specific intervals and require manual downloading by a maintenance team.

A number of pervasive sensor networks in use today evolved from simple data-logging systems. For example, the Argus video system for near-shore monitoring evolved from passive video recording in 1992 to active intelligent processing of images.[3] Data reduction within the sensor network prevents video data from overloading the communications infrastructure.

In addition, the automated monitoring of soil-water sensors was a forerunner of current small-scale habitat-modeling sensors.[4] These include

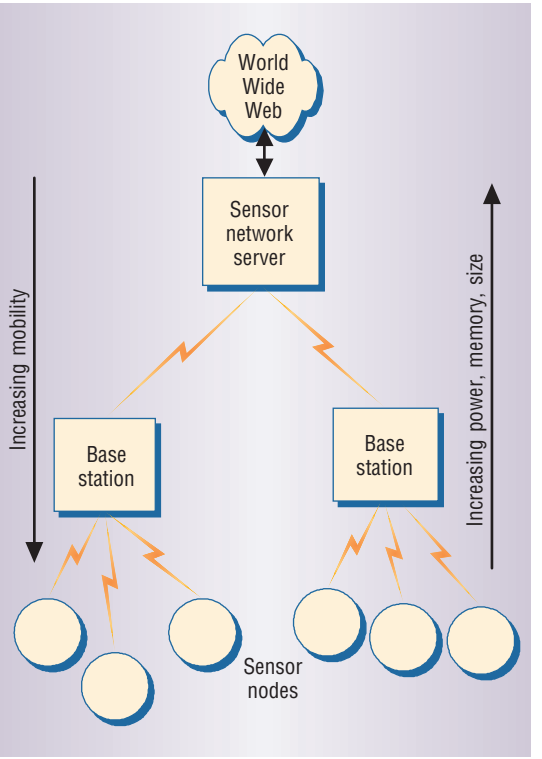


Figure 1. Generic environmental sensor network architecture. Sensor nodes gather data autonomously, and the network passes this data to one or more base stations, which forward the data to a sensor network server.

systems developed by UCLA's Center for Embedded Network Sensing that measure the population of birds and other species (http://zdnet.com.com/2100-1105-976377.html), NASA's Huntington Botanical Gardens project (http://sensorwebs.jpl.nasa.gov/resources/huntington_sw31.shtml), and UC Berkeley's habitat modeling at Great Duck Island, Maine.[5]

On a much larger scale, the environmental observations and forecasting system combines real-time, in situ monitoring with distribution networks that carry data to central processing sites. One pilot EOFS project is studying Oregon's Columbia River estuary (www.ccalmr.ogi.edu/CORIE/about.html). The University of Southampton's Envisense Center (http://envisense.org/) hosts a number of EOFS projects, including FloodNet, an intelligent sensor network designed to provide more accurate flood warnings, and self-organizing collegiate sensor (Secoas) networks to monitor coastal erosion around small islands intended as wind farms.

The next logical step for environmental sensor networks is to expand monitoring to more remote or hostile locations. NASA maintains sensor Webs in New Mexico deserts and in Antarctica[6] and plans to use intelligent sensor networks on both Mars and Jupiter's moon, Europa (http://sensorwebs.jpl.nasa.gov/).

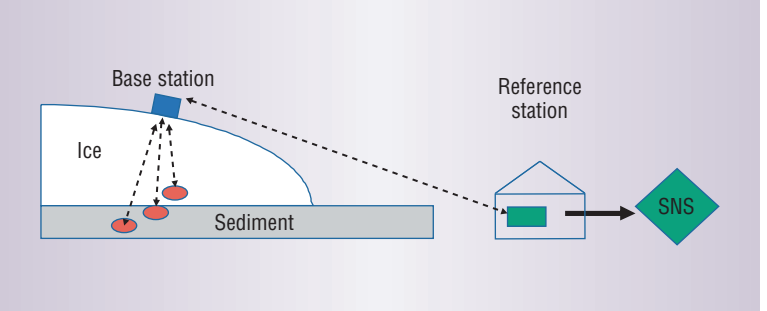


Figure 2. GlacsWeb overview. The system consists of nine probes inserted in the glacier, a base station on the glacier surface, and a reference station that relays data to the sensor network server (SNS) in Southampton, England.

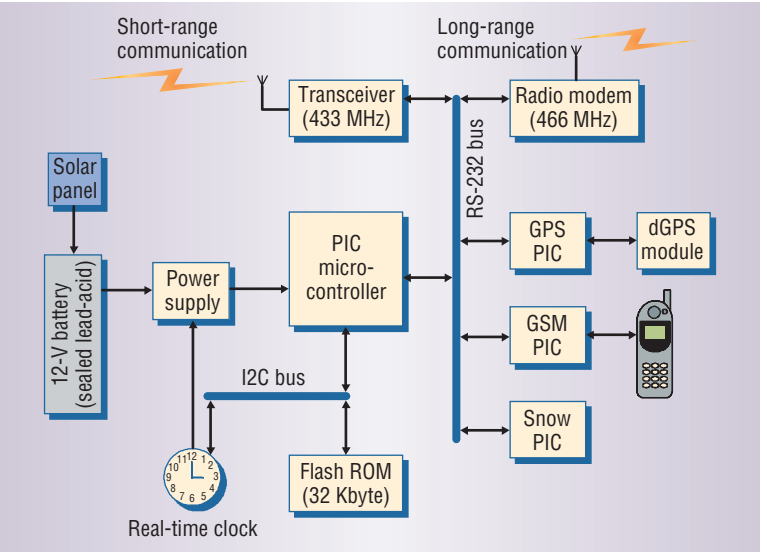


Figure 3. GlacsWeb base station system diagram. The base station serves as a communication relay between the probes and the reference station and as the controller for autonomous operation. A peripheral interface controller manages the base station, which uses other PICs to interface to some modules.

Table 1. GlacsWeb system timeline.

Time	Probe	Base station	Reference station
00:00	Data log	—	—
03:00	—	GPS log	GPS log
04:00	Data log	—	—
08:00	Data log	—	—
12:00	Data log	—	—
16:00	Communication	Communication	Communication
19:00	—	—	Transfer to SNS
20:00	Data log	—	—

GLACSWEB MONITORING SYSTEM

The Envisense GlacsWeb project is developing a monitoring system for a glacial environment that will be transferable to other remote locales both on Earth and in space.

Monitoring ice caps and glaciers provides valuable information about global warming and climate change. GlacsWeb focuses on the ongoing research area of subglacial bed deformation—how the ground on which glaciers rest affects their movement.[7,8] To accurately study this environment, the system must autonomously record glaciers over a reasonable geographic area and a relatively long time. It also must be as noninvasive as possible to let the sensor nodes, or probes, mimic the movement of stones and sediment under the ice.

System overview

The GlacsWeb system, shown in Figure 2, consists of probes inserted in the glacier, a base station on the glacier surface, and a reference station that relays data to the SNS in Southampton, England. Most of the nine probes, deployed in 2003, are at the ice-sediment boundary from 50 to 80 meters deep. Each probe is equipped with pressure, temperature, and orientation (tilt in three dimensions) sensors. The probes are not recoverable.

As Figure 3 shows, the base station serves as a communication relay between the probes and the reference station and as the controller for autonomous operation. It includes temperature and tilt sensors, a snow meter, a webcam, and a differential Global Positioning System (dGPS) to follow ice movement. In the 2003 version, PICs are used exclusively for all base station functions.

The reference station is a mains-powered, Linux-based gateway for transferring data. It acts as the position reference point and records a dGPS file daily. This PC relays the data from the probes, base station, and dGPS to the data server in Southampton on a daily basis via ISDN.

To survive for one year, most of the system uses a real-time clock (RTC) to power off between readings. The power budget allows the probes to wake up every four hours to take readings. However, the communication channel is only opened once a day during a system-wide window. Table 1 shows the daily sequence of events. At the end of each period, the probe and base station configure their RTCs to the next wake-up time before shutting down.

Sensor nodes

GlacsWeb sensor nodes must be

- low cost,
- low power for long-term operation,
- automated (maintenance free),
- robust to withstand errors and failures,
- nonintrusive to minimize environmental disturbance, and
- low polluting.

The electronics and sensors are enclosed in sealed plastic cylindrical capsules 10 centimeters long. As Figure 4 shows, each of these probes has one 100-psi pressure sensor, two dual-axis microelectromechanical-system tilt sensors, and a temperature sensor. To enable failsafe control and remote reprogramming, two PIC microcontrollers read sensor data and store it in flash ROM. The PICs can receive and interpret commands including reprogramming and time-schedule changes.

Probes communicate with their base station via a transceiver with an omnidirectional antenna. In case of communication interruptions—which are especially common during the wet summer deployment period—a 16-Kbyte ring buffer functions as a long-term data-storage mechanism that enables the probe to remain in isolation for up to six months. In standby, the probes only consume 10µA. To withstand low temperatures and to maximize lifespan, the probes are powered by six 3.6-V, 1-Ah lithium/thionyl chloride 1/2 AA batteries. The RTC controls the efficient switch-mode regulator, which can tolerate a voltage drop from the batteries.

Communications

The glacial environment imposes certain communications demands, including:

- omnidirectional high power for the probes,
- long-range communication between base stations and the reference station,
- a low data rate,
- error detection and correction, and
- backup channels.

Because the communication chain can fail at any point, GlacsWeb relies on a store-and-forward data-transfer mechanism. Base stations use a ring-buffer technique similar to the probes to ensure that data flows when communication channels are available. The system links a long-range (2.5-km) hop between the base stations and reference station with a 500-mW, 466-MHz radio modem with built-in error handling—9.6 Kbps suffices. If this fails, the base station uses a backup Global System for Mobile communications phone to send data directly to the SNS. This actually occurred when one radio modem failed in the reference station.

Inclusion of PIC microcontrollers ruled out the use of TCP/IP. Instead, we devised a custom packet-based protocol with error detection that allowed lower overhead and a greater degree of experimentation using a multimaster bus-like network topology. The system's extensive use of store-and-forward, time-outs, checksums, and retries makes it resilient to communication errors—for example, broadcast packets allow system-time synchronization.

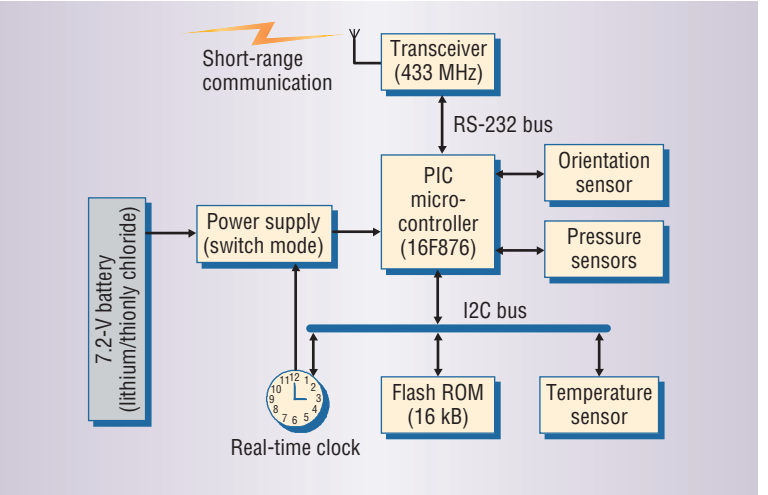


Figure 4. GlacsWeb probe system diagram. Each probe is designed to withstand severely cold temperatures and long communication interruptions.

Computing

To create a sensor network, GlacsWeb must incorporate a range of different computer systems and software:

- microcontrollers for sensor nodes,
- a small operating system for sensor nodes,
- low-power systems for base stations,
- routing and message-passing capabilities,
- a server for the SNS, and
- publishing software for visualization and services.

Simple microcontrollers dictate using nonstandard operating systems. Given the small memory available on a PIC, we opted not to use an OS. Many wireless embedded sensor networks employ systems such as TinyOS, which would be less buggy and easier to maintain, but the limited storage capacity of these systems makes it difficult to implement complex algorithms.

Base stations must be able to communicate with many systems using both custom and standard devices. A low-power board with suspend mode and a real operating system are ideal for development purposes. PICs are not suitable for standard Wi-Fi networking, but they do enable using some software in place of hardware—for example, the RS-232 bus for routing. We are currently using a low-power ARM-based board running Linux.

A Xeon-based Linux server is the final destination for the data and all Web publication. A key element of GlacsWeb is the ability to publish Web services and data descriptions so that future search engines can gather the data.

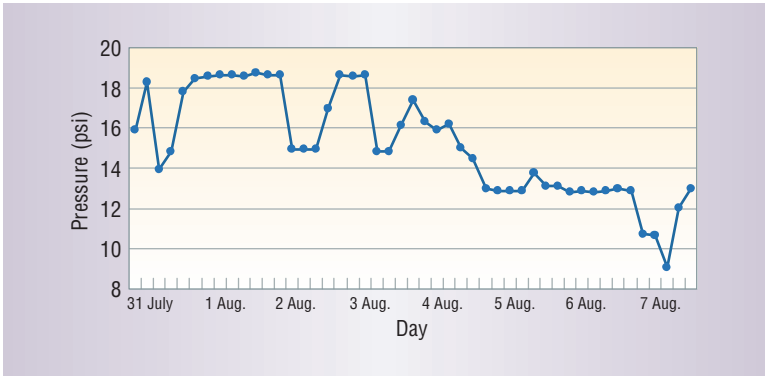


Figure 5. Probe 8 pressure readings, 2003. The probe likely stopped transmitting data after 7 August because it slipped into the water-filled zone at the bottom of the hole.

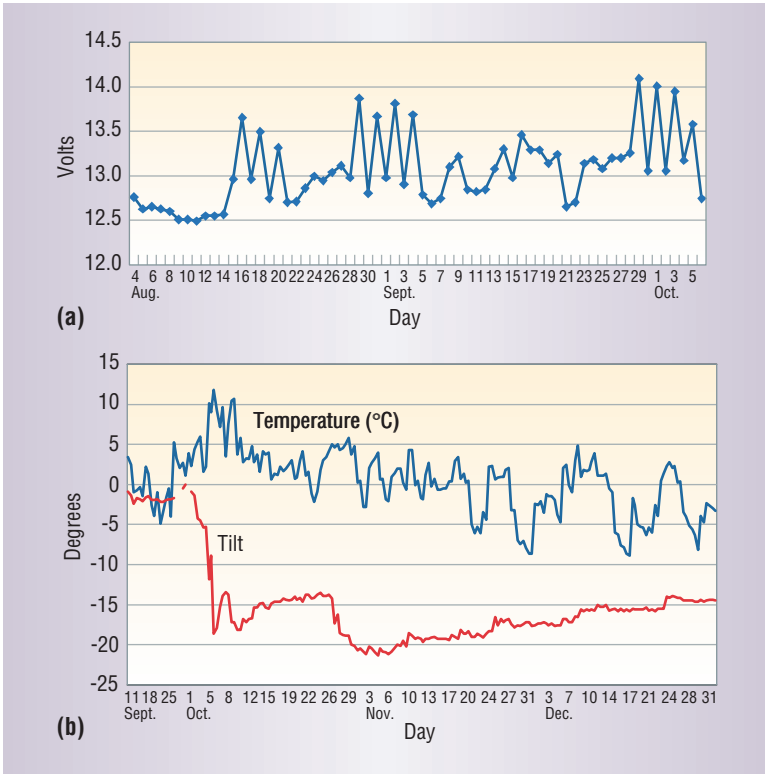


Figure 6. Base station results, 2003. (a) Battery voltage increased during the summer due to the 10-watt solar panel mounted on top. (b) The base station moved during warm periods but was reasonably stable on a 15° slope.

PRELIMINARY RESULTS

We installed a prototype system at Briksdalsbreen, Norway, in August 2003. After conducting a ground-penetrating radar survey to map out rivers in and under the ice, we drilled holes with a modified high-pressure, hot jet wash. The base station used a tripod sitting on the ice to hold the antennas, in case snow built up during the winter.

Figure 5 shows the pressure readings received from Probe 8 for seven days after deployment in the glacier. The temperature was a constant 0.8° C; the tilt readings (not shown) were also constant throughout this period. Probe 8 was tightly wedged approximately 20 meters into the glacier. The probe likely stopped transmitting after 7 August because it slipped into the water-filled zone at the bottom of the hole. Under such circumstances, communication is impossible until the water freezes in the winter, when we expect each probe to transmit its data backlog.

Figure 6 illustrates battery voltage, tilt, and temperature results obtained from the base station. As Figure 6a shows, the battery voltage level fluctuated between 12.5 V and roughly 13.5 V over the course of 65 days. Overall, battery charge actually increased during the summer due to the 10-W solar panel mounted atop the base station.

The tilt sensor (red) and temperature (blue) readings in Figure 6b indicate that the base station moved during warm periods but was reasonably stable on a 15° slope. The glacier's temperature slowly decreased during the onset of winter yet remained within the components' operating range. We anchored the equipment using rocks and a submerged pole as a backup. The dGPS readings indicate that the ice surface melted down around 2 meters over the summer, confirming that drilled poles cannot function as anchors on a glacier.

These preliminary results confirm the soundness of the GlacsWeb system architecture. Future work includes designing a position-measuring system to locate the probes and further miniaturizing their electronics. The second version of the system will be installed in the summer of 2004, building on our experiences and adding new sensors.

SENSOR NETWORK CHALLENGES

Extracting data gathered by sensor nodes in remote locations involves some unique challenges. The GlacsWeb project has tackled many of these issues, contributing to an increased understanding of the solutions.

Miniaturization

Because sensor networks often are deployed in confined spaces, miniaturization can ensure that they are unobtrusive. Antenna size can be a limitation for low-frequency radios. We used dielectric antennas measuring only $5 \times 7 \times 0.5$ mm to save space as well as for their other properties. In addition, subsystem miniaturization must be balanced with battery size and radio power requirements. We used surface-mounted components together with double-sided boards, but further integration could be achieved via other means, including using programmable logic.

Power management

Long-term operation of sensor networks such as GlacsWeb requires economic power management. In common with other projects, we used a time schedule to minimize power consumption and employed high-efficiency, regulated switch-mode power supplies. Concerns about lengthy communication delays and reliability made it risky to initially use an extremely adaptive power management scheme. However, a rate-of-change-driven data-capture system and interprobe ad hoc communications would, in theory, further reduce power use. Systems requiring a long boot or resume time must be avoided, as power savings can become the dominant factor.

Radio communications

Wet and windy locations often hinder reliable radio communications. We found that theoretical calculations of radio losses in glacier ice were a poor guide to actual performance, and this is likely to be true for unpredictable factors in other environments—for example, leaf-cover changes in a forest habitat. The ability to alter transceiver power and the use of lower frequency or acoustic fallback systems will be common in the future. Communications between sensor nodes is useful where some may be out of their base station's range, but energy used to set up such ad hoc networks and clusters must be minimized.

Scalability

It is necessary to regularly add groups of sensors as well as manage potentially large numbers of systems. Our initial network topology allowed up to 256 unique devices to connect to one base station. A communications window could limit scalability because the time could be insufficient for many probes to send a backlog of data. However, in our case the probes can simply send more data the next day or receive commands to remain awake longer. Arrays of base stations or gateways will typically be needed to increase scalability.

Remote management

Because researchers cannot regularly visit systems in isolated locations, remote access is necessary to fix bugs, shut down subsystems, change schedules, and so on. In our case, we added a webcam to the base station to monitor the physical status of the site and systems. Power control to completely isolate subsystems was also essential for power management as well as workarounds—for example, as the Duck Island project demonstrated, water can cause short circuits in sensors. Custom communications complicate remote access because normal logins and routing are unavailable. More software development and failure scenario testing is necessary to achieve good remote management.

Usability

To be practical, environmental sensor networks must primarily consist of off-the-shelf components that are relatively easy to deploy, maintain, and understand—much like the devices in a wireless home network. The average earth scientist could not install the GlacsWeb system because it requires a wide range of computer and electronics technologies with complex interfaces. Plug-and-play-style developments will help in this area.[9] In addition, researchers must be able to effectively access large volumes of data gathered by network sensors. The BeanWatcher[10] is one promising tool that addresses semiautomatic monitoring and management.

To be practical, environmental sensor networks must primarily consist of off-the-shelf components.

Standardization

Compatibility between off-the-shelf modules such as dGPS units and weather stations in environmental sensor networks is very low, and in practice every integrated module requires separate code. In some cases, drivers are available—for a webcam, for example—but without the correct operating system, the devices are unusable. A future challenge will be to standardize interfaces and even implement some radio networking to allow interoperability among products from different sensor network vendors.[11]

The research community must agree on some common ontologies to describe the domains, then standardize publication of final data. This could be accomplished by connecting environmental sensor networks to the Semantic Web (www.semanticweb.org). The GEOscience Network (www.geongrid.org/research.html) and IrisNet[12] represent a move in this direction.

Security

All sensor network levels must take security into account. Systems must blend into the environment and, when appropriate, carry warnings, alarms, and other information. Some networks can cope with the loss of one or more nodes due to failure or damage, and in remote areas, physical security may not be an issue. In addition, data may need protection against deliberate or accidental alteration. However, security mechanisms should not

hamper public access to information. Striking a balance between security and accessibility helps ensure that all parties can trust the systems.

Environmental sensor networks offer an exciting research opportunity. They make it possible, for the first time, to merge data sets of different types and scales to enhance our understanding of the Earth. However, designing sustainable sensor networks for the natural environment is a demanding task. Communications engineering, power management, deployment, weatherproofing, stability, and remote diagnostics all present difficult technical challenges. In the long term, integrating environmental sensor networks with the Semantic Web could make useful information available globally. ■

Acknowledgments

We thank Intellisys, BTExact, and Topcon for their financial support; Sue Way, Daniel Miles, David Robertson, and Al Ridoch for field assistance; and Inge and Gro Melkevoll for use of their café to locate the reference station.

References

1. C-Y. Chong and S.P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proc. IEEE,* vol. 91, no. 8, 2003, pp. 1247-1256.
2. I.F. Akyildiz et al., "A Survey on Sensor Networks," *IEEE Comm.,* vol. 40, no. 8, 2002, pp. 102-114.
3. R. Holman, J. Stanley, and T. Özkan-Haller, "Applying Video Sensor Networks to Nearshore Environment Monitoring," *IEEE Pervasive Computing,* vol. 2, no. 4, 2003, pp. 14-21.
4. M.E. Grismer, "Field Sensor Networks and Automated Monitoring of Soil-Water Sensors," *Soil Science,* vol. 154, no. 6, 1992, pp. 482-489.
5. R. Szewczyk et al., "Lessons from a Sensor Network Expedition," *Proc. 1st European Workshop Wireless Sensor Networks* (EWSN 04), LNCS 2920, Springer, 2004, pp. 307-322.
6. K.A. Delin et al., "Sensor Web in Antarctica: Developing an Intelligent, Autonomous Platform for Locating Biological Flourishes in Cryogenic Environments," *Proc. 34th Lunar and Planetary Science Conf.* (LPSC 03), 2003; http://sensorwebs.jpl.nasa.gov/resources/Delin_et_al_Antarctica_2003.pdf.
7. R.B. Alley, "Continuity Comes First: Recent Progress in Understanding Subglacial Deformation," A.J. Maltman, B. Hubbard, and M.J. Hambrey, eds., *Deformation of Glacial Materials,* Geological Soc. of London, 2000, pp. 171-180.
8. J. Hart and J. Rose, "Approaches to the Study of Glacier Bed Deformation," *Quaternary Int'l,* vol. 86, no. 1, 2001, pp. 45-58.
9. Y. Gsottberger et al., "Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments," *Proc. 1st European Workshop Wireless Sensor Networks* (EWSN 04), LNCS 2920, Springer, 2004, pp. 291-306.
10. A. Lins et al., "BeanWatcher: A Tool to Generate Multimedia Monitoring Applications for Wireless Sensor Networks," *Proc. 6th IFIP/IEEE Int'l Conf. Management of Multimedia Networks and Services* (MNMS 03), LNCS 2839, Springer, 2003, pp. 128-141.
11. F.C. Delicato et al., "A Flexible Middleware System for Wireless Sensor Networks," *Proc. 2003 ACM/IFIP/Usenix Int'l Middleware Conf.,* LNCS 2672, Springer, 2003, pp. 474-492.
12. P.B. Gibbons et al., "IrisNet: An Architecture for a Worldwide Sensor Web," *IEEE Pervasive Computing,* vol. 2, no 4, 2003, pp. 22-33.

Kirk Martinez *is a senior lecturer in the School of Electronics and Computer Science at the University of Southampton. His research interests include sensor networks, image processing, and the Semantic Web. Martinez received a PhD in electronics systems engineering from the University of Essex. He is a member of the IEEE, the IEEE Computer Society and the American Geophysical Union. Contact him at km@ecs.soton.ac.uk.*

Jane K. Hart *is a Reader in the School of Geography at the University of Southampton. Her research interests include glaciology, glacial sedimentology, and climate change. Hart received a PhD in environmental science from the University of East Anglia. She is a member of the International Glaciological Society and the American Geophysical Union. Contact her at jhart@soton.ac.uk.*

Royan Ong *is a research assistant in the School of Electronics and Computer Science at the University of Southampton. His research interests include sensor networks and control systems. Ong received a PhD in electronics systems engineering from the University of Leicester. He is a member of the IEE. Contact him at hlro1@ecs.soton.ac.uk.*

Radiation Detection with Distributed Sensor Networks

Sean M. Brennan, Angela M. Mielke, and David C. Torney
Los Alamos National Laboratory

Arthur B. Maccabe
University of New Mexico

In any assessment of potential terrorist attacks, the nuclear threat takes center stage. Although weapons-grade nuclear materials are heavily guarded, a plausible scenario involves terrorists detonating a simple radiological dispersion device (RDD) capable of broadcasting nonfissile but highly radioactive particles over a densely populated area. In most cases, a motor vehicle would have to transport the device and its payload—commonly known as a "dirty bomb"—to the target destination. As a final defense against such a weapon, select traffic choke points in the US have large portal monitoring systems to help detect illicit isotopes.

The Distributed Sensor Network project at Los Alamos National Laboratory, in cooperation with the University of New Mexico, is developing a network of radiation detectors that, coupled with other sensors that collect supportive data, is suitable for RDD interdiction in either urban or rural environments. Compared to a portal monitor, a DSN is much less visible, uses less power per detector, is hand carried and thus more rapidly deployable, and simplifies coverage of multiple transport avenues. Also, to function effectively, portal monitoring systems typically require slow or halted traffic, whereas our DSN can be tailored for any moderate traffic speed.

HARDWARE

Expanding on earlier work suggesting that bigger is not necessarily better in radiation detectors,[1,2] our project seeks to provide a flexible, discreet radiation detection solution that enhances not just national security but also global nonproliferation.

Our model DSN consists of arrays of 75-mm sodium iodide (NaI) scintillators directly connected to PDA-sized platforms that provide in situ processing of raw gamma counts. In situ processing eliminates a single point of failure and can potentially weed out faulty measurements. We chose PDAs because they have the processing capability to cope with more complex algorithmic requirements in the future, and a PDA is usually smaller than the radiation detection equipment to which it attaches. Also, PDAs with a general-purpose OS—in this case, Linux—can use familiar and well-tested software tools to manipulate and communicate data.

On each side of a typical two-lane road, 6 to 7 meters wide, we deploy an independently operating array of detectors. The detectors are several meters apart and well away from the roadway, so that the two arrays are approximately 10 meters from each other. In the forward and rear positions and interspersed among the radiation detectors are simpler nodes, such as Crossbow's MICA2 mote, that use accelerometers, magnetometers, and similar sensors to directly detect and track vehicles through the DSN operational space.

METHODOLOGY

To compensate for the smaller detectors' reduced efficiency and source interaction time, the system combines gamma counts across the detection array and coordinates this data with the radioactive source's motion. This *coherent signal addition* method uses an integration window that follows the source as it moves past each detector in the array. During algorithm development,[3] we discovered that when this window length matches detector spacing and expected time lag, increasing the number of detectors also increases the signal-to-noise ratio (SNR) along a $\sqrt{n}$ curve.

In the absence of traffic, the system collects background radiation measurements, compiles these as a mean and standard deviation to produce a baseline threshold, and updates its statistical noise model. When a vehicle approaches the DSN, the forward sensors cue the detection apparatus. These forward units typically detect either seismic vibrations (using an accelerometer) or variations in the local magnetic field (using a magnetometer) and broadcast a time-stamped report when the source exceeds threshold values. This chatter rapidly propagates to the first radiation detector, which informs its neighbors of an oncoming target.

The motes use a separate radio frequency from the PDAs, but one or more PDAs can listen in via an attached mote gateway. Using the mote time stamp, the first radiation detector records gamma counts while the vehicle is within a designated interaction range. For example, the detector can begin counting when the vehicle is 10 meters from its closest approach and cease when it has receded 10 meters. At this moment, the next detector begins

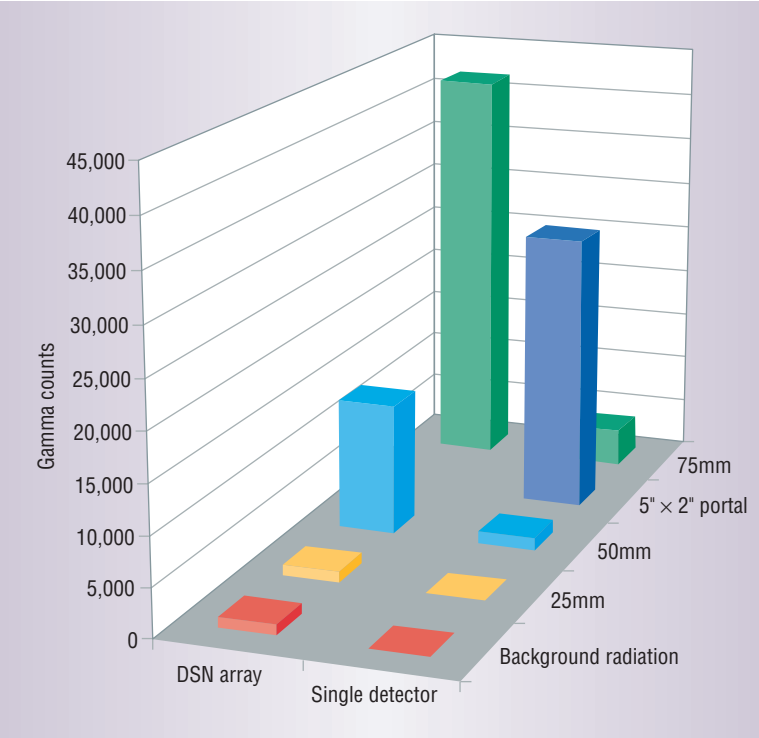


Figure 1. Detector performance. The integration time for one detector is 1 second, yielding a total of 11 seconds for an array. The portal monitor integration time is 11 seconds. Measured background count for the portal monitor is equivalent to that of the entire DSN array.

recording readings, and so on down the array.

In this way, the system passes on the gamma count across all channels minus background noise, along with local background statistics, from one node to the next. The end node adds the values, calculates a threshold from the noise statistics, and determines if the source total significantly exceeds the threshold. If so, the DSN propagates an alert to an uplink for human intervention.

Our analysis assumes that a suspect vehicle will travel at a constant speed ranging from 25 to 45 mph (11-20 m/s). However, future refinements could handle acceleration. Previous research suggests that small accelerations will not greatly impact the SNR.[3]

In practice, we limit the number of detectors to about 11 when using coherent addition. At increasingly larger scales, this algorithm's additive effect on the SNR reaches an asymptote. If our support sensors can provide accurate speed estimates, we can space the radiation detectors more widely and dynamically adjust the interaction window as appropriate for the reported speed. With slower sources, the system would act as if it had regular gaps in the array, yet it would be equally effective.

Alternatively, we could cover various constant speeds, accelerating sources, and, to a hardware-dependent extent, characterize the spectral signature of these sources using more computationally intensive Bayesian methods.

SIMULATION

To thoroughly explore the design space and test our software before field tests, we simulated the DSN along with the technically more efficient portal monitor. We assumed a radiation source composed primarily of cesium-137 due to its industrial availability, typically in powdered form as cesium chloride, and because its extremely high radioactivity would likely promote its use in an RDD. To test our detection scheme's limits, only a small, unshielded mass is transported, equivalent in detectability to a larger measure of the isotope in a lead container of significant thickness.

Our simulations also assumed a speed limit of 45 mph (20 m/s). Given this expected speed, we placed the 11 detectors at intervals of 20 meters within each array and set the detection integration time—how frequently a detector reports its gamma count—at one second.

We expected our simulations to merely show our approach's suitability for certain types of deployments. Instead, we found that the DSN promises improved performance over a single detector to the point of being comparable to some portal monitors that measure vehicles moving at much lower speeds.

We compared the performance of three NaI detector sizes—75, 50, and 25 mm—to evaluate contenders for the radiation detection component. The 75-mm detector is the most common, but the other two are significantly smaller and less expensive. We wanted to establish and quantify the additive effect of our method as well. Using the coherent addition algorithm, we collected a total gamma count across all channels exclusive of background. To bypass shielding effects, we describe the sources in terms of their radioactivity as measured in curies, in this case .01 curie.

As Figure 1 shows, the 75-mm NaI detector array performs distinctly better than one portal monitor over the same total system integration time. The 50-mm detector, also in a coherent additive array of 11 sensors, collects a total count that appears to be sufficient for our purposes because it is still significantly greater than the sampled background. For a small .01-curie source, the 25-mm detector is inadequate even with coherent addition. This demonstrates a limitation of the DSN approach: The system cannot improve resolution if the component detectors collect an insignificant signal.

An individual 75-mm detector is small enough to be convenient, yet an array of 11 outperforms a single portal monitor. Because our system uses

many detectors, it can surpass a given performance ceiling and achieve that performance for much faster sources than is typical.

Our simulation studies demonstrate that there is great potential for DSNs to play a significant role in radiation detection. Radiological DSNs can complement the portal monitor approach by enabling rapid deployment and much greater transparency to the public while achieving equal or greater performance.

While we continue to make algorithmic improvements, we are realizing our goal of using commercial off-the-shelf hardware. Our current implementation uses Crossbow MICA2 motes to detect vehicular passage with magnetometers, and Sharp Zaurus PDAs to act as communication bridges between the ISM band motes and the 802.11b PDAs. Other Zaurus PDAs are connected by serial cable to Black Cat Systems Geiger-Mueller tube radiation counters.

In this proof-of-concept implementation, our target radiation source is rather large: approximately 1 curie. This increase by two orders of magnitude is not unreasonable—actual RDDs using this isotope may very well be even larger, without accounting for shielding.

This implementation is extremely inexpensive, and any later detector upgrades will directly yield increased sensitivity. We are currently integrating and testing the component subsystems, and our results are not finalized. However, our experiments to date have validated both the theory and simulation of this DSN.

DSNs show promise not just for radiation detection and rapid response, but also for in situ and real-time detection of a multitude of dangerous phenomena. In the coming years, as sensor hardware improves, we will expand our efforts to reduce other threats such as chemical and biological weapons. ■

Acknowledgment

This work was supported by US Department of Energy/NNSA and Los Alamos National Laboratory funds under University of California contract W-7405-ENG-36 and approved for public release under LA-UR-04-0664.

References

1. M.A. Nelson and B.D. Rooney, *Modeling and Bench-Top Results of the Arrayed Vehicle Monitor*, tech. report LA-UR-03-8055, Los Alamos Nat'l Laboratory, 2003.
2. K.P. Ziock and W.H. Goldstein, "The Lost Source, Varying Backgrounds and Why Bigger May Not Be Better," *Unattended Radiation Sensor Systems for Remote Applications*, J.I. Trombka, ed., Am. Inst. Physics, 2002.
3. R.J. Nemzek et al., "Distributed Sensor Networks for Detection of Mobile Radioactive Sources," to be published in *IEEE Trans. Nuclear Science*, vol. 51, no. 4, 2004.

Sean M. Brennan *is a technical staff member at Los Alamos National Laboratory. His research interests include scalability and fault tolerance of distributed computing and sensory systems as well as practical deployment issues. Brennan received an MS in computer science from the University of New Mexico and is currently working toward a PhD. He is a member of the IEEE Computer Society. Contact him at brennan@lanl.gov.*

Angela M. Mielke *is the project leader for the NNSA-funded Distributed Sensor Networks with Collective Computation Project at Los Alamos National Laboratory. Her primary research interest is in the design and algorithm development of statistical pattern recognition techniques for data analysis and discovery. She received a BEE from Villanova University. Contact her at amielke@lanl.gov.*

David C. Torney *is a technical staff member at Los Alamos National Laboratory. He is an applied mathematician with a PhD in biophysics and mathematics from Stanford University. Contact him at dct@lanl.gov.*

Arthur B. Maccabe *is an associate professor in the Department of Computer Science and the director of the Center for High Performance Computing at the University of New Mexico. His primary research focus is on the development of runtime and operating systems for very large scale, high-performance computing systems. Maccabe received a PhD in information and computer sciences from the Georgia Institute of Technology. He is a member of the IEEE Computer Society. Contact him at maccabe@cs.unm.edu.*

Shooter Localization in Urban Terrain

Miklos Maroti, Gyula Simon, Akos Ledeczi, and Janos Sztipanovits
Vanderbilt University

Detecting and accurately locating snipers has been an elusive goal of the armed forces and law enforcement agencies for a long time. Existing countersniper systems use acoustic, visual, infrared, or electromagnetic signals related to gunfire to determine a shooter's bearing or exact location.[1] The main limiting factor in these systems is a line-of-sight requirement.

Most successful sniper-detecting systems are based on acoustic measurements. As Figure 1 shows, the observable sound originates from either the muzzle blast or the acoustic shock wave that a supersonic projectile produces.

PINPTR ACOUSTIC SYSTEM

The performance in most current acoustic systems significantly degrades when used in urban terrain because multipath effects typically corrupt the available sensor readings. However, we have developed an acoustic system that works well even in complex urban environments (www.isis.vanderbilt.edu/projects/nest/applications.html).

Funded through the Network Embedded Systems Technology program of the US Defense Advanced Research Projects Agency's Information Exploitation Office, the PinPtr system uses a wireless network of many low-cost sensors to determine both a shooter's location and the bullet's trajectory by measuring both the muzzle blast and the shock wave.[2]

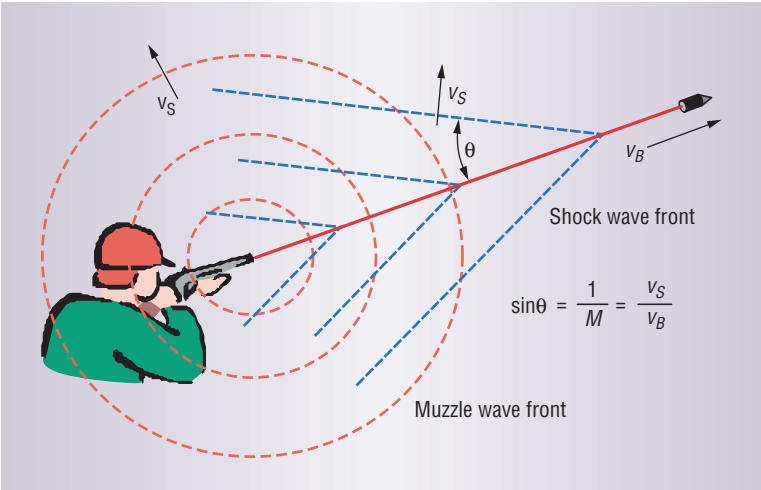


Figure 1. Acoustic events generated by a shot. The muzzle blast produces a spherical wave front that travels at the speed of sound (v_s) from the muzzle. The supersonic projectile generates a shock wave in every point of the trajectory, producing a cone-shaped wave front (assuming the speed of the projectile is constant v_B). The projectile's Mach number M determines the angle of the shockwave cone.

Similar to traditional systems, PinPtr estimates the source location based on the measured time of arrival of acoustic events, known sensor locations, and the speed of sound. However, traditional systems use time-difference-of-arrival equations, making them susceptible to multipath effects.

The PinPtr sensor-fusion algorithm, which runs on a base station, performs a search on a hypersurface defined by a consistency function. This function provides the number of sensor measurements that are consistent with hypothetical shooter positions and shot times. The algorithm automatically classifies measurements and eliminates those that result from multipath effects or are otherwise erroneous. A fast search algorithm finds the global maximum of the surface,[2] which corresponds to the shooter position.

PinPtr can use hundreds of sensors. Users can either deploy the sensors manually, placing them in predetermined locations, or drop them in a random formation by some other means. After deployment, the sensors automatically establish an ad hoc communication network, perform self-localization, and establish a common time base. The system is then ready to use.

When the system detects an event, it measures the time of arrival and uses a specially tailored data aggregation and routing service to propagate the information through the network to the base station.

PROTOTYPE IMPLEMENTATION

We built a PinPtr prototype on top of the University of California, Berkeley's MICA2 mote platform running TinyOS.[3]

While it provides an excellent low-cost hardware and software platform for sensor network applications, severe resource constraints prohibit the implementation of muzzle blast and shockwave detection on the mote itself. Therefore, we developed a custom acoustic sensor daughter card to execute the signal-detection and time-stamping algorithms on an on-board field-programmable gate array.

The mote itself runs the operating system; middleware services such as time synchronization, data

Figure 2. PinPtr interface. In an overhead shot of the test area, the red dot shows the estimated shooter position, and the red line indicates the direction of the shot. The coordinates, including the elevation (8.4 m), are displayed on the right-hand side. Colored circles and dots show the sensor locations: white—no detection; blue—detection; cyan—line-of-sight detection.

aggregation, and message routing; and application-specific routines.

We demonstrated and evaluated the prototype in a US Army test facility that offered a realistic urban environment. Figure 2 shows the 100 × 100-meter test area.

We deployed 60 sensors for the test, and the system accuracy was approximately 1 meter on average in three dimensions. This meant that PinPtr could determine the exact window from which a particular shot was fired. The system latency was less than 2 seconds.[2]

The PinPtr system and test show the value of wireless networks in bringing highly redundant sensing and distributed processing to solve a difficult problem. ■

References

1. A. Vick et al., *Aerospace Operations in Urban Environments: Exploring New Concepts*, Rand, 2000.
2. G. Simon et al., "Sensor Network-Based Countersniper System," to be published in *Proc. 2nd ACM Conf. Embedded Networked Sensor Systems* (Sensys), ACM Press, 2004.
3. J. Hill and D. Culler, "Mica: A Wireless Platform for Deeply Embedded Networks," *IEEE Micro*, vol. 22, no. 6, 2002, pp. 12-24.

Miklos Maroti *is a research assistant professor in the Electrical Engineering and Computer Science Department at Vanderbilt University. Maroti received a PhD in mathematics from Vanderbilt University. Contact him at miklos.maroti@vanderbilt.edu.*

Gyula Simon *is a research assistant professor in the Electrical Engineering and Computer Science Department at Vanderbilt University. Simon received a PhD in electrical engineering from the Technical University, Budapest. Contact him at gyula.simon@vanderbilt.edu.*

Akos Ledeczi *is a research assistant professor in the Electrical Engineering and Computer Science Department and a senior research scientist in the Institute for Software Integrated Systems at Vanderbilt University. Ledeczi received a PhD in electrical engineering from Vanderbilt University. Contact him at akos.ledeczi@vanderbilt.edu.*

Janos Sztipanovits *is an E. Bronson Ingram distinguished professor in the Department of Electrical and Computer Engineering and director of the Institute for Software Integrated Systems at Vanderbilt University. Sztipanovits received a PhD in electrical engineering from the Technical University, Budapest. Contact him at janos.sztipanovits@vanderbilt.edu.*

WiseNET: An Ultralow-Power Wireless Sensor Network Solution

The WiseNET platform uses a codesign approach that combines a dedicated duty-cycled radio with WiseMAC, a low-power media access control protocol, and a complex system-on-chip sensor node to exploit the intimate relationship between MAC-layer performance and radio transceiver parameters.

Christian C. Enz
Amre El-Hoiydi
Jean-Dominique Decotignie
Vincent Peiris
Swiss Center for Electronics and Microtechnology

A wireless sensor network consists of many energy-autonomous microsensors distributed throughout an area of interest. Each node monitors its local environment, locally processing and storing the collected data so that other nodes can use it. Network nodes share this information via a wireless link. Using data fusion, specific features of interest to the end user can be extracted from the information that several nodes collect while a multihop communication scheme propagates this information to a base station node.[1-4]

Since these networks often are deployed in regions that are difficult to access, the nodes should not require maintenance. They must be energetically autonomous, using batteries that do not need to be replaced or recharged. In many application scenarios, the targeted node lifetime typically ranges from two to five years, imposing drastic constraints on power consumption. With a single 1.5-V AA alkaline battery, the average power consumption ranges from 100 to 10 microwatts, for a node lifetime ranging between two and seven years. Given that today's commercially available radio transceivers typically consume several tens of milliwatts, keeping the transceiver constantly active is clearly impossible. Maintaining the required power consumption requires having the nodes sleep most of the time. This can be achieved by introducing duty cycling on the order of 0.1 percent to 1 percent, while keeping a low sleep-mode current no larger than the battery leakage current.

Reducing power consumption requires optimization across all layers, from the physical layer, channel coding, and media access control layer up through the routing, transport, and application layers. The MAC layer plays the most crucial role in the communication protocols' overall energy efficiency, especially for networks with low-duty-cycling radios.

To optimize power consumption, the Swiss Center for Electronics and Microtechnology has developed WiseNET, an ultralow-power platform for the implementation of wireless sensor networks that achieves low-power operation through a careful codesign approach. WiseNET combines a dedicated duty-cycled radio with WiseMAC, a low-power MAC protocol designed for low-duty-cycle wireless sensor networks.[5] The WiseNET solution consumes about 100 times less power than comparable solutions available today.

WISEMAC PROTOCOL

At the outset, we analyzed WiseMAC's operation to identify the most important radio parameters

Published by the IEEE Computer Society

affecting power consumption. Given that available radio transceivers did not meet expectations and could not be modified, we designed a dedicated WiseNET radio transceiver that optimized these parameters for WiseMAC. This codesign approach—which optimizes overall power consumption by exploiting the intimate relationship between MAC layer performance and the radio transceiver parameters—is the heart of the WiseNET solution.

Typically, a microsensor node performs several functions, including

- sensing environmental physical parameters,
- processing the raw data locally to extract characteristic features of interest,
- storing this information momentarily, and
- using a wireless link to transmit the information to its neighbors.

The node must also operate as a relay for implementing multihop communication by receiving the data coming from one or several of its neighbors and then processing it before routing it to the next neighbor toward the destination.

To perform these functions, a sensor node—which includes many subsystems—can be integrated into a single system on chip to minimize power consumption and reduce the cost. Therefore, in addition to the radio optimized for the WiseMAC protocol, we developed a complete sensor node SoC that includes most required functions on-chip.

WIRELESS NETWORK ARCHITECTURES

Developers use wireless networks in a spectrum of applications that lie between two extremes: the infrastructure mode and the ad hoc mode. In the infrastructure mode, mobile nodes communicate through *base stations,* special nodes that link together through a conventional network.

When a mobile node wants to communicate, it first registers at a base station in the direct communication range. When a node wants to send a packet to another node, it sends it to the base station. If the destination is registered to that base station, the base station transmits the packet directly to the destination node. Otherwise, the base station forwards the packet through the infrastructure to the base station where the destination node has registered. Typical examples of this infrastructure mode include cellular telephony, paging systems, and wireless LANs that use IEEE 802.11.

In the ad hoc mode, there is no base station infrastructure. If the destination is in range of the source node, that node sends the packet to the destination node. If the destination is not in range, the source node sends the packet to an intermediate node, which forwards the packet to other nodes until the packet reaches its destination or fulfills some other termination criterion. With this architecture, successive hops transport a packet from the source to the destination over several nodes—hence the term *multihop transmission*. PicoRadio is one example of such a network architecture.[2]

A sensor node can be integrated into a single system on chip to minimize power consumption and reduce the cost.

Both architectures assume that there is a way to find the route a packet must follow from its source to its destination. Defining this route is the subject of intensive research. Here, we assume that nodes are static enough for this problem to be considered a secondary issue.

The infrastructure-based architecture is popular for several reasons, particularly its relative simplicity. Base stations do not have power restrictions and enjoy a better spectrum usage because they allow frequency planning. On the other hand, advocates of ad hoc networks cite their higher versatility and potentially lower power consumption. Since no planning and no infrastructure are required, ad hoc networks can be deployed quickly and in remote areas.

Exploiting multihop communication can reduce the transmission range for each hop. Theoretically, as the attenuation increases at least quadratically with the distance, it is more beneficial from the power-consumption viewpoint to use two hops of length L than a single hop of length $2L$. While true in principle, with current transceiver efficiency, thermal power dissipation dominates radio-frequency-radiated power, which significantly diminishes the power advantage of multihop transmission.

Other hybrid solutions lie somewhere in between these two extreme architectures. For example, an infrastructure-assisted architecture would mix both approaches. A mobile node beyond a base station's range could use other nodes in the range to relay the packets to the base station. In this architecture, a packet may or may not go through the infrastructure depending on the location of the source and destination nodes. Similarly, two or more separate infrastructure-based networks can exchange packets through a sequence of hops on mobile nodes, thus forming a single network.

A building sensor system is an example of such a hybrid architecture. Rooms can be equipped with wireless sensors for temperature, moisture, gas, and fire detection. These sensors relay the information toward a number of base stations located in the building, which link to one another through a wired network. The information can thus be made

Without proper design, communication will increase network power consumption significantly.

available to control rooms and, possibly, other remote locations.

To offer maximum flexibility for different kinds of applications, the WiseNET project sought to achieve low-power operation for all three network operation modes.

REDUCING POWER CONSUMPTION

To reduce power consumption, nodes remain sleeping until they need to undertake a specific task. At some defined time, a sensor node will wake up and perform a measurement. An external event also can trigger this wake-up. The node can then decide to communicate the gathered information to a neighbor and send it a message. Unfortunately, the neighbor might be sleeping to save energy. The node must thus keep sending the information until the neighbor awakens and acknowledges receipt of the information.

If a node needs information from a neighbor, it can transmit a request until it receives a response. Alternatively, the requesting node can stay awake and wait until the neighbor decides to send the information spontaneously.

These two simplistic examples show that, without proper design, communication will increase network power consumption significantly because listening and emitting are power-intensive activities. The challenge, then, is to minimize energy consumption by reducing and eventually even eliminating the energy waste caused by

- idle listening—a node waits, listening until another node emits a message;
- overemitting—a node sends a message and the destination node is not ready to receive;
- overhearing—a node listens for a message that is sent to another node and for which it is not the destination; or
- collisions—two nodes transmit at the same time with the consequence that they must retransmit the messages later.

Reducing wasted energy guided development of WiseNET's MAC design.

HARDWARE AND SOFTWARE CODESIGN

Achieving the WiseNET project team's ultimate goal—to reach the lowest possible power consumption for hybrid networks—requires proper hardware and software codesign. Often, communication protocols are designed for existing integrated circuits, forcing developers to take into account the special features and limitations of those circuits.

On our project, we decided from the start to design the radio and protocol concurrently. A few important parameters of the radio clearly impact the higher protocol layers, including

- power consumption in receive and transmit mode;
- wake-up time—the time necessary to switch from idle mode to receive or transmit mode; the node cannot receive while power consumption is approximately the same as in receive mode;
- bit and frame synchronization time at reception because the frame synchronization pattern has an impact on both the overhead—its length—and the frame error rate when the system detects frame starts erroneously;
- the presence of an effective *receive signal strength indicator*, which can be used to improve the reliability of frame detection; the RSSI also reduces idle listening when used as a silence detector;
- some way to filter incoming packets to reduce overhearing;
- the time necessary to switch from receive to transmit mode or vice versa, during which no communication is possible while power consumption is at least that of the frequency synthesizer;
- receiver sensitivity and maximum transmit power, which affect the required number of hops;
- the capacity to adjust transmit power and receiver sensitivity to reduce the frame error rate and collision probability; and
- power consumption in sleep mode with a running, accurate clock that, depending on the application, can consume the major share of the current during the node's life.

The special context of sensor networks required making some basic choices. First, we decided not to minimize the transmit power, but rather to choose a reasonably high constant value—such as the maximum allowed transmit power in the chosen bands, which is 10 dBm for the European 434-MHz ISM band. Because nodes usually transmit rarely, the transmit energy is not the most important parameter to be optimized. Second, we concentrated on reducing both the energy consumption and wake-up time in receive mode. Since WiseNET uses a form of preamble sampling at the MAC layer, waking up quickly has a clear impact on energy consumption.

For these reasons, we carefully elaborated the RSSI information. For a third level of optimization, we added robust bit synchronization and packet filtering based on a programmable pattern. Both these measures reduce frame losses, while packet filtering additionally reduces overhearing. In the first MAC simulations, we found that the capacity to adjust transmit power and receiver sensitivity was less important; therefore, we chose not to implement this feature because doing so would require additional circuitry.

PREAMBLE SAMPLING

WiseMAC is a single-channel contention protocol based on nonpersistent *carrier sense multiple access*. Combining nonpersistent CSMA with preamble sampling mitigates idle listening. The preamble sampling technique consists of regularly sampling the medium to check for activity. In this context, sampling means periodically measuring the received signal strength.

All nodes in a network sample the medium with the same constant period, but their relative sampling schedule offsets are independent. If a node finds the medium busy, it continues to listen until it receives a data packet or the medium becomes idle again. At the transmitter, a wake-up preamble of size equal to the sampling period is transmitted ahead of every data packet to ensure that the receiver will be awake when data transmission begins. This technique enables low power consumption when traffic is low, as is usually the case in sensor networks. It also provides the lowest possible power consumption in the absence of traffic and for a given wake-up latency using a conventional receiver.

The main disadvantage of the fixed-length preamble protocol is its high power consumption overhead, both in transmit and receive, due to the wake-up preamble. Also, in an ad hoc network, both the intended destination and all other nodes overhearing the transmission pay the cost of reception.

WiseMAC introduces a novel scheme to reduce the length of this costly wake-up preamble. As Figure 1 shows, the scheme learns the sampling schedule of direct neighbors and exploits this knowledge to use a minimized wake-up preamble.

The nodes learn or refresh their neighbor's sampling schedule during every data exchange by piggybacking into the acknowledgment messages the remaining time until the next sampling instant. Every node keeps an updated table of sampling time offsets for all its usual destinations. Since a node has only a few direct destinations, it can manage such a table even with limited memory resources.

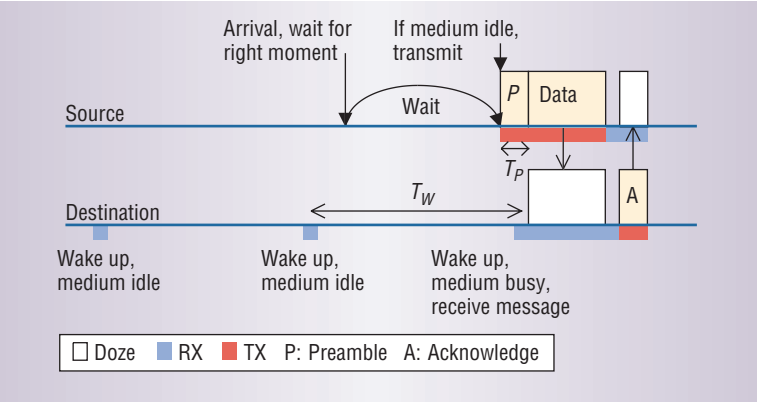


Figure 1. WiseMAC preamble minimization. A low-power, media access control protocol, WiseMAC uses a scheme that learns the sampling schedule of direct neighbors and exploits this knowledge to minimize the wake-up preamble length.

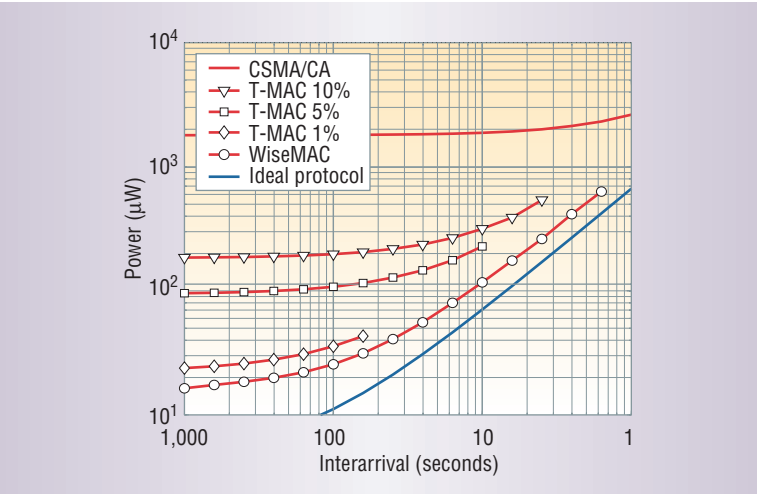


Figure 2. Lattice multihop topology simulation. The results show average power consumption as a function of the network's node traffic. The T-MAC protocol can provide either low-power consumption in low-traffic conditions or high throughput.

The duration of the wake-up preamble must cover the potential drift between the source clock and the destination clock. This drift is proportional to the time since the last acknowledgment was received. Thus, the wake-up preamble's required duration is given by $T_P = \min(4\theta T_C, T_W)$, where θ is the frequency tolerance of the time-based quartz, T_W is the sampling period, and T_C the interval between communications. WiseMAC schedules a transmission so that the middle of the wake-up preamble coincides with the destination's expected sampling time. Using a randomized wake-up preamble mitigates the systematic collisions that this synchronization can introduce.

Because the preamble's length is proportional to the interval between packets, WiseMAC's overhead adapts to the traffic. WiseMAC can thus provide both ultralow average power consumption in low-traffic conditions and high energy efficiency in high-traffic conditions. Figure 2 shows simulation results of a lattice multihop topology.

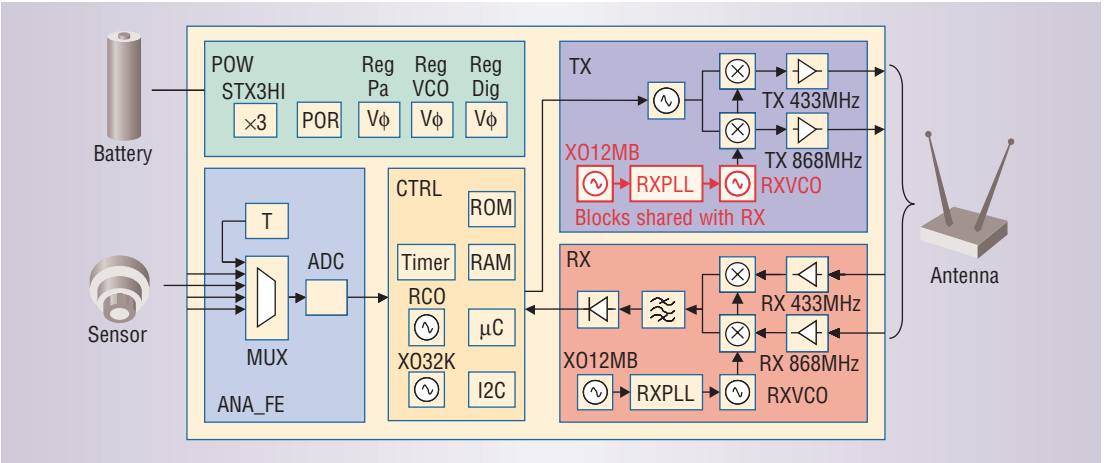


Figure 3. Generic WiseNET SoC building blocks. In addition to the ultralow-power dual-band radio transceiver (Tx and Rx), the architecture includes a sensor interface with a signal conditioner and two analog-to-digital converters (ANA FE), a digital control unit based on a Cool-RISC microcontroller (µC) with on-chip low-leakage memory, several time-basis and digital interfaces, and a power management block (POW).

In the chosen radio range, every node has eight neighbors. Traffic flows through each node, with the interarrival time shown on the *x*-axis. In addition to providing power consumption below 20 microwatts in low-traffic conditions, WiseMAC can approach the power consumption of an ideal protocol in high-traffic conditions. With an interarrival time of 100 seconds, the power consumption amounts to as little as 25 microwatts—which translates into more than a five-year lifetime for a single AA alkaline battery.

For comparison, Figure 2 also shows the performance of

- T-MAC[6]—an improved version of S-MAC[7] with different duty cycles,
- CSMA/CA—CSMA with collision avoidance, and
- an ideal protocol.

We used the WiseNET transceiver's power consumption and timing parameters to simulate these protocols.

CSMA/CA's power consumption is limited at low-traffic levels by the power consumed in receive mode because the transceiver is never switched off. The ideal protocol represents the lower bound that low-power MAC protocols should strive for—the minimum power consumption required to transmit the data without any overhead.

The T-MAC protocol requires choosing the duration of the listen and sleep phases. In the absence of traffic, this ratio is actually the transceiver's duty cycle. In Figure 2, the T-MAC protocol is plotted only up to the point above which more than 5 percent of the packets drop because of congestion. T-MAC can thus provide either low-power consumption in low-traffic conditions, or high throughput.

We selected a single-channel contention protocol to ease self-configuration. WiseMAC requires no setup signaling or network-wide time synchronization. The combination of preamble sampling and wake-up preamble-length minimization provides both ultralow power consumption in low-traffic conditions and high energy efficiency in high-traffic conditions.

Although designed initially for multihop networks, WiseMAC has also proven suitable as the downlink of an infrastructure network. In low-traffic conditions, WiseMAC provides lower power consumption than the power-save scheme that IEEE 802.11 and IEEE 802.15.4 specify.[8] WiseMAC can thus be used in a hybrid network topology, to receive data from both battery-powered nodes and energy-unconstrained base stations.

WISENET NODE ARCHITECTURE

Since a WiseNET network has many distributed microsensors, we do not envision replacing or recharging the batteries. The nodes therefore require long-term autonomy—typically more than three years—and, consequently, low average power consumption. In addition, the nodes must be tiny to fit into all kinds of spaces and, given their high number, they must also be inexpensive. Building from these basic specifications, we used a SoC approach to design the nodes as highly integrated devices that use a dedicated integrated circuit.

Figure 3 shows the basic WiseNET SoC architecture. In addition to the ultralow-power dual-band radio transceiver, the SoC also includes a sensor interface with a signal conditioner and two analog-to-digital converters, a digital control unit based on a Cool-RISC microcontroller with on-chip low-leakage memory, several time-basis and digital interfaces, and a power management block.

Although all these blocks contribute to power consumption, the most critical block is the RF transceiver.

Researchers have found that power consumption as low as one milliwatt in receive mode can be achieved with a –95-dBm sensitivity at a 24-Kbps data rate in the 434-MHz European ISM band.[9,10] These encouraging results led to the design of the WiseNET transceiver along roughly the same lines, but taking advantage of moving from a 0.5-micrometer to a 0.18-micrometer standard digital CMOS process.

Design objectives

In addition to incorporating the features listed in Table 1, our most important design objectives were to

- keep the power consumption within the 1-milliwatt range while in receive mode, adding the 868-MHz SRD band to extend the operation to dual band;
- achieve several years of autonomy by operating from a single 1.5-V AA alkaline battery with a supply voltage as low as 0.9 V, corresponding to the battery end-of-life voltage;
- use a 0.18-micrometer standard digital CMOS process that has no precision analog components such as capacitors and resistors or dedicated RF technology options such as substrate isolation; and
- minimize both the number of external components and the cost.

Dual-band operation, combined with the use of several channels, allows for frequency diversity to solve the difficult problem of rejecting strong nearby interferers. We chose to stay in the 434-MHz and 868-MHz bands instead of moving to the popular and globally available 2.4-GHz band mainly to limit power consumption. Indeed, about 50 percent of a receiver's power consumption relates directly to the circuits operating at or close to the RF frequency, such as the frequency synthesizer and the RF front end, which consists of a low-noise amplifier and a power amplifier.

Since the power consumption of these blocks is approximately proportional to frequency, choosing sub-GHz bands limits power consumption to the milliwatt range. In addition, operating at 2.4 GHz with the chosen 0.18-micrometer CMOS process would require a supply voltage on the order of 1.8 V, which is incompatible with the target 0.9-V minimum supply voltage.

Table 1. WiseNET sensor network characteristics.

WiseNET radio parameters

Operating frequency	433 MHz (ISM) and 868 MHz (SRD)
Channel separation	600 kHz (primary), 200 kHz (secondary)
Propagation range	~2 km outdoors — ~10 m indoors
Data rate/modulation	<100 Kbps with FSK (Δf = 25 kHz) <4 Kbps with 00K
Power consumption in Rx mode	1.8 mW
Power consumption in Tx mode	31.5 mW
Wake-up time	800 µs
Rx to Tx and Tx to Rx turnaround time	400 µs

Main measured results for the Rx and Tx blocks.

Supply voltage	V_{DD} = 0.9 V — 1.5 V (Rx and Tx)
Sleep current	3.5 µA
Receiver (Rx) (measured at V_{DD} = 1 V and at 25 °C)	
Sensitivity	–105 dBm @ BER = 10^{-3} and 25 Kbps
PLL phase noise	–120 dBc/Hz @ 600 kHz offset
Supply current	I_{Rx} = 2 mA
Transmitter (Tx) (measured at V_{DD} = 1 V and at 25 °C)	
Output power	10 dBm
Efficiency @ 10 dBm	30%
Supply current @ 10 dBm	I_{Tx} = 24 mA (PA-preamp)

Radio parameters

Choosing a 0.18-micrometer CMOS standard digital process allows the selected subGHz-frequency bands to trade the high-frequency capability of minimum-length transistors with lower current consumption by biasing the devices at lower current densities, even for devices working at radio frequency. We achieve this by moving the transistor operating points to the moderate and weak inversion regions.[11] Biasing the devices in moderate inversion also offers a good tradeoff between high-current efficiency, low-voltage operation, and reasonable linearity.

The selected radio architecture also strongly conditions power consumption. Although we sought a highly integrated solution, implementing the rejection of strong nearby interferers on-chip would have required prohibitive dynamic range and power consumption due to the low-voltage requirement. To circumvent this problem, we used an external RF filter.

The radio architecture we chose, which builds on similar work done by other researchers,[12] consists of a superheterodyne with a high intermediate frequency, followed by conversion to DC. This architecture offers the advantage of achieving a significant gain at the intermediate frequency without prohibitive power consumption, while the channel selection is performed around DC. Having sufficient gain at the intermediate frequency also reduces the signal-to-noise ratio degradation from the strong 1/f noise present in deep-submicron CMOS processes. This noise strongly affects the

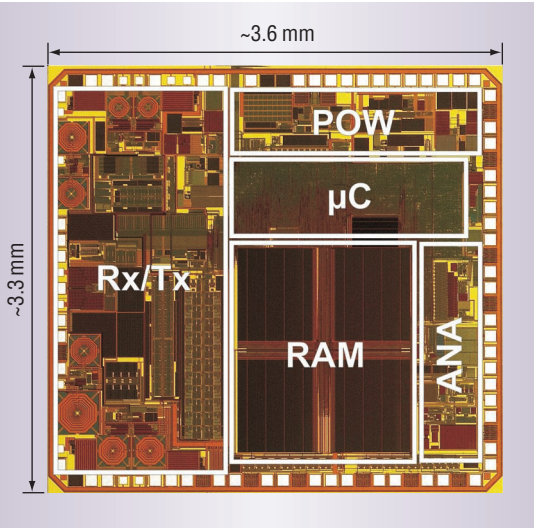


Figure 4. The WiseNET system-on-chip sensor node. Key SoC components include the dual-band transceiver (Rx/Tx), the sensor interface with two ADCs (ANA), the power management block (POW), the control unit (µC) with an 8-bit CoolRISC processor, and the embedded low-leakage memory (RAM).

analog baseband. Other external devices include a high-Q inductor used for the LC-tank circuit to achieve the desired low-phase noise and a few capacitors for impedance matching to the antenna.

Power consumption

We also designed a special on-chip varactor to operate at the required low voltage, while still offering a sufficient tuning range and a high-quality factor that helps avoid degrading the tank's overall Q-factor.[13] The WiseNET transceiver offers a digital RSSI as well, which the WiseMAC protocol uses for preamble sampling and carrier sense activities.

If all contributors within the receive chain are on, the total current drain makes long-duration operation on a single battery impossible. Analyzing the individual contributions reveals the current consumption to be significantly larger for the RF blocks and smaller for the baseband blocks. This disparity occurs because the current directly relates to the frequency of operation or the required bandwidth.

On the other hand, turn-on times will vary inversely with the frequency of operation because baseband blocks require more time for all the nodes in the circuit to reach a quiescent state after being turned on than RF blocks. Therefore, the system can save significant energy by waking up the lower-power baseband blocks before the power-consuming RF circuits, which wake up quickly.

Optimization parameters

The WiseNET chip offers a high degree of flexibility in controlling each block's individual operation. A wake-up sequence implementation can, for example, start with powering on the low-frequency reference clock; then the baseband path of channel filters, limiters, and RSSI; followed by the frequency synthesizer; and, eventually, the intermediate frequency amplifiers and the RF front end's low-noise amplifier and down-conversion mixers.

In addition, the RSSI can efficiently leverage WiseMAC's sampling nature by turning on only the required block in the baseband. Once it has activated the receiver chain up through the baseband channel filter, the RSSI can measure the signal strength. If it determines that the detected activity level is sufficient, the RSSI can turn on the demodulation blocks for further symbol and frame analysis and error correction. If the power level is lower than an acceptable threshold, the RSSI turns off the complete receiver, thus optimizing the power burned during both idle listening and receiving. The RSSI can use the same power sequencing techniques for the transmit path.

The second most important optimization parameter is the wake-up time. To optimize this parameter, we carefully defined the activation sequence for the different transceiver blocks. Implementing this measure mandated using proper circuit design techniques to minimize the wake-up times of the slower blocks.

This delay is not an issue for RF blocks because the larger currents and small parasitic capacitances required to achieve RF performance allow very quick settling. Given the significant energy savings it can provide, optimizing the setup delay in the intermediate frequency amplifiers, intermediate frequency to baseband mixers, and baseband blocks is worth investigating. Deep-submicron CMOS offers the clear advantage of making operation within the 100-MHz range feasible with proper biasing of the signal path transistors in weak or moderate inversion regions. In the chosen high intermediate frequency architecture, the intermediate frequency blocks can operate at frequencies that minimize setup time without degrading power consumption.

Baseband blocks exhibit an obvious limitation due to their inherent low-frequency poles. Nonetheless, techniques such as storing the bias points between channel sampling, boosting the start-up currents momentarily at wake-up, or switching the internal poles to higher frequencies for fast settling and then setting them back to the nominal frequency reduce the settling times.

The third key optimization parameter is the receive-to-transmit and transmit-to-receive switching times. During these turnaround delays, all blocks that remain on will burn power that is extraneous to the communication. Thus, we designed the WiseNET SoC architecture shown in Figure 4 to meet this particular MAC parameter. For example, the receive and transmit paths both use the same synthesizer and LO signals. The design makes this possible because it implements the receiver and transmitter according to the same high intermediate frequency superheterodyne architecture. In this way, the synthesizer remains on, working around the same operating points to yield fast turnaround times.

Using the WiseNET transceiver with the WiseMAC protocol, a relay sensor node consumes about 25 microwatts when forwarding 56-byte packets every 100 seconds. The radio transceiver used on motes[14] requires 24 milliwatts in receive mode or 8 milliAmps at 3 V. Using the S-MAC or T-MAC protocols with such a transceiver, at a duty cycle of 10 percent, yields a minimum power consumption of 2.4 milliwatts—about 100 times more than that achieved with the WiseNET solution we have presented.

In a next step, we plan to study how to leverage WiseMAC for routing in more dynamic networks. Finding power-efficient solutions to achieve data consistency and action synchronization will be next on the agenda. On the hardware side, we currently are investigating new narrow-band radio architectures using high-Q bulk acoustic wave resonators. These architectures should allow even lower power consumption and higher operating frequencies. Alternatively, ultrawideband techniques using impulse radios could make the communication more robust to fading, while maintaining low power consumption.

Acknowledgments

We thank the Swiss Center for Electronics and Microtechnology design team for its invaluable contribution to the success of the WiseNET project, particularly T. Melly and E. Le Roux for the information they provided. The work presented here was supported in part by the National Competence Center in Research on Mobile Information and Communication Systems, a center supported by the Swiss National Science Foundation under grant number 5005-67322.

References

1. I.F. Akyildiz et al., "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, 2002, pp. 393-422.
2. J.M. Rabaey et al., "PicoRadio Supports Ad Hoc Low Power Wireless Networking," *Computer*, July 2000, pp. 42-48.
3. G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM*, May 2000, pp. 51-58.
4. A. Chandrakasan et al., "Power Aware Wireless Microsensor Systems," *Proc. European Solid-State Circuits Conf.*, Univ. of Bologna, 2002, pp. 47-54.
5. A. El-Hoiydi et al., "WiseMAC: An Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network," *Proc. 1st ACM SenSys Conf.*, ACM Press, 2003, pp. 302-303.
6. T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. 1st ACM SenSys Conf.*, ACM Press, 2003, pp. 171-180.
7. W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE Infocom Conf.*, IEEE Press, 2002, pp. 1567-1576.
8. A. El-Hoiydi and J-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks," *Proc. IEEE Symp. Computers and Comm.*, IEEE Press, 2004, pp. 244-251.
9. A-S. Porret et al., "An Ultralow-Power UHF Transceiver Integrated in a Standard Digital CMOS Process: Architecture and Receiver," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, 2001, pp. 452-466.
10. T. Melly et al., "An Ultralow-Power UHF Transceiver Integrated in a Standard Digital CMOS Process Transmitter," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, 2001, pp. 467-472.
11. C. Enz, "An MOS Transistor Model for RF IC Design Valid in All Regions of Operation," *IEEE Trans. Microwave Theory and Techniques*, Jan. 2002, pp. 342-359.
12. H. Darabi and A. Abidi, "A 4.5-mW 900-MHz CMOS Receiver for Wireless Paging," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, 2000, pp. 1085-1096.
13. A-S. Porret et al., "Design of High-Q Varactors for Low-Power Wireless Applications Using a Standard CMOS Process," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, 2000, pp. 337-345.
14. Crossbow, "MICA 2—Wireless Measurement System Datasheet," 2004, document part number 6020-0042-04; www.xbow.com.

***Christian C. Enz** is the vice president of the micro-*

electronics division at the Swiss Center for Electronics and Microtechnology (CSEM). He is also a professor at the Swiss Federal Institute of Technology, Lausanne (EPFL). His research interests include low-power analog and RF IC design, wireless sensor networks, and device modeling. Enz received a PhD in electrical engineering from the EPFL. Contact him at christian.enz@csem.ch.

***Amre El-Hoiydi** is an R&D engineer at CSEM. His main research interest is the design of energy-efficient communication protocols for wireless sensor networks. El-Hoiydi received an MS in electrical engineering from ETH Zurich. Contact him at amre.el-hoiydi@csem.ch.*

***Jean-Dominique Decotignie** is a section head at CSEM. His research interests include real-time and power-aware networks and middleware for robustness and reliability of nomadic systems. He is also a professor at the EPFL. Decotignie received a PhD in electronic engineering from the EPFL. He is Fellow of the IEEE. Contact him at decotignie@ieee.org.*

***Vincent Peiris** heads the RF and Analog IC design group at CSEM. His research interests include low-power RF CMOS transceiver design for wireless sensor networks. Peiris received a PhD in electrical engineering from the EPFL. Contact him at vincent.peiris@csem.ch.*

The Flock: Mote Sensors Sing in Undergraduate Curriculum

Integrating wireless sensor networks in an undergraduate embedded systems course exposes students to an important emerging technology in the core of the computer engineering curriculum.

Bruce Hemingway
Waylon Brunette
Tom Anderl
Gaetano Borriello
University of Washington

Computer engineering curricula have evolved dramatically over the past 20 years. The early focus was on computer architecture and CPU design. In the 1990s, attention shifted to systems built around highly integrated microcontrollers. By 2000, the software for these embedded systems assumed a more important role, and embedded operating systems provided high-level design abstractions. More recently, wireless communication capabilities have greatly expanded the embedded applications space, leading to new system paradigms such as sensor networks.

Educational excellence requires exposing students to the current edge of research. To ensure that student projects are along the same trajectory that the industry is traveling, educators must continually introduce emerging techniques, practices, and applications into the curriculum.

At the University of Washington's Department of Computer Science & Engineering (UW CSE), we have integrated the emerging field of wireless sensor networks into our undergraduate computer engineering curriculum. While many graduate-level classes focus on sensor networks,[1-3] they do not provide a good template for the undergraduate curriculum because they assume a much greater breadth of knowledge than undergraduate students usually have as well as greater maturity to absorb new topics on their own.

Other efforts to introduce undergraduates to this important application area have focused on using sensor networks in capstone project courses that challenge senior students to apply their acquired knowledge and skills to a final project. The UW CSE "Flock of Birds" project integrates the theory and practice of wireless sensor networks into the mainstream curriculum early enough to form a basis for all students' understanding of embedded computing—not just a short-lived application exercise for some capstone design projects. Figure 1 shows one sensor node—the bird is optional—from the project.

COMPUTER ENGINEERING CURRICULUM

The UW CSE computer engineering program relies on a core curriculum shared with the computer science program. It consists of introductory courses in programming, discrete mathematics, data structures, formal methods, comparative programming languages, logic design, and computer architecture. Computer engineering students specialize with additional second-tier courses in electrical engineering, operating systems, networks, embedded software, and digital system design, as well as a capstone design course that takes a product idea from concept to prototype.

The embedded software, digital design, and capstone design courses give the computer engineering

Published by the IEEE Computer Society
0018-9162/04/$20.00 © 2004 IEEE

program its character. They integrate software and hardware design skills and prepare students to build modern digital systems from start to finish.

In the embedded software course, students learn to use microcontrollers and their interfaces effectively to build systems that control physical devices. The digital design course teaches them to program algorithms into hardware. In the capstone design course, the students apply all their skills to a product that we try to make similar to those on which industry engineers are currently working—in other words, products that will appear on the market one or two years after the students graduate.

CSE 466: Software for Embedded Systems

The students in our embedded software course (CSE 466) have completed computer architecture and digital design courses, and most have studied operating systems as well.

This course exposes the students to the design issues that characterize embedded systems. These include constrained resources, such as limited memory space, I/O, and CPU frequency; the absence of an operating system to handle low-level tasks such as interrupts and peripheral device interfaces; and a variety of protocols for communication between components.

Further, embedded systems often require a debugging tool set different from that of the full-scale systems familiar to our students. Print statements are seldom available, nor are breakpoints always an option. Programmers and, in this case, students must find other methods to signal code milestones.

Through prerequisite courses on data structures, digital design, and machine architecture, CSE 466 students have a solid understanding of how the hardware works and how to build efficient data structures for their algorithms. This gives them the foundation they need to write software at the low level required by embedded systems. The students build on these prerequisites with a hands-on project that is designed to lead them through the process of applying what they already know while exposing them to the new issues that arise with embedded systems.

When students finish the 10-week class, they should be comfortable writing code that directly manipulates I/O registers and establishes communications between multiple devices. They should also understand how interrupts work and how to handle them as well as how to interpret the datasheet of whatever chip they decide to work with in the future.

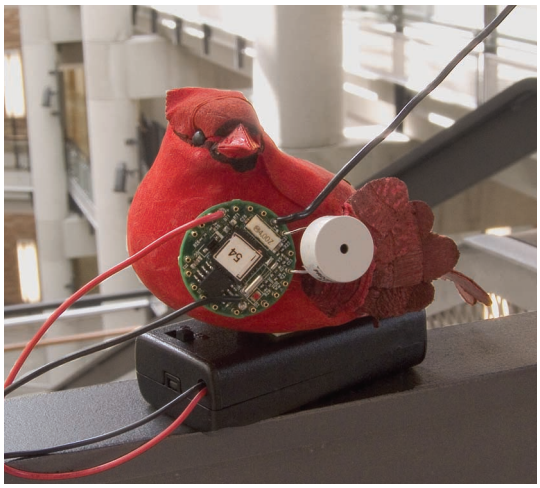


Figure 1. A mote-sensor bird. The "Flock of Birds" embedded systems project integrates the theory and practice of wireless sensor networks into the mainstream classroom curriculum.

Project design for wireless sensor networks

Each edition of CSE 466 must define a project that addresses the complexities of embedded systems within the constraints of a 10-week course. For pedagogical reasons, we want a project that students can complete individually. At the same time, students are generally more engaged by class-wide projects, so we like the individual work to contribute to an overall class effort.

Students are also more interested in projects that use current technology as opposed to obsolete components that they will never see again after they graduate. As both memory and CPU cycles become more plentiful in most of today's microprocessors, designing a project that is resource constrained poses another challenge.

The computer engineering curriculum already used Atmel ATmega AVR-series microprocessors (www.atmel.com) and the AVR-GCC (www.openavr.org) C compiler, so students were familiar with these core materials. To incorporate wireless sensor networks into the curriculum, we selected the Crossbow (www.xbow.com) Mica2Dot platform. The product is an implementation of TinyOS sensor motes (www.tinyos.net), originally created at the University of California, Berkeley.[4] Mica2dot features a standard platform with built-in hardware, the existing TinyOS code base, and a convenient form factor for adding predefined sensors.

The event-based style of TinyOS helped students understand time constraints and code structure by forcing them to write short, nonblocking routines. Its modular design simplified the integration of components like the radio stack, saving countless hours of coding. TinyOS also provided a degree of abstraction within the embedded system context and introduced students to a style of event-based coding that stresses real-time as well as functional issues.

We made sound generation a major focus of the project and the overall class. Students can understand and implement sound generation in a month, yet it taxes a system's cycle and memory capacity enough to make efficiency an important design con-

Figure 2. The Microsoft Atrium in the Paul G. Allen Center for Computer Science & Engineering, University of Washington. The "Flock of Birds" project concluded with a concert in the atrium consisting of 50 "bird" motes that sang songs based on what other birds within radio range were singing. Photo courtesy of Ed LaCasse.

straint. A sample rate in the tens of kilohertz range can generate usable sound. Visual displays such as video on LCDs demand many more computation cycles, but human observers seldom notice visual timing errors on the order of tens of milliseconds. On the other hand, errors of even a few microseconds are discernible in sound generation, giving students quick feedback on program accuracy.

A FLOCK OF BIRDS

The "Flock of Birds" project is a simple distributed system that meets our course objectives by combining sound generation with emergent behavior in an ad hoc network. Each student programs a mote to act as a bird that has several songs stored in its local memory. The programs execute a common rule base, but each bird acts independently—deciding which song to sing based on what the other birds within radio range are singing. In combination, the songs create the sound a flock of birds makes.

The flock is designed to work in any random configuration with any number of nodes. We implemented the project in the Microsoft Atrium of the University of Washington Paul G. Allen Center for Computer Science & Engineering, shown in Figure 2.

The primary goal was subjective: to generate behavior that mimics the effect of birds cooperating to sing the same song but vary the particular song over time. We implemented monitoring software to measure the effect, but the final judgment of success was aesthetic: Does it sound right? The subjective measure contrasts with the usual quantitative measures we used in most of our past projects. A visual display of the data associated with the sound reinforced the aural perception.

The algorithm for generating behavior uses radio packets collected for random amounts of time from other motes. After the time lapses, the mote decides which song to sing based on the data about what songs the neighboring motes are singing. The mote then shuts its radio off, sings its song, turns the radio back on, and transmits a packet announcing which song it just sang.

This algorithm requires using timers, radio communication, and direct manipulation of hardware registers for sound production. Students had to first develop a complex software module on a breadboard and then explore the challenges of porting it to the mote within the TinyOS constraints.

A monitoring node served as the gateway to a control laptop that used special radio packets to start and stop the algorithm and to modify global parameters, such as the min/max limits for the random number generation that determined an individual node's behavior. This added a dynamic aspect to the flock process. For example, we could control the temporal density of bird songs by changing the maximum value for the listen time between song events. Lowering the radio transmit level in the large space of the atrium prevented distant motes from hearing each other and made the flocking effect operate in multiple local areas.

We presented the flock concept to the students in the context of emergent behavior and the basics of cellular automata and gave them the simplified template algorithm shown in Figure 3.

The initial flock specification was purposely incomplete, as we wanted student feedback and participation in completing its definition. This is analogous to what students might experience in the real world, where others have decided on a protocol and high-level algorithm, and the developer's job is to implement a system that realizes the functionality.

The incomplete specification also exposed students to different interpretations of the design documentation. They had to identify the ambiguities and then clarify them either with other developers or with us, "the customer."

We asked students to invent a methodology for predicting the success of this algorithm and to suggest three improvements to it. We incorporated many of these suggestions into the final algorithm to enhance students' experience of participating in the project design. A complete procedural docu-

ment is available on the class Web site (www.cs.washington.edu/education/courses/cse466/03au/).

COURSE WORK

Most students in the course had no prior embedded programming experience. To keep the first few assignments simple, easy to debug, and somewhat familiar, we had the students begin programming with embedded C rather than TinyOS. The assignments introduced some basic embedded systems programming concepts such as decoding binary numbers to pins for a pair of seven-segment displays, using an analog-digital converter to measure sensor voltage and current levels, and writing interrupt handlers for timers.

Breadboard basics

The students compiled their programs with AVR-GCC and uploaded them to an ATmega16 processor on a solderless breadboard. These chips come in a standard dual-inline package that is easy to place on a standard breadboard. The breadboard allowed space for students to add peripheral devices and debug their circuits by connecting probes for oscilloscopes and logic analyzers.

After introducing the basics, the course moved quickly to more specific applications—most notably, sound generation with a piezoelectric transducer. The first sound assignment involved using a wave table to generate sine waves at various frequencies. This introduced the students to the notion of using a fixed sampling rate with a phase increment to control frequency.

Also, because the ATmega microprocessor series has no digital-to-analog converter, the students had to use pulse-width modulation instead to produce analog outputs to drive the piezoelectric transducer. Using PWM for this purpose was new to many students, and few of them saw immediately that they could use a low-pass filter in conjunction with PWM to produce a reasonably accurate DAC. By the assignment's end, however, all the students had at least a basic grasp of both PWM and simple hardware-based low-pass frequency filtering.

For the next assignment, students had to build a system that could play several different songs. The system implemented familiar songs based on cell phone ring tones, ranging from *Für Elise* to *Theme from "The A Team."* Using familiar songs helped the students quickly determine whether their implementation worked. If they recognized the song, they were on the right track; if not, they likely had a problem. Using familiar songs also raised the interest level in the class. This was one of the rare times that computer engineering students could show off their projects to nontechnical friends.

Goals for the flock:
Birds sing the same song for a little while.
Songs start, spread, then die out.
Over time, different songs emerge as dominant for some period of time.
Flock process flow:
1. Initialization tasks; select x = random(0-15).
2. Radio off; sing birdsong[x]; radio on.
3. Listen for Random(min1, max1) sec.
4. SendMessage "I sang song x".
5. Listen for Random(min2, max2) sec.
6. Decide which song to sing next:
   a. Determine nearest songs.
   b. If my song is the same as any of the nearest songs, then I'll repeat the same song.
   c. If all nearby motes are singing the same song, then I'll switch to a different song.
   d. If all nearby songs are distinct, then I'll switch to a different song.
7. Go to step 2 and repeat.

Figure 3. Goals and simplified algorithm for each bird in the flock.

Sound generation uses both wave and sequencing tables, which require large amounts of memory—more than main memory could accommodate. Because the standard method of simply putting these tables in RAM as static arrays was not possible, students had to put the data in program space. This was slightly less convenient because the lookup required special calls. However, it introduced students to standard practice for handling static data in an embedded environment. This reinforced the concept of memory mapping and resource allocation.

The final assignment on the ATmega16 was to play 16 different bird songs. The songs were in MIDI format, which is slightly different from the previous format and required the students to convert to a storage format of their own devising. However, the sound-generation portion of the assignment was very similar to the earlier assignment, so students were able to reuse much of the code they had already written.

Students were not as familiar with the birdsongs as they were with the tunes from the previous assignment. By playing the MIDI files on their PC and comparing the sound with their implementation's, they were able to get their code up and running quickly.

The ATmega128 processor that the Mica2dot motes use is very similar to the ATmega16, so much of the code from this assignment was reused as a component in the final project.

Motes and TinyOS

Once the students were comfortable with basic embedded-systems programming issues, we introduced motes and TinyOS. TinyOS is based on an event-driven paradigm instead of the polling paradigm that was standard for previous student assignments. Some students had been exposed to the

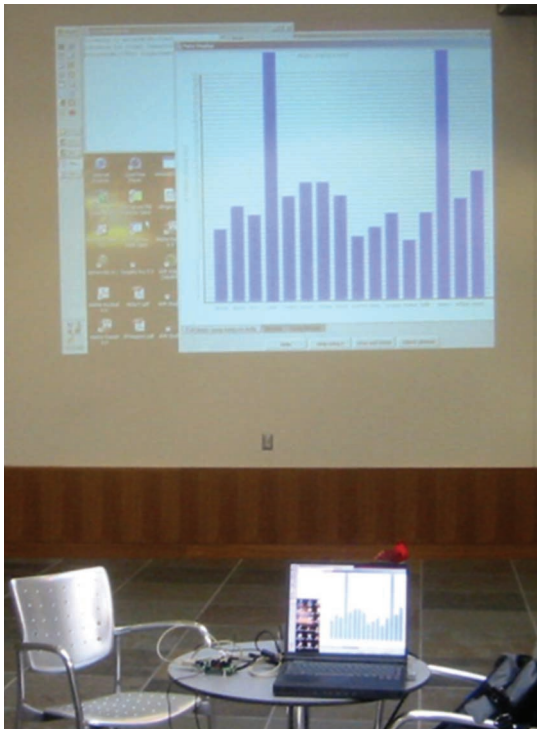


Figure 4. "Flock of Birds" monitoring laptop and wall display. The frequency of different bird songs was projected in real time during the concert.

event-based paradigm through operating system or user-interface programming classes, but many of them had not.

Students spent the first few weeks learning what a component, module, and interface were and how to wire modules together. To add to the confusion about event-driven programming, the students struggled to learn NesC (http://nescc.sourceforge.net/), an extension to the C programming language designed at UC Berkeley for TinyOS.[5]

We used tutorials from the TinyOS Web site and simple assignments to introduce this subject to students. After a rough start, students began to see how quickly they could build programs with considerable functionality and only a little custom code connected to existing software components.

Many students used a 15.625-kHz sampling rate on the Crossbow Mica2dot motes. Running at the standard 4 MHz, the system outputs samples every 256 cycles. This forced students to code their sampling algorithms carefully so that the processing could finish within the 256 cycles.

Debugging the Mica2dot motes presents interesting challenges. Traditionally, students debug their programs by embedding print statements in their code or using interactive debuggers with breakpoint and inspection capabilities. The Mica2dot has one light-emitting diode for use in feedback. While TinyOS includes a simulator, many of the mote components that control hardware, such as the radio and sound-generation components, cannot be effectively debugged with the simulator. Students could use some print statements with the serial port while the mote is on the programmer, but they had to be more inventive when debugging in the wireless communication environment.

Given that the flock algorithm runs on a mote among many other motes running the same algorithm, each node receives large amounts of input that the test cycle must simulate to emulate all the other nearby nodes. This required a substantial test fixture that could produce large numbers of packets at predetermined times and then report on the packets sent by the mote under test.

Instead of having the students implement the entire test fixture, we provided a fixture that took as input a simple description of packets to send and the time delay between them and sent these to the mote under test. This test fixture handled all of the necessary low-level interactions and allowed the students to write arbitrary and repeatable tests. This freed them from having to worry about the details of sending packets from a computer to the mote under test.

Writing these tests also helped the students to better understand the timing details of the communication protocol. The students not only had to figure out how to generate a comprehensive test for the protocol but also how to create situations that would produce specific desired responses. Where they lacked understanding or the documentation lacked detail, they had to chase down their bugs systematically.

Concert exam

The conclusion of the project was a two-hour "concert" of 50 motes in the Allen Center Atrium. Students had to qualify their birds for admission by passing a special test designed to exclude rogue birds from the flock. Well-behaved birds graduated to the Allen Center atrium for the performance. Students reprogrammed failed motes with code from motes that did pass, allowing everyone to participate in the concert.

For the performance, we had to modify the testing program to send specific packets that would trigger the motes to begin their process. The underlying system was already implemented, so we simply built a GUI to it. The GUI also allowed us to monitor the motes and project a real-time wall display of the frequency of different birdsongs the motes were singing. Figure 4 shows a snapshot of the central monitoring laptop and projected wall display during the concert.

We used the existing TinyOS SerialForwarder component to put control motes at various places in the room and have them forward their received packets to the monitoring laptop via 802.11 wireless Ethernet. In this way, we could turn the trans-

mit strength down very low on the bird motes while still ensuring that the monitoring system recorded every packet sent.

The overall sound effect was pleasing, and even worked in three dimensions when some students moved their birds to upper balconies. The aural feedback contrasted with more conventional projects that require extended data analysis to understand the results. The results were also easier to understand than many simulation schemes.

The project succeeded from an instructional perspective—integrating communication protocols, constrained resources, hardware control, and a novel application that required student projects to interact. The motes supported a project that would otherwise have been too complex to implement over a 10-week quarter.

PROJECT CHALLENGES

TinyOS is a work in progress. The large-scale installation in our laboratories was daunting, particularly relative to conflicts with preinstalled tools such as Cygwin and Java. These tools are used systemwide by many more classes and students than our immediate group, and we spent much staff time solving configuration problems. These problems will change with each new TinyOS version, so this aspect of the course will remain time-consuming until the tools and installation procedures mature.

The tutorials provided with TinyOS helped to introduce the students to the basics. However, students could implement the tutorial examples by simply copying code or following instructions by rote. To make sure students understood the underlying concepts, we gave an exam to test their knowledge. This motivated the students to take the time necessary to learn the concepts. After the exam, many students claimed that TinyOS was simple once they understood it. Future versions of the tutorials might benefit from scattering conceptual questions along the way to help students focus on the high-level structures of TinyOS in addition to its mechanics.

Some nondeterminism in our specification often made it difficult to accurately predict whether a mote under test was going to be listening at any given time. Because of this, our testing required us to provide very clear cases in which several dropped packets would not affect the outcome. We also had to allow for some packets to be dropped when transmitting them back from the node under test.

We determined that a fully automated testing system for grading would not be feasible. Instead, we scripted seven tests totaling roughly 10 minutes, after which the staff ran through the data by hand. This was not a problem, however, since each test produced only 5 to 20 packets from the mote—a small number that was easy to examine.

The test program represented a large undertaking for the staff. It totaled more than 4,000 lines of Java even though it used the SerialForwarder component shipped with TinyOS. The program nevertheless proved quite useful to both staff and students, enabling bombardment of a mote with large numbers of random packets as well as just a few carefully chosen packets.

We have considered making several changes to this project if we use it again in the future. First, the sound quality of the piezoelectric transducers, while sufficient, was not exactly high fidelity. The choice was based on its extremely low power requirement, but in other operating environments this might not be a consideration. Research into other transducers or speakers could enable a wider repertory of sounds. For example, we have considered including insects, rain, other animal sounds, and whispered voices, along with contexts and rules sets for emergent behavior to fit the sound types.

We might add input sensors. Light sensing could allow the virtual entity to react to the diurnal cycle of ambient light. Proximity sensors could support behavior modifications according to the movements of people near a mote.

Expanding the project to include rechargeable batteries and power management could allow a life cycle that would require activity only when power was sufficient, in the style of BEAM robots (www.nis.lanl.gov/projects/robot/).

Controlling light-source color and intensity could add visual interest and another level of complexity.

From the instructional viewpoint, a vital ingredient of the course is having an experienced TinyOS programmer on the staff. Until the accompanying teaching materials for motes and TinyOS mature, this expertise will be an important condition in attempts to scale the project. ■

References

1. A. Chien, "Programming Sensor Networks," graduate course CSE 291, Univ. California, Davis, 2003; www-csag.ucsd.edu/teaching/cse291s03/.
2. M. Beigl et al., "Teaching a Practical Ubicomp Course

with Smart-Its," poster at Ubicomp 2002; www.teco.edu/~michael/publication/smart-it_ubicomp.pdf.
3. M. Welsh, "Wireless Communications and Sensor Networks," graduate course CS 263, Harvard Univ., 2004; www.eecs.harvard.edu/~mdw/course/cs263/fa04/.
4. J. Hill et al., "System Architecture Directions for Network Sensors," *Proc. 9th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS-IX), ACM Press, 2000, pp. 93-104.
5. D. Gay et al., "The nesC Language: A Holistic Approach to Networked Embedded Systems," *Proc. Programming Language Design and Implementation* (PLDI) 2003, ACM Press, 2003, pp. 1-11.

Bruce Hemingway *is a lecturer in the University of Washington's Department of Computer Science & Engineering and manager of its W.T. Baxter Computer Engineering Laboratory. His research interests include the development of autonomous computer-music generators that will sense mood from physiological indicators and produce music pleasing to the listener. Hemingway received an AB in music from Indiana University. Contact him at bruceh@cs.washington.edu.*

Waylon Brunette *is a research engineer and teaching associate in the Department of Computer Science & Engineering at the University of Washington. His research interests include mobile and ubiquitous computing, wireless sensor networks, and personal area networks. Brunette received a BS in computer engineering from the University of Washington. Contact him at wrb@cs.washington.edu.*

Tom Anderl *is a teaching assistant in computer engineering at the University of Washington. His research interests include application-specific hardware and resource-limited computation. Anderl received a BS in computer engineering from the University of Washington. Contact him at tanderl@cs.washington.edu.*

Gaetano Borriello *is a professor of computer science and engineering at the University of Washington. His research interests are in ubiquitous computing, principally new hardware devices that integrate seamlessly into the user's environment with particular focus on location and identification systems. Borriello received a PhD in computer science from the University of California, Berkeley. Contact him at gaetano@cs.washington.edu.*

FREE AND DISCOUNTED ONLINE TRAINING

- Improve Your Professional Development!
- Increase Your Career Prospects!
- Sharpen Your Technical Skills and Learn New Ones!
- Build Management Skills!

IEEE Computer Society

Distance Learning Campus

Powered by KnowledgeNet®

FREE Technical Courses

IEEE Computer Society members get **free** access to 350 course modules in more than 40 subjects, including:

- Java
- HTML
- XML
- Oracle 9i
- CompTIA Network+
- CompTIA Security+
- Sun Developer for the Java 2 Platform
- Sun Programmer for the Java 2 Platform
- Sun Microsystems Solaris 9
- Microsoft Project
- Microsoft Excel
- Microsoft PowerPoint
- Designing Security for Microsoft Networks
- UNIX System Administration
- Introduction to Cisco IP Phone
- Cisco Internetwork Troubleshooting
- Interconnecting Cisco Network Devices
- Visual C++
- Programming with C#
- Microsoft .NET
- Programming with Visual Basic .NET
- Microsoft Windows Server 2003
- Implementing Windows XP Professional
- SQL Server 2000

Discounted Management & Business Strategy Courses

IEEE and IEEE Computer Society members get low-cost access to 26 business and management courses from renowned sources such as the American Management Association (AMA), Peter Drucker, and more. Courses include:

- AMA – Negotiate to Win
- AMA – Managing Employee Conflict
- AMA – A Practical Guide to Time Management
- AMA – The Characteristics of Effective Teams
- The Conference Board – Managing Through a Volatile Economy
- The Conference Board – How to Build High-Performance Teams
- Peter Drucker – Effective Decision Making
- Peter Drucker – Permanent Cost Control
- Peter Drucker – Alliances: The Rules for Successful Partnership
- Tom Hopkins – Mastering the Art of Communication
- Forbes – Profit from the Core: Finding and Defining Opportunities for Growth with Christopher Zook

IEEE

Take a course now. View the entire course catalog.
Get more information at…
www.computer.org/DistanceLearning

IEEE COMPUTER SOCIETY

Students Earn Top IEEE Computer Society Prizes at Intel Science Fair

Kimberly Elise Reinhold of Saint Joseph Junior-Senior High School in Hilo, Hawaii, received an IEEE Computer Society prize of $700 at the 2004 Intel International Science and Engineering Fair (ISEF) for her project, "Artificial Visual Perception: An Integrated Approach to Neuroadaptive Modeling." Among other prizes, Reinhold also received a $1,500 award from Intel for excellence in computer science.

Sharing first-place honors with Reinhold was Nimish Ramanlal of Seminole High School in Sanford, Florida, for his project, "A Quantum Algorithm for the Simultaneous Evaluation of Functions: A Combinatorics Solution with Fractal Properties."

Reinhold and Ramanlal were among the more than 1,200 students who gathered in Portland, Oregon, to demonstrate their projects. In preparation for the Portland event, Intel and fair organizer Science Service cooperated in sponsoring nearly 500 preliminary science fairs at high schools located throughout the world. Winners of those regional events advanced to the Portland finals, competing for more than $3 million in scholarships, internships, travel, and equipment grants.

Students compete in 15 categories that include all facets of the sciences with projects that represent their own original work. Professional associations and companies that are active in the ISEF categories send judges to select winners in their specialty areas. In addition, the Intel Foundation bestows three grand-prize Young Scientist Awards upon the individuals whose projects the judges deem best overall. The Young Scientist Award winners each receive a high-performance computer and a $50,000 scholarship paid in eight installments.

Winners of 2004 IEEE Computer Society ISEF special swards pose with (back row, from left) Computer Society judges John Prohodsky and Christina Schober and IEEE President-Elect W. Cleon Anderson.

Nine competitors at Intel ISEF received cash awards from the Computer Society. Five were first- through third-place individual winners, and the remaining four shared first- and second-place team awards. Several participants also received other awards at ISEF.

IEEE Computer Society award winners were:

- **First Place, Individual ($700):** Kimberly Reinhold, of Saint Joseph Junior-Senior High School in Hilo, Hawaii, for "Artificial Visual Perception: An Integrated Approach to Neuroadaptive Modeling."
- **First Place, Individual ($700):** Nimish Ramanlal, of Seminole High School in Sanford, Florida, for "A Quantum Algorithm for the Simultaneous Evaluation of Functions: A Combinatorics Solution with Fractal Properties."
- **Second Place, Individual ($500):** Ho-Seung Ryu, of Pacific Grove High School in Pacific Grove, California, for "An Algorithmic Approach to DNA-Based Data Storage and Error Correction." Ryu also received a $1,000 award from Intel.
- **Second Place, Individual ($500):** Yuanchen Zhu, of Shanghai Foreign Language School in Shanghai, for "Real-Time Remeshing with Optimally Adapting Domain." Zhu also received one of three $50,000 Intel Young Scientist Awards and a $1,000 first place award from the ACM, among other top prizes.
- **Third Place, Individual ($350):**

George Hotz, of the Academy for Engineering and Design Technology in Hackensack, New Jersey, for "Mapping Robot."

- **First Place, Team ($500 per team member):** Soner Yilmaz and Enver Kayaaslan of Samanyolu Private High School in Ankara, Turkey, for "Robot Motion Planning."
- **Second Place, Team ($400 per team member):** Jose Riedel and Daniel Rubino of Technician School No. 9 in Buenos Aires, Argentina, for "Orthopedic Robotics Bed."

Award winners also receive a gift certificate for any Computer Society publication and a one-year subscription to a Society magazine of their choice.

IEEE Computer Society Conferences and Tutorials Board Vice President Christina Schober led a delegation of volunteer judges from the Portland chapter of the Computer Society. The judges were Edward Berg, Nadya Boone, Dave Brown, Bryan Dinteman, John Hartman, Karen Karavanic, Joshua Poulson, John Prohodsky, Sarihan Tan, and Gary Spivey.

Capturing the top prize, the Intel Young Scientist Award, at this year's ISEF were Sarah Langberg of Canterbury School in Fort Myers, Florida, for "Petrology, Morphology, and Geochemistry of the Southern Juan de Fuca Ridge"; Uwe Treske of the Paul Gerhardt Gymnasium in Grafenhainichen, Germany, for "Low-Cost Scanning Tunneling Microscope"; and IEEE Computer Society award winner Yuanchen Zhu. In addition to a $50,000 scholarship, Young Scientist Award winners receive and an all-expense-paid trip to the International Youth Science Seminar at the Nobel Prize ceremonies in Stockholm.

The largest award presented by a professional association at ISEF is the $10,000 IEEE President's Scholarship, which this year went to Elena Glassman, a Computer Society individual winner in 2003.

In 2005, the Intel International Science and Engineering Fair moves to Phoenix. Further information is available at www.sciserv.org/isef/.■

Editor: Bob Ward, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; bnward@computer.org

Pennsylvania Student Wins IEEE President's Scholarship with Neural Interface Design

Elena Leah Glassman, a 17-year-old high school senior from Doylestown, Pennsylvania, received the $10,000 IEEE Presidents' Scholarship at the 2004 Intel International Science and Engineering Fair in Portland, Oregon. She also received a $3,000 first-place award for computer science from the Intel Foundation, an $8,000 Office of Naval Research scholarship, a $1,000 award from the National Anti-Vivisection Society, and a $500 second-place award from the ACM. Glassman was a 2003 IEEE Computer Society individual award winner at ISEF.

A team of IEEE judges selected Glassman's project, "Brain-Computer Interface for the Muscularly Disabled," from a field of more than 1,200 entries. The $10,000 IEEE scholarship, which recognizes an outstanding achievement in information technology or electrical engineering, is the largest award given by a special-interest professional association at ISEF.

The "Brain-Computer Interface for the Muscularly Disabled" computer program measures and interprets EEG signals from a user, replacing mouse and keyboard inputs. The program is designed to enable a user, wearing a special cap that detects brain waves, to direct the movement of a computer's cursor by thinking about it. Glassman's key to success was devising an algorithm that could predict a user's intended movements based on captured EEG signals. In preparation for ISEF 2004, she used herself as a test subject, collecting EEG wavelets through electrodes on her scalp.

Glassman, who will attend MIT in the fall, has applied for a US patent based upon her work.

2004 IEEE President's Scholarship winner Elena Glassman accepts honors from IEEE President-Elect W. Cleon Anderson.

CRA Releases Taulbee Survey Numbers for 2002-2003

In each of the past 33 years, the Computing Research Association (CRA) has conducted a survey to document trends in student enrollment, employment of graduates, and faculty salaries in computer science and computer engineering.

Survey results this year indicate a drop in undergraduate enrollments and less growth in computer science and engineering departments across North America than expected. The combined annual PhD graduation rate for these programs fell from a high of 1,100 new degrees in 1992 to the current total of 875 doctoral graduates. In contrast, the number of students entering PhD programs at the responding schools has risen sharply in recent years, from an average of 12.5 per department in 2000-2001 to a record high of 17.5 in 2001-2002. Last year, that number slipped slightly to 16.5.

According to published reports from the CRA, this increase in graduate school enrollment can be attributed in large part to a slow turnaround in the general economy and particularly in the dot-com economy. In keeping with this trend, the number of PhDs who are entering industry has dropped precipitously, while the number of those moving into academia has jumped, reversing a pattern seen throughout the 1990s.

Undergraduate enrollments in computer science and computer engineering programs experienced a steep slide in 2002-2003, with an annual 23,000 newly-declared majors over the last three years giving way to about 17,500 in the most recent survey. Accordingly, the expected production of bachelor's graduates is also projected to fall in 2004.

In the US, faculty salaries increased by between 1.9 and 2.5 percent, depending on tenure status. These increases are less than the 3 percent increase seen last year. Canadian salaries decreased by up to 2 percent over the same year.

The Taulbee survey is named in honor of the late Orrin Taulbee, a University of Pittsburgh researcher who conducted the survey for the CRA until 1984. For a detailed breakdown of the findings of the 2002-2003 Taulbee survey, visit www.cra.org/statistics/.■

Nominations Open for IEEE Division VIII Director-Elect

IEEE Computer Society members are invited to submit nominations for candidates to serve as 2006 IEEE Division VIII director-elect and 2007-2008 Division VIII director.

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

James Isaak currently serves as 2003-2004 IEEE Division VIII director. Gene F. Hoffnagle is serving as 2004-2005 IEEE Division V director.

Submit nominations by **8 October** to Stephen Diamond, Chair, Nominations Committee, IEEE Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1992; s.diamond@computer.org.

Robert W. Bemer, 1920-2004

Bob Bemer, a computer pioneer involved in the creation of the American Standard Code for Information Interchange coding system and Cobol, passed away in June at the age of 84.

Bemer worked on the ASCII coding system during his tenure at IBM, where he contributed the "ESC" (escape) key and the backslash, as well as several other characters not previously used in the programming of computers. Computer innovator Admiral Grace Hopper also credited Bemer with coining the terms Cobol (for Common Business-Oriented Language) and Codasyl (the Conference on Data Systems Languages), a 1960s organization devoted to developing a universal data system language for business.

In the era of large, expensive mainframe computers, it was not cost-effective for a single user to have exclusive access to a machine for interactive use. Bemer, in a 1957 article in *Automatic Control Magazine*, was the first to suggest that multiple users could share a machine via a hardware interrupt that would pause a running process and grant processing time to another user.

The IEEE Computer Society presented Bemer, often referred to as the "Father of ASCII," with its Computer Pioneer Award in 2003. His citation reads, "For meeting the world's needs for variant character sets and other symbols, via ASCII, ASCII-alternate sets, and escape sequences."

To read Bemer's musings on topics ranging from the early days of Fortran to the delights of Paris, visit www.bobbemer.com.

Computer Society Cluster Task Force Merges with Supercomputing Technical Committee

At the June IEEE Computer Society meeting series in Long Beach, California, the Task Force on Cluster Computing (TFCC) and the Technical Committee on Supercomputing Applications (TCSA) merged to form the new Technical Committee on Scalable Computing (TCSC).

The TFCC was originally formed to promote cluster computing research and education and to promote technical standards in the area. Cluster computing task force members considered issues related to the design, analysis, and implementation of cluster systems, especially the design and analysis of distributed architectures and algorithms. Since 1999, annual attendance at TFCC events, which include the International Conference on Cluster Computing (Cluster), the International Symposium on Cluster Computing and the Grid (CCGrid), and the IEEE/ACM International Workshop on Grid Computing (Grid), has grown from 100 to nearly 1,000 participants.

The TCSA was established to investigate applications of classical vector supercomputers, new parallel architectures, and high-end workstations. The TCSA also sponsored the biennial High Performance Computing Conference (HPCC), which drew 7,500 attendees last year.

The IEEE Computer Society's Technical Activities Board (TAB) approved the merger as part of a move to reduce overlapping missions among various working groups throughout the Society. Large-scale events sponsored by both the TFCC and the TCSA are expected to continue under the aegis of the new TCSC. Officers of the newly combined groups will present details of their merger at the November TAB meeting in New Orleans.

For more information on IEEE Computer Society technical committees, councils, and task forces, visit www.computer.org/tab/. ■

CALLS FOR PAPERS

ISMVL 2005, 35th Int'l Symp. on Multiple-Valued Logic, 18-21 May, 2005, Calgary, Canada. Submissions due **1 Nov.** www.enel.ucalgary.ca/ISMVL2005/cfp.html

IPDPS 2005, Int'l Parallel & Distributed Processing Symp., 4-8 Apr., 2005, Denver, Colo. Papers due **8 Oct.** www.ipdps.org/ipdps2005/2005_cfp.htm

ARITH-17, 17th IEEE Symp. on Computer Arithmetic, 27-29 June, 2005, Cape Cod, Mass. Papers due **12 Oct.** http://arith17.polito.it/cfp.html

CALENDAR

SEPTEMBER 2004

6-9 Sept: 3DPVT 2004, 2nd Int'l Symp. on 3D Data Processing, Visualization, & Transmission, Thessaloniki, Greece. www.umiacs.umd.edu/conferences/3dpvt04/

6-10 Sept: RE 2004, 12th IEEE Int'l Requirements Eng. Conf., Kyoto, Japan. www.re04.org/

7-10 Sept: PARELEC 2004, Int'l Conf. on Parallel Computing in Electrical Eng., Dresden, Germany. www.parelec.org/

8-10 Sept: CODES + ISSS 2004, 2nd IEEE/ACM/IFIP Int'l Conf. on Hardware/Software Codesign & System Synthesis, Stockholm. www.ida.liu.se/conferences/codes/

9-10 Sept: MTV 2004, 5th Int'l Workshop on Microprocessor Test & Verification, Austin, Texas. http://dropzone.tamu.edu/MTV/

11 Sept: WSE 2004, 6th Int'l Workshop on Web Site Evolution, Chicago. http://blackbox.cs.fit.edu/~wse2004/

11-17 Sept: ICSM 2004, 20th Int'l Conf. on Software Maintenance (with METRICS 2004, SCAM 2004, and WSE 2004), Chicago. www.cs.iit.edu/~icsm2004/

12-14 Sept: BTW 2004, Board Test Workshop, Loveland, Colo. www.molesystems.com/BTW04/

13-15 Sept: CEC04-East, IEEE Conf. on E-Commerce Technology for Dynamic E-Business, Beijing. http://tab.computer.org/tfec/cec04-east/

14-16 Sept: METRICS 2004, 10th Int'l Symp. on Software Metrics, Chicago. www.swmetrics.org/

15-16 Sept: SCAM 2004, 4th IEEE Int'l Workshop on Source Code Analysis & Manipulation, Chicago. www.brunel.ac.uk/~csstmmh2/scam2004/

15-18 Sept: SCC 2004, IEEE Int'l Conf. on Services Computing, Shanghai. http://conferences.computer.org/scc/2004/

17 Sept: WESS 2004, 9th IEEE Workshop on Empirical Studies of Software Maintenance (with ICSM 2004), Chicago. www.ise.gmu.edu/wess04/

17-19 Sept: STEP 2004, Software Technology & Eng. Practice Workshops (with ICSM 2004), Chicago. www.step2004.uwaterloo.ca/

20-23 Sept: CLUSTER 2004, IEEE Int'l Conf. on Cluster Computing, San Diego, Calif. http://grail.sdsc.edu/cluster2004/

20-24 Sept: EDOC 2004, 8th IEEE Int'l Conf. on Enterprise Distributed Computing, Monterey, Calif. www.edocconference.org/

20-24 Sept: WI-IAT 2004, IEEE/WIC/ACM Int'l Conf. on Web Intelligence & Intelligent Agent Technology, Beijing. www.maebashi-it.org/WI04/

20-25 Sept: ASE 2004, 19th IEEE Int'l Conf. on Automated Software Eng., Linz, Austria. www.ase-conference.org/

26-29 Sept: VL/HCC 2004, IEEE Symp. on Visual Languages & Human-Centric Computing, Rome. http://vlhcc04.dsi.uniroma1.it/

27-29 Sept: ASAP 2004, IEEE 15th Int'l Conf. on Application-Specific Systems, Architectures, & Processors, Galveston, Texas. www.ece.rice.edu/asap2004/

28-30 Sept: COMPSAC 2004, 28th Ann. Int'l Computer Software & Applications Conf., Hong Kong. http://rachel.utdallas.edu/compsac/

28-30 Sept: SEFM 2004, 2nd IEEE Int'l Conf. on Software Eng. & Formal Methods, Beijing. www.iist.unu.edu/SEFM2004/

29 Sept.-3 Oct: PACT 2004, Int'l Conf. on Parallel Architectures & Compilation Techniques, Antibes Juan- les-Pins, France. www.pactconf.org/

OCTOBER 2004

5-7 Oct: MASCOTS 2004, 12th Ann.

Submission Instructions

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting. Complete instructions for submitting conference or call listings are available at www.computer.org/conferences/submission.htm.

A more complete listing of upcoming computer-related conferences is available at www.computer.org/conferences/.

Meeting of the IEEE/ACM Int'l Symp. on Modeling, Analysis, & Simulation of Computer & Telecommunication Systems, Volendam, Netherlands. www.mascots-conf.org/

5-8 Oct: ICNP 2004, 12th IEEE Int'l Conf. on Network Protocols, Berlin. www.icnp2004.de.vu/

6-8 Oct: PG 2004, 12th Pacific Conf. on Computer Graphics & Applications, Seoul. http://graphics.snu.ac.kr/pg2004/

10-15 Oct: VIS 2004, IEEE Conf. on Visualization, Austin, Texas. http://vis.computer.org/vis2004/

11-13 Oct: ICCD 2004, IEEE Int'l Conf. on Computer Design: VLSI in Computers & Processors, San Jose, Calif. www.iccd-conference.org/

11-15 Oct: UML 2004, 7th Int'l Conf. on Unified Modeling Language, Lisbon, Portugal. www.umlconference.org/

13-15 Oct: AIPR 2004, 33rd Applied Imagery Pattern Recognition Workshop, Washington, D.C. www.aipr-workshop.org/

13-15 Oct: NCSBB 2004, N. Carolina Symp. on Biotechnology & Bioinformatics, Raleigh, N.C. http://ewh.ieee.org/cmte/tech4life/

17-19 Oct: FOCS 2004, 45th Ann. IEEE Symp. on Foundations of Computer Science, Rome. www.cs.brown.edu/~aris/focs04/

18-20 Oct: SRDS 2004, 23rd Symp. on Reliable Distributed Systems, Florianópolis, Brazil. www.srds2004.ufsc.br/

20-23 Oct: FIE 2004, Frontiers in Education Conf., Savannah, Ga. www.fie-conference.org/

21-23 Oct: DS-RT 2004, 8th IEEE Int'l Symp. on Distributed Simulation & Real-Time Applications, Budapest. www.cs.unibo.it/DS-RT2004/

25 Oct: WWC, IEEE CS 7th Workshop on Workload Characterization, Austin, Texas. www.ece.utexas.edu/~ljohn/wwc/wwc7/

25-27 Oct: MASS 2004, IEEE Int'l Conf. on Mobile Ad hoc & Sensor Systems, Ft. Lauderdale, Fla. www.ececs.uc.edu/~cdmc/mass/

26-28 Oct: ITC 2004, Int'l Test Conf., Charlotte, N.C. www.itctestweek.org/

31 Oct.-3 Nov: ISWC 2004, 8th IEEE Int'l Symp. on Wearable Computers, Arlington, Va. www.cc.gatech.edu/ccg/iswc04/

NOVEMBER 2004

1-4 Nov: ICDM 2004, 4th IEEE Int'l Conf. on Data Mining, Brighton, UK. http://icdm04.cs.uni-dortmund.de/

2-5 Nov: ISMAR 2004, 3rd IEEE and ACM Int'l Symp. on Mixed and Augmented Reality, Arlington, Va. www.ismar04.org/

2-5 Nov: ISSRE 2004, 19th Int'l Symp. on Software Reliability Eng., Saint-Malo, France. www.issre.org/2004/

6-12 Nov: SC 2004, High-Performance Computing, Networking, & Storage Conf., Pittsburgh. www.sc-conference.org/sc2004/

7-11 Nov: ICCAD 2004, IEEE/ACM Int'l Conf. on Computer Aided Design, San Jose, Calif. www.iccad.com/

10-12 Nov: HLDVT 2004, IEEE Int'l High-Level Design Validation & Test Workshop, Sonoma, Calif. www.hldvt.com/04/

11-12 Nov: WRTLT 2004, 5th Workshop on RTL & High-Level Testing, Osaka, Japan. http://wrtlt04.su.cit.nihon-u.ac.jp/~wrtlt/

15-17 Nov: ATS 2004, 13th Asian Test Symp., Kenting, Taiwan. http://ats04.ee.nthu.edu.tw/~ats04/

15-17 Nov: ICTAI 2004, 16th IEEE Int'l Conf. on Tools with AI, Boca Raton, Fla. www.cse.fau.edu/~ictai04/

15-19 Nov: AISTA 2004, Int'l Conf. on Advances in Intelligent Systems: Theory & Applications, Kirchberg, Luxembourg. http://aista.tudor.lu/aista2004/index.html

16-18 Nov: LCN 2004, 29th IEEE Conf. on Local Computer Networks, Tampa, Fla. www.ieeelcn.org/

DECEMBER 2004

2-3 Dec: WMCSA 2004, 6th IEEE Workshop on Mobile Computing Systems & Applications, Cumbria, UK. http://wmcsa2004.lancs.ac.uk/

4-8 Dec: MICRO-37, 37th Ann. IEEE/ACM Int'l Symp. on Microarchitecture, Portland, Ore. www.microarch.org/micro37/

13-15 Dec: MSE 2004, 6th Int'l Symp. on Multimedia Software Eng., Miami, Fla. www.cs.fiu.edu/MSE04/

18-21 Dec: ISSPIT 2004, 4th IEEE Symp. on Signal Processing & Information Technology, Rome. www.isspit.org/isspit/2004/index.html

19-22 Dec: HiPC 2004, 11th Int'l Conf. on High-Performance Computing, Bangalore, India. www.hipc.org/

JANUARY 2005

3-6 Jan: AICCSA 2005, 3rd ACS/IEEE Int'l Conf. on Computer Systems & Applications, Cairo. http://engr.smu.edu/cse/AICCSA-05/

31 Jan.-4 Feb: SAINT 2005, Int'l Symp. on Applications & the Internet, Trento, Italy. www.saint2005.org/

UNIVERSITY OF OTAGO. Lecturer/ Senior Lecturer, University of Otago, Dunedin, New Zealand The Department of Information Science invites applicants with teaching and research experience in telecommunications, computer networking or network security for a confirmation path position (PhD required). For more information, see http://www.business.otago.ac.nz/infosci/vacancies.htm

PROGRAMMER ANALYST. Plan, develop and test computer programs. BS in Comp Sci or equiv req. Send ad w/resume to: Logic House, 3888 S. Iroquois Lane, Gold Canyon, AZ 85218.

CONSULTANT (Unisys/Tallahassee). Design, develop & implement solutions for public sector clients particularly in area of public assistance & child support & enforcement. Work done principally in IBM mainframe environment. Responsibilities incl on-line & batch application programming using TELON, COBOL, IMS DB/DC. Also responsible for IBM JCL creation, modification & maintenance. Reqs. Bachelors Degree or equiv in engr'g or closely related field. 3 yrs exp in job offered or 3 yrs exp as Computer Consultant/Systems Analyst. Must have good working knowl of JCL & at least 3 yrs of exp working in IBM mainframe environment & with IMS DB/DC, COBOL & on-line application programming in public sector/public assistance domain. 40 hrs/wk - 8:30 to 5:30. $70,250/yr. Send resume to IEEE Computer Society, 10662 Los Vaqueros Circle, Box# COM4, Los Alamitos, CA 90720.

SUBMISSION DETAILS: Rates are $275.00 per column inch ($300 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: mander son@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..

ADVERTISER / PRODUCT INDEX AUGUST 2004

Advertising Personnel

Marion Delaney
IEEE Media, Advertising Director
Phone: +1 212 419 7766
Fax: +1 212 419 7589
Email: md.ieeemedia@ieee.org

Marian Anderson
Advertising Coordinator
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: manderson@computer.org

Sandy Brown
IEEE Computer Society,
Business Development Manager
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: sb.ieeemedia@ieee.org

Advertising Sales Representatives

Mid Atlantic (product/recruitment)
Dawn Becker
Phone: +1 732 772 0160
Fax: +1 732 772 0161
Email: db.ieeemedia@ieee.org

New England (product)
Jody Estabrook
Phone: +1 978 244 0192
Fax: +1 978 244 0103
Email: je.ieeemedia@ieee.org

New England (recruitment)
Robert Zwick
Phone: +1 212 419 7765
Fax: +1 212 419 7570
Email: r..zwick@ieee.org

Connecticut (product)
Stan Greenfield
Phone: +1 203 938 2418
Fax: +1 203 938 3211
Email: greenco@optonline.net

Midwest (product)
Dave Jones
Phone: +1 708 442 5633
Fax: +1 708 442 7620
Email: dj.ieeemedia@ieee.org

Will Hamilton
Phone: +1 269 381 2156
Fax: +1 269 381 2556
Email: wh.ieeemedia@ieee.org

Joe DiNardo
Phone: +1 440 248 2456
Fax: +1 440 248 2594
Email: jd.ieeemedia@ieee.org

Southeast (recruitment)
Jana Smith
Phone: +1 404 256 3800
Fax: +1 404 255 7942
Email: jsmith@bmmatlanta.com

Midwest/Southwest (recruitment)
Darcy Giovingo
Phone: +1 847 498-4520
Fax: +1 847 498-5911
Email: dg.ieeemedia@ieee.org

Southwest (product)
Josh Mayer
Phone: +1 972 423 5507
Fax: +1 972 423 6858
Email: josh.mayer@wageneckassociates.com

Northwest (product)
Peter D. Scott
Phone: +1 415 421-7950
Fax: +1 415 398-4156
Email: peterd@pscottassoc.com

Southern CA (product)
Marshall Rubin
Phone: +1 818 888 2407
Fax: +1 818 888 4907
Email: mr.ieeemedia@ieee.org

Northwest/Southern CA (recruitment)
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Southeast (product)
Bob Doran
Phone: +1 770 587 9421
Fax: +1 770 587 9501
Email: bd.ieeemedia@ieee.org

Japan
Sandy Brown
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: sbrown@computer.org

Europe (product/recruitment)
Hilary Turnbull
Phone: +44 1875 825700
Fax: +44 1875 825701
Email: impress@impressmedia.com

D. E. Shaw Research and Development

Systems Architects and ASIC Engineers

Specialized Supercomputer for Computational Drug Design

Extraordinarily gifted systems architects and ASIC design and verification engineers are sought to participate in the development of a special-purpose supercomputer designed to fundamentally transform the process of drug discovery within the pharmaceutical industry. This early-stage, rapidly growing project is being financed by the D. E. Shaw group, an investment and technology development firm with approximately US $8 billion in aggregate capital. The project was initiated by the firm's founder, Dr. David E. Shaw, and operates under his direct scientific leadership.

This project aims to combine an innovative, massively parallel architecture incorporating 90-nanometer "system on a chip" ASICs with novel mathematical techniques and groundbreaking algorithmic advances in computational biochemistry to direct unprecedented computational power toward the solution of key scientific and technical problems in the field of molecular design. Successful candidates will be working closely with a number of the world's leading computational chemists and biologists, and will have the opportunity not only to participate in an exciting entrepreneurial venture with considerable economic potential, but to make fundamental contributions within the fields of biology, chemistry, and medicine.

The candidates we seek will be unusually intelligent and accomplished, with a demonstrated ability to design and implement complex, high-performance hardware solutions based on the latest semi-custom technologies. We are prepared to reward exceptionally well-qualified individuals with above-market compensation.

Please send resume, along with GPAs, standardized test scores (SAT, GRE), and compensation history, to career9@desrad.deshaw.com.

D. E. Shaw Research and Development, L.L.C. does not discriminate in employment matters on the basis of race, color, religion, gender, national origin, age, military service eligibility, veteran status, sexual orientation, marital status, disability, or any other protected class.

DE Shaw & Co

NTT DoCoMo

DoCoMo USA Labs

World Class Opportunities at DoCoMo USA Labs

DoCoMo Communications Laboratories USA, Inc.

DoCoMo USA Labs is part of the world's leading mobile communications company, NTT DoCoMo, the premier wireless service provider in Japan and an innovator recognized throughout the world of mobile communications. NTT DoCoMo launched the world's first 3G mobile service in 2001 and operates the world's largest mobile internet service, i-mode. The company conducts its Research and Development at Yokosuka Research Park in Tokyo and at three global R&D locations, San Jose USA, Munich Germany and Beijing China.

DoCoMo USA Labs is located in Silicon Valley where we are conducting research into advanced operating systems, wireless applications, networks, protocols and media. The laboratory is organized into four distinct labs: the Media Lab (ML), the Mobile Software Lab (MSL), the Network Architecture Lab (NAL), and the Network Service and Security Lab (NSSL). The four labs are unified by the common goal of providing breakthroughs in mobile Internet technologies and future platform definition.

This is your opportunity to be part of the future and pursue your dreams at DoCoMo USA Labs. We are expanding and seeking suitably qualified applicants for the following positions.

Network Architecture Lab (nal@docomolabs-usa.com)

- **Wireless Networking Project Manager (ref. NAL1s)**
  Address fundamental research in wireless networks, protocols and systems. IP-based RAN and core network design including routing and addressing. MAC layer and mobility management and security.

Mobile Software Lab (msl@docomolabs-usa.com)

- **Mobile Operating Systems Project Manager (ref. MSL1s)**
  Lead research into new operating systems principles and concepts. Topics include memory management, scheduling, file systems, and communications.
- **Secure Languages Researcher (ref. MSL3s)**
- **Operating System Researcher in Kernel Design and Implementation (ref. MSL4s)**
- **Middleware Researcher (ref. MSL6s)**

Network Service and Security Lab (nss@docomolabs-usa.com)

- **Mobile Network Security Senior Researcher (ref. NSSL3s)**
- **Mobile Network Security Researcher (ref. NSSL4s)**
- **Mobile Middleware & Web Services (ref. NSSL2s)**

Media Lab (mm@docomolabs-usa.com)

- **Mobile Multimedia Researcher (ref. MM1s)**
  Speech and audio coding, video coding, non-linear signal processing and delivery of multimedia content over IP networks.

Qualifications and Experience

Applicants are expected to have a Ph.D. or equivalent in C. S., E.E., or C.E. Applications are invited from candidates with all levels of experience. A track record of research is expected for senior positions.

To Apply

Please apply by emailing your resume to the email address shown after each Lab's name and be sure to include job title and job reference number. Example: if you are applying for Mobile Operating Systems Project Manager (ref. MSL1s) you would email to msl@docomolabs-usa.com and put Project Manager / ref. MSL1s in the subject title of the email.

For further information, visit the careers section on our website

www.docomolabs-usa.com

An Equal Opportunity Employer

Achieving Software Quality Through Teamwork, Isabel Evans. Successful software depends not only on technical excellence but on how members of the software team work together. The author provides an overview of the team culture required to develop quality software.

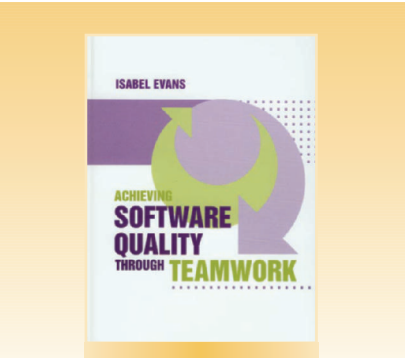
Reflecting the different views on the nature of software quality, the book offers methods for helping participants in a software team communicate more effectively and overcome the conflict created by their different perceptions of quality. Readers learn the roles and activities of team members, including customers, throughout a software product's life, from before development starts to after the software has been deployed.

The author describes popular software quality models such as EFQM, Watts Humphreys' personal and team software processes, TMAP, and CMM. The book also includes numerous examples, sources for further information, and tools and techniques that can be applied to real projects. Defining the key groups within a software team and their different definitions of quality can help readers improve the communications and relationships between team members throughout a project and the software's lifetime.

Artech House; www.artechhouse.com; 1-58053-662-X; 324 pp.; $79.00

Formal Engineering for Industrial Software Development: Using the SOFL Method, Shaoying Liu. Formal methods involve using mathematical notation and calculus in software development. Yet applying such methods to large-scale systems with practical constraints such as limited developer skills, time and budget restrictions, and changing requirements is difficult.

The author claims that formal engineering methods may bridge this gap. He advocates the incorporation of



mathematical notation into the software engineering process, thus substantially improving the rigor, comprehensibility, and effectiveness of the methods commonly used in industry.

This book provides an introduction to the Structured Object-Oriented Formal Language (SOFL) method that the author has designed and industry-tested. Written in a style suitable for lecture courses or professional use, the book contains many exercises and a significant real-world case study, providing readers with knowledge and examples needed to successfully apply the author's method to their own projects.

Springer; www.springeronline.com; 3-540-20602-7; 408 pp.; $79.95

Algorithms and Data Structures: The Science of Computing, Douglas Baldwin and Gregory W. Scragg. This book focuses on three core topics: design, the architecture of algorithms; theory, mathematical modeling and analysis; and the scientific method, experimental confirmation of theoretical results.

Throughout the book, the authors strive to help students see that computer science is about problem solving, not simply the memorization and recitation of languages. They take an integrated approach to explaining the methods of inquiry so that students can see explicitly how these methods interact. The authors emphasize recursion and object-oriented programming as the main control structure and abstraction mechanism, respectively, in algorithm design.

This book reflects that computer science is not solely about learning how to speak in a programming language. Using an approach slanted especially toward students, the authors cover recursion, binary trees, stacks, queues, hash tables, and object-oriented algorithms. The accompanying Web site includes lab exercises, code, and instructor's notes.

Charles River Media; www.charlesriver.com; 1-58450-250-9; 620 pp.; $59.95

The Art of Assembly Language, Randall Hyde. This comprehensive book presents assembly language from the high-level programmer's point of view so that readers can start writing meaningful programs within days. Readers can use the High Level Assembler (HLA) that accompanies the book to write portable assembly language programs that run under either Linux or Windows with nothing more than a recompile.

The accompanying CD-ROM includes the HLA and HLA Standard Library, all the source code from the book, and more than 50,000 lines of additional sample code, all documented and tested. The code compiles and runs as is under Windows and Linux.

No Starch Press; www.nostarch.com; 1-886411-97-2; 928 pp.; $59.95

Editor: Michael J. Lutz, Rochester Institute of Technology, Rochester, NY; mikelutz@mail.rit.edu. Send press releases and new books to *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; fax +1 714 821 4010; newbooks@computer.org.

ReSharper Program for Visual Studio .NET 2003

JetBrains recently announced an access program for JetBrains ReSharper, the company's add-in tool for developers using Visual Studio .NET 2003. The ReSharper integrated plug-in is designed to enhance Visual Studio with intelligent C# code-editing assistance and many productivity enhancements from the company's IntelliJ Idea 4.0.

New ReSharper features include integration with Visual Studio, eliminating any need to switch to an outside application, plus the capability to jump from a method to its declaration or from a method in a base class or interface to its implementation and back.

Contact the company for licensing information; www.jetbrains.com.

DSP Development Tool Supports Matlab 7 and Simulink 6

Xilinx announced a new version of the System Generator for DSP development tool that supports the Matlab 7 and Simulink 6 software packages, recently released as parts of MathWorks 14.

According to Xilinx, the latest version of System Generator provides performance and database enhancements designed to make it easier for developers to build DSP systems with Virtex Pro and Spartan-3 series FPGAs.

The System Generator for DSP tool is used with the ISE 6.2i FPGA design suite from Xilinx. It costs $1,995; www.xilinx.com.

Nios 2 IDE Based on Eclipse Platform

Altera Corporation has announced Nios 2, a development environment based on the open Eclipse platform. This platform has become one of the preferred development frameworks among embedded tool providers such as Wind River and QNX.

The Nios 2 IDE provides a C/C++ development suite that includes a text editor, project manager, build tools, debugger, and CFI-compatible flash programmer. The debugger connects to a range of targets, including FPGA hardware, the Nios 2 Instruction Set Simulator, and the ModelSim-Altera software.

Nios 2 is available with each Nios 2 development kit, including Stratix, Stratix Pro, and Cyclone editions. All active Nios subscribers will automatically receive Nios 2 IDE as part of their Nios 2 update. Subscriptions cost $495 annually; www.altera.com.

Sun Releases Java Desktop System 2

To compete more effectively with Microsoft in enterprise environments, Sun Microsystems released the Sun Java Desktop System 2, which now offers integrated, centralized management functionality specifically designed for administrators working with large networks. The new operating system, based on Linux, provides auto-update and management capabilities plus an IDE for Linux and Java developers that includes Sun Java Studio.

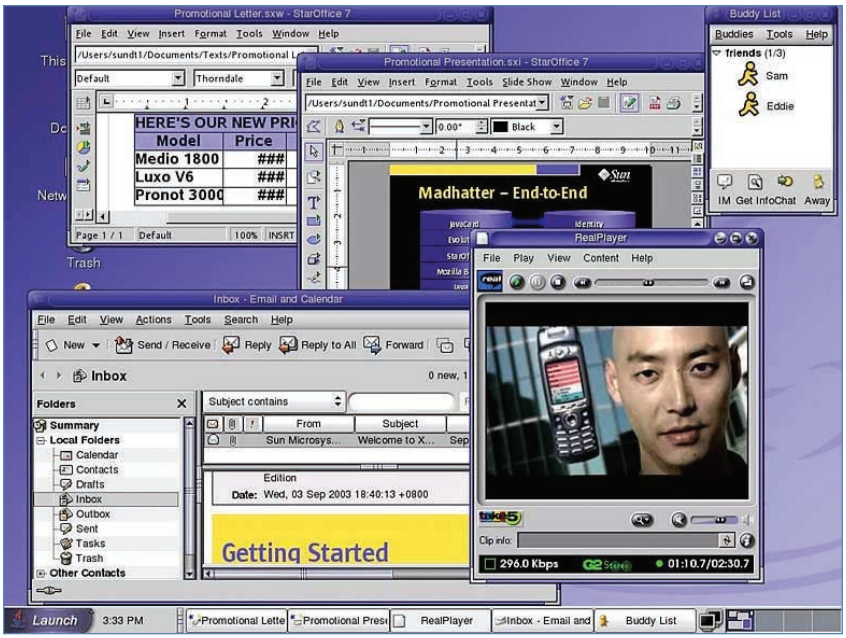
Sun's Java Desktop System 2 is available at 50 percent off its original price at an annual $50 per desktop or $25 per employee until 2 December 2004. Buyers can purchase the Java Desktop System through Sun, its iForce partners, or Wal-Mart on Microtel hardware; www.sun.com/javadesktopsystem.

ST7 Microcontroller Development Software Upgraded

STMicroelectronics has released an upgrade to its STVD7 development software, which supports the ST7 family of 8-bit microcontrollers.

ST7 Visual Develop 3.0 adds build and programming features to the debug tools of earlier versions. Users can select a specific ST7 microcontroller unit for their project at the build stage and then tailor and optimize their application code, taking into account the MCU's memory map, pin-out, and registers. The upgrade integrates the ST7 Assembler and Linker toolsets, as well as C compilers from Cosmic and Metrowerks. Errors are automatically highlighted in source files.

The software is available as a free



Sun's Java Desktop System 2 update offers integrated management tools to help set policies and configurations for groups and automatically distribute system updates and patches across a network to all Java Desktop Systems from a central location.

Please send new product announcements to products@computer.org.

download for ST7 developers from http://mcu.st.com.

AltiWare 5.0 IP Telephony Platform Released

AltiGen Communications released AltiWare 5.0, the next iteration of its software platform supporting AltiServ IP telephony and call-center systems. Version 5.0 reportedly offers significant enhancements for every aspect of the AltiGen system, including call center, distributed communications, voice over IP, general telephone systems functionalities, management and reporting, and integration capabilities. AltiGen has also enhanced the overall look and feel of all administrative and desktop interfaces.

Contact the company for licensing information; www. altigen.com.

Application Framework Based on JRules

ILOG has released Business Rule Studio, an application framework based on JRules—a key component of the company's Business Rule Management System product line.

Built on the Eclipse IDE, BR Studio gives Java developers rule-authoring, testing, and debugging tools for their applications in that environment. BR Studio is available for free by Web download and requires Eclipse 2.1 or WebSphere Application Developer 5.1.

Deployment of business rule applications requires separate deployment licenses of the full version of ILOG JRules, which are sold separately; www.ilog.com.

System Architect Activity-Based Methodology

Popkin Software announced that System Architect now supports an *activity-based methodology*. ABM automates enterprise architecture development for projects using the Department of Defense Architecture Framework (DoDAF). According to Popkin, System Architect is the first tool to offer support for ABM, a methodology created in part by MITRE Corp.

The System Architect technology is designed to help architecture teams get started using DoDAF by providing predefined and automated guidelines that support information capture and the generation of views for analysis and reporting. System Architect provides a maintainable set of standard reference objects and a set of additional DoDAF products and reports.

Contact the company for licensing information; www.popkin.com.

Next-Gen Connectivity Processors for Telematics

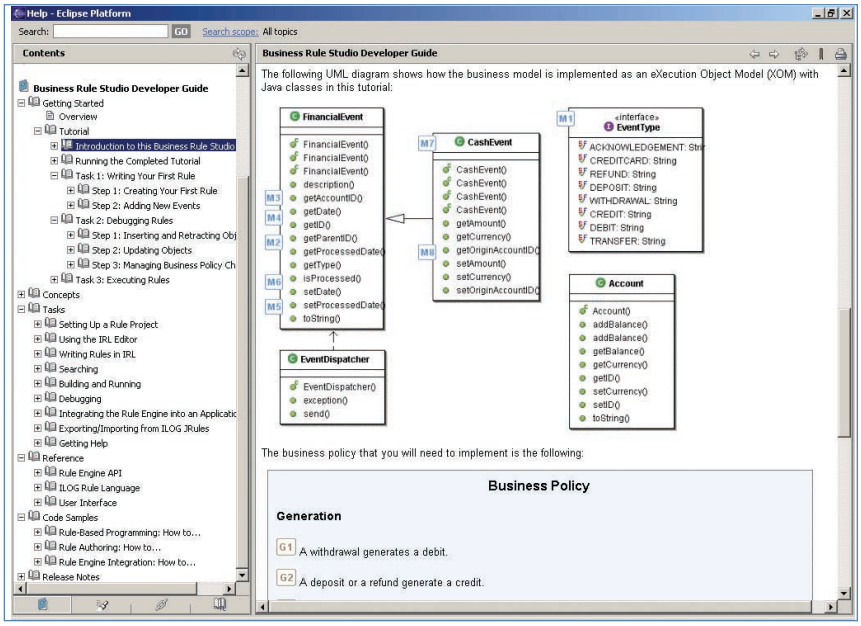
National Semiconductor has announced four new CP3000 processors designed specifically to connect different types of devices in automobiles via wide area, local area, or personal area networks.

National's new processors integrate Bluetooth wireless technology, controller area networks, and USB interfaces with additional software to accelerate development of telematics applications such as hands-free communication devices. For wireless applications, the CP3000 processors include a fully embedded Bluetooth interface with baseband processing, a host protocol stack, and dedicated profiles.

The new processors are available now for roughly $10 per processor in OEM quantities; www.national.com/appinfo/cp3000.

Linux Development Tools from Metrowerks

Metrowerks announced commercial-grade Linux development tools designed to help with board bring-up, kernel-level debugging, driver development, and application creation—all within a Linux-hosted environment. The CodeWarrior development studios are designed to complement Metrowerks' existing lineup of hardware, board-support package, and code coverage tools to offer a collection of Linux-development applications. The suites include a debugger that delivers the capabilities required to target a Linux OS to a new hardware platform, including driver development; www.metrowerks.com.



Business rule technology can make applications more agile, capturing business logic as embedded rules. But the learning curve for creating such applications can be steep. ILOG Business Rule Studio integrates ILOG's JRules engine within the Eclipse environment so developers can write, debug, and execute rules separately from Java code.

Power Management in Networked Devices

Ken Christensen, University of South Florida
Bruce Nordman and Rich Brown, Lawrence Berkeley National Laboratory

Networks are one of the most significant developments in computing and a hallmark of modern society. However, along with increasing efficiency and productivity, both at home and in the workplace, networks have costs. One cost is the additional energy that electronic devices consume when attached to networks.

Recent studies indicate that the economic and environmental impact of such energy use is substantial (www.eere.energy.gov/buildings/documents/pdfs/office_telecom-vol1_final.pdf). In 1999, for example, office and network equipment—primarily PCs connected to the Internet—accounted for about 2 percent of US electricity consumption (http://enduse.lbl.gov/Info/LBNL-45917b.pdf). This roughly corresponds to $6 billion in energy costs and the emission of more than 50 million metric tons of carbon dioxide per year.

During the next two decades, energy consumption by digitally networked devices is expected to increase at a faster rate than other types of energy use in buildings. Many such devices will not be PCs but emerging technologies designed for the home rather than the office.

POWER MANAGEMENT

Power management, a standard feature of modern PCs, was primarily developed to increase battery lifetime in laptop PCs, which historically were not network-connected when using battery power. Today, however, many laptops are connected to a network—typically a Wi-Fi network—as are the majority of desktop computers.

The Advanced Configuration and Power Interface (www.acpi.info)—an open industry specification jointly developed by Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba—defines standard power management interfaces for modern personal computers. PCs using Microsoft Windows XP and other current operating systems can enter a sleep state that typically decreases overall power consumption by more than 90 percent.

However, low-power sleep modes usually are not compatible with network connectivity. Consequently, in about 95 percent of desktop PCs, power management is disabled, either during machine setup or the first time that the feature interferes with a network application. Display power management is enabled on most PCs, but tens of millions more could readily be enabled.

We estimate that enabling greater use of existing power management features in computer systems would, within a few years, save from 11 to 36 terawatt hours—equal to $0.8 to $2.7 billion—per year. However, achieving these savings without disrupting system performance will require making network devices, applications, and protocols more "energy aware."

DRIVERS OF ENERGY USE

Three key drivers of energy use are *induced consumption* by devices prevented by network connections from entering low-power states, *increasing link data rates* that inherently consume more energy for the network interfaces, and proliferation of *network-connected displays* that actively update and display data when no one is present.

Making networked devices "energy aware" could save more than $1 billion per year.

Induced consumption

Most desktop PCs are continuously left fully powered even though there is no user demand for their resources most of the time. This is largely because such resources are increasingly shared and must thus be accessible by remote users and other computers 24/7. Companies customarily back up files and update software at night, while many distributed applications, including server database applications with client front ends, require full-time TCP connections.

A number of existing and emerging protocols, including Universal Plug and Play, require consumer electronic

devices to respond to routine "housekeeping" messages, for tasks such as maintaining IP addresses and routing tables, or they risk disappearing from the network. Network protocols designed to provide security or protect content also can increase device ontimes.

Existing Ethernet network interface controllers, which can operate on auxiliary power when a PC is in a low-power sleep state, offer one possible solution to the problem of induced consumption. NICs can awaken the main system when they receive a packet defined for this purpose or specific types of ordinary IP packets.

Adding proxy capabilities could enable NICs to directly respond to most low-level protocol messages, triggering a wakeup only when handling a message that requires the main system's resources. This would require a small general-purpose processor for the NIC and new protocols to divide responsibilities between the system and the controller during low-power sleep states. The concept of proxying for power management—adding new hardware and software to a system to reduce overall power consumption—requires further research.

Increasing network link data rates

Link data rates for wired desktop computers connected to the Internet have increased from 10 Mbps to 1 Gbps, which further strains computing system resources. Within a few years, 10-Gbps NICs, which some servers currently use, may be standard in desktop PCs. A 10-Mbps NIC consumes on the order of 1 watt, while a 10-Gbps NIC consumes tens of watts.

Reducing the network link data rate for PCs in low-power states would not affect user productivity. Ethernet autonegotiation mechanisms already exist that allow the link data rate to change. However, new methods are needed to modulate link data rates with traffic levels to scale energy consumption with actual service demand. For example, NICs could be designed to have staged functionality and power states similar to the main PC system. Collectively applying this strategy could potentially save nearly 1 terawatt hour per year in US personal computers alone.

Display proliferation

The rapid transition from CRT to LCD technology, combined with increasingly easy (and often wireless) networking, makes it feasible to utilize multiple displays—each consuming tens of watts—throughout homes and commercial buildings. LCDs are replacing previously unpowered displays in many applications. This large number of displays, hosts, and network interfaces could significantly increase energy consumption.

Researchers must adapt power management efforts to the Internet and the devices that connect to it.

Although power management is enabled in most PC displays, research is needed to develop communication protocols for evolving display usage scenarios and increasingly sophisticated interfaces—among devices as well as users—that facilitate more robust power management. Displays should be asleep or off when not needed; at a minimum, the network processor and interface should be in a low-power state except when explicitly updating display data.

CURRENT RESEARCH

Next-generation wireless devices and sensor networks drive much of the ongoing work on power management. The proliferation of network applications and central data stores has raised concerns about energy use, power infrastructure costs, reliability, and cooling of data centers.

Researchers are developing new network protocols and standards for sharing files, routing packets, and so on to better match server resources to the load in real time. They are also working on methods to reduce heat in processors and lengthen the battery life of the portable devices that use them. In addition, new technologies such as magnetic RAM will enable users to fully power up a PC in hundreds of milliseconds rather than the few seconds required today.

However, to address the increasing energy problems related to networks, developers and designers must explicitly adapt power management efforts to the Internet and the devices that connect to it. Indeed, failure to do so could undo much of the success of Energy Star (www.energystar.gov), a US government labeling program that promotes energy-conscious purchasing by consumers and encourages manufacturers to build more energy-efficient products.

Despite the billions of dollars in potential savings from improved power management, particularly among the growing number of home-networked non-PC devices, little research is going on in this area. Network-related consumption is a new factor as a large consumer of electricity, which helps explain the lack of attention. In addition, industry has many other issues to grapple with that more directly affect sales and profitability. Thus far, the federal government has not filled the void.

The University of South Florida's Green TCP/IP Project (www.csee.usf.edu/~christen/greentcp/main.html) and the Energy End-Use Forecasting group at Lawrence Berkeley National Laboratory in California (http://enduse.lbl.gov/Projects/InfoTech.html) are currently exploring this issue. Maruti Gupta and Suresh Singh at Portland State University also addressed the "greening of the Internet" at the ACM SIGCOMM 2003 conference (www.cs.pdx.edu/~singh/ftp/sigcomm03.pdf).

The IEEE is currently balloting IEEE 1621, a proposed new international standard for power control user interfaces designed to make it easier to enable and use power management in office and consumer electronic devices.

In addition, commercial products that globally control power management settings of desktop PCs are emerging (http://eetd.lbl.gov/Controls/1621). Two companies offering such products are Verdiem in the United States (www.verdiem.com) and 1E in the United Kingdom (www.1e.com).

The evolution of network hardware, software, and applications shows no sign of slowing down. We believe that only an ongoing government-backed research program can realize the full potential of power management in office and consumer equipment. In addition to addressing the research needs outlined here, this would entail uncovering and documenting other major network-related energy problems as well as reviewing major network protocols, both current and in development, to determine how best to incorporate power management.

Government funding of proof-of-concept research could yield a significant return on a very small investment and also help get the attention of industry, which is critical to implementing these solutions. ■

Ken Christensen is an associate professor in the Department of Computer Science and Engineering at the University of South Florida. Contact him at christen@csee.usf.edu.

Bruce Nordman is a principal research associate in the Environmental Energy Technologies Division at Lawrence Berkeley National Laboratory. Contact him at bnordman@lbl.gov.

Rich Brown is a scientist in the Environmental Energy Technologies Division at Lawrence Berkeley National Laboratory. Contact him at rebrown@lbl.gov.

Editor: Upkar Varshney, Department of CIS, Georgia State University, Atlanta; uvarshney@gsu.edu

"Patch on Demand" Saves Even More Time?

Angelos D. Keromytis, Columbia University

In the June 2004 Security column ("A Patch in Nine Saves Time?" pp. 82-83), Bill Arbaugh makes two interesting observations: first, whoever has the tightest observe-orient-decide-act (OODA) loop will prevail in a confrontation; second, the infection rates of recent worms suggest that the good guys are losing the battle.

Arbaugh offers some sensible suggestions to vendors and security professionals on improving patch management. However, the best indication that we are losing the battle is not the infection rates of worms such as Slammer and Blaster, but the shrinking interval between discovering and announcing a new vulnerability and the appearance of a worm or attack that exploits it.

The most recent example is the Witty worm, which effectively exploited a vulnerability present in a small population of hosts—approximately 12,000 computers. Although it was first discovered on 8 March 2004, the vendor didn't announce the vulnerability until 18 March, after it had made a patch available. A little over a day after the announcement, Witty made its first appearance.

ZERO-DAY ATTACKS

Given such a short turnaround time, we can reasonably expect to soon experience a *zero-day* worm. Zero-day attacks are those for which users receive no prior warning and thus have no preventive measures in place.

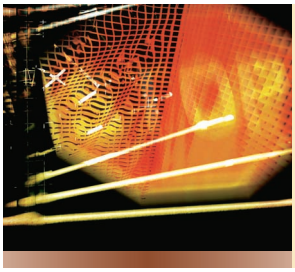
To date, a combination of aggressive packet filtering and proactive application patching could—at least in principle—defeat all the worms we have encountered. Although we could, in theory, deploy patches and network filters automatically, the practicality of employing such measures and their effect on regular system operation are an entirely different story.

Witty came close to being a zero-day worm; for most organizations, it was. Few system administrators had even seen the announcement before the attack, much less downloaded and installed the necessary software patch.

Furthermore, as Arbaugh ("Windows of Vulnerability: A Case Study of Analysis," *Computer*, Dec. 2000, pp. 52-59) and others such as Eric Rescorla ("Security Holes … Who Cares?" *Proc. 12th Usenix Security Symp.*, Usenix, 2003, pp. 75-90) have noted, many administrators find it impractical, if not otherwise unacceptable, to patch or upgrade their systems when a vulnerability is announced. Instead, they wait for news of an actual exploitation. In many cases, this is simply too late.

PATCH ON DEMAND

What to do then? One new idea is to integrate the vulnerability discovery, patch generation, and patch application cycles into a system that would automatically detect a new attack, analyze its modus operandi, determine the best software patch, and apply it at the desired level of granularity—LAN, enterprise, or Internet-wide.



A vaccination system could automatically generate patches to protect an application's source code.

Although the system would still function by reacting to attacks, its response time would be significantly shorter. Perhaps most importantly, it could operate autonomously.

Furthermore, by retaining a focus on software patching, this approach could avoid at least some pitfalls of network-based defense techniques. These techniques, which include packet and content filtering, can fall prey to worms that exploit opportunistic encryption, polymorphism, or metamorphism. Traditional viruses use polymorphism to make detection more difficult: A small decoder, which changes periodically, decrypts the virus's main body prior to execution.

Metamorphic viruses, on the other hand, completely translate the virus code to use different but equivalent instructions every time it infects a new target. Classic signature-matching techniques are, for the most part, incapable of detecting such viruses.

Metamorphism, in particular, makes detecting new viruses extremely diffi-

cult and time-consuming. Worse, several advanced metamorphic virus engines, such as Zmist, have been publicly available for a few years. I suspect that their use in a future worm is only a matter of time. One reason we have not seen them yet may be that current practices are so effective in creating highly infectious worms.

Vaccination system

Can we develop a patch-on-demand system that is both practical and safe?

This is the question we are investigating at Columbia University's Network Security Lab (http://nsl.cs.columbia.edu/). We believe that some of the necessary technology, at least for the initial steps, is already available.

To begin with, we think it is much easier to identify a previously unknown vulnerability by observing the actions of an attacker, such as a worm, that must "demonstrate" its use (and side effects) while attacking a new system.

Figure 1 shows an automated worm vaccination system that includes a *honeypot*—a security resource that pretends to be an easy target for purposes of attracting and exposing attackers. Using honeypots for early-warning systems has increased over the past few years. A properly instrumented vaccination system could monitor the attack activity at the software level and detect previously unknown attacks against what the adversary considers a valid target for infection.

As the worm attempts to infect the system, it reveals not only the vulnerability's existence but also its precise characteristics. For example, in the case of buffer overflows, a trace of the stack shows up at the time of the attack, along with identification of the routine where the overflow occurred, the vulnerable buffer, and the attack vector.

Developers could use this system to get the information needed to develop a patch manually. We propose going a step further. By applying a series of transformations to the vulnerable application's source code, we want to automatically create a narrowly defined

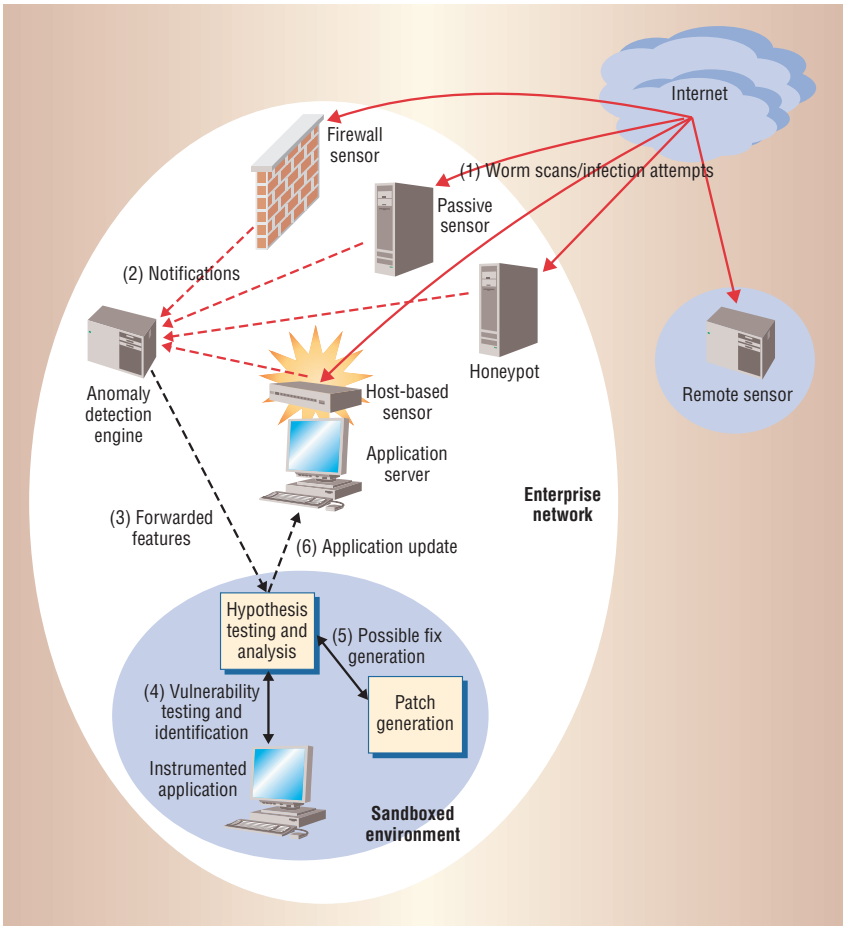


Figure 1. Worm vaccination architecture. (1) Sensors deployed at various locations in the network detect a potential worm and (2) notify an analysis engine. (3) The engine forwards the potential infection vector and other relevant information to an isolated environment, where (4) the potential infection vector is tested against an appropriately instrumented version of the targeted application and the vulnerability is identified. (5) Several software patches are generated and tested using different heuristics. If one of the patches is found to be immune to the infection, (6) the application server is updated.

patch that protects against the specific vulnerability. At its simplest, such a patch would recompile the vulnerable part of the application with a mechanism like StackGuard.

More advanced fixes are also possible. For example, we have been experimenting with patches that maintain the application's availability (http://nsl.cs.columbia.edu/projects/wormv/).

Results

Our preliminary experiments with several vulnerable open source applications show a better than 80 percent success rate in protecting against an attack and maintaining continuous operation. In the remaining cases, the patch successfully thwarts the attack, but the application terminates its execution.

An analysis of an Apache server that our system patched in response to a Slapper-like attack showed no impact on performance. Furthermore, the patching process took less than 10 seconds. Naturally, additional research is needed to determine the mechanism's limits with a richer corpus of vulnerable software.

CHALLENGES

In our exploratory work we have already identified issues that require further research, such as reasoning about the system's reliability and security, extending it to cover other types of software failures and attacks, and handling binary-only systems—to name but a few.

Risk

As with any fully automated task, the risks of relying on automated patching and testing as the only real-time defense techniques are not fully understood. Furthermore, it should be obvious that this system does not address the growing number of e-mail worms that spread by exploiting other types of security flaws and human curiosity.

Software engineering, programming languages, security, and networking are all components in developing the system. At the very least, some of the techniques we have developed should help in the traditional "penetrate and patch" cycle.

Trust

At a different abstraction level, these ideas could serve as existential proof for reactive decentralized mechanisms that do not require collaboration but can respond through purely local action to global phenomena such as worms.

Several researchers have proposed the need for large-scale collaborative mechanisms to detect and filter future worm attacks. For example, Vern Paxson proposed a Cyber Center for Disease Control for identifying outbreaks, rapidly analyzing pathogens, and fighting the infection ("How to Own the Internet in Your Spare Time," *Proc. Usenix Security Symp.*, Usenix, 2002, pp. 149-167). The CCDC could use our mechanism to quickly create needed patches.

The trust issues inherent in such an approach present unique challenges. It seems unlikely that many organizations would trust such a third entity to dynamically alter their own defensive posture—for example, by pushing network filters to network firewalls or sending software patches.

A CCDC whose role was confined to managing a large number of honeypots and attack-detection sensors would be valuable in spreading the word of new attacks. Organizations, or their security service providers, could run software vaccination systems indepen- dently of each other. All that this hypothetical CCDC would do is broadcast the newly captured attack vector and allow individual organizations to develop their own patches in real time: Trust but verify! ■

Angelos D. Keromytis *is an assistant professor in the Department of Computer Science and director of the Network Security Lab at Columbia University. Contact him at angelos@cs.columbia.edu.*

Editor: William A. Arbaugh, Dept. of Computer Science, University of Maryland at College Park; waa@cs.umd.edu

Context-Aware Trails

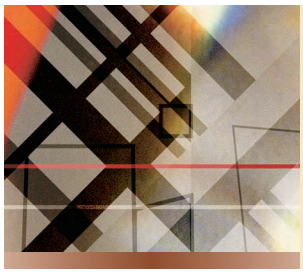
Siobhán Clarke and Cormac Driver, Trinity College Dublin

The emergence of converged mobile devices with a wide range of computing, communications, entertainment, and sensing capabilities represents a major step in the evolution of wireless computing. Such devices increasingly shift the decision-making power from the user to the machine, which has the capacity to be better informed about the current environment and can respond more quickly.

People spend much of their time planning where they're going, what to do along the way, and recalling where they've been. Trinity College Dublin's Hermes project (http://hermes.dsg.cs.tcd.ie) is exploring innovative ways to enhance these activities with proactive, context-aware mobile applications.

Drawing on previous work with the EU-funded Global Smart Spaces project (www.gloss.cs.strath.ac.uk), we use the notion of a *trail* to capture a mobile user's daily activities. At the most general level, a trail is a collection of locations, together with associated information about these locations and a recommended order for visiting them.

The trails metaphor makes it possible to explore adaptive characteristics common to all mobile, context-aware applications. Adaptation in this context involves altering the set of interest points on a trail and their visiting order with timeliness, accuracy, and relevance while remaining in tune with user expectations.



The trails metaphor makes it possible to explore adaptive characteristics common to all mobile, context-aware applications.

PROPERTIES

The most common and obvious characteristic of an adaptive trails-based application is providing support for planning an appropriate route. This involves processing a rich set of contextual information such as current traffic congestion along different roadways, mode of transport, weather, route characteristics such as scenic or fastest, crowd density, time of day, and so on.

Activities at different locations along the trail also influence the aggressiveness of application adaptation. If the tasks are deemed mandatory or high priority, the application should make it difficult for the user to ignore them. If the tasks are optional, more subtle adaptation is appropriate.

Reliable and timely dissemination of contextual information and services is another key property of adaptable applications. A mobile device with sensing capabilities can determine the information and services available in the current environment and disseminate what is appropriate for the user's activity.

Trails-based applications can also incorporate an element of team-based communication or cooperative work. To this end, models that support ad hoc communication between mobile users provide a useful feature.

APPLICATIONS

Mobile, context-aware trails-based applications range from single-user systems that focus on individual daily activities to multimedia groupware that supports a wide collection of user requirements.

For example, a simple route planner can help a business traveler in an unfamiliar city get from the airport to a meeting location. The application, deployed on a PDA, calculates the best route based on factors including the user's current location, personal transportation preferences, real-time road traffic data, and bus and train schedules. It also presents a digital map of the area, overlaid with currently feasible routes and their estimated traversal time. Dynamic route adaptation may occur if traffic conditions make a different mode of transport or route more appropriate.

Other applications can manage the activities of multiple users. For example, a delivery company can equip operatives on bicycles or motorcycles with PDAs running a trails-based information system administered by staff in a traditional office. Administrators receive client requests and assign them to mobile couriers based on their current location, mode of transport, workload, schedule, and shift end time.

The application notifies couriers of new jobs via vibrotactile alerts and displays a trail on their PDA showing the

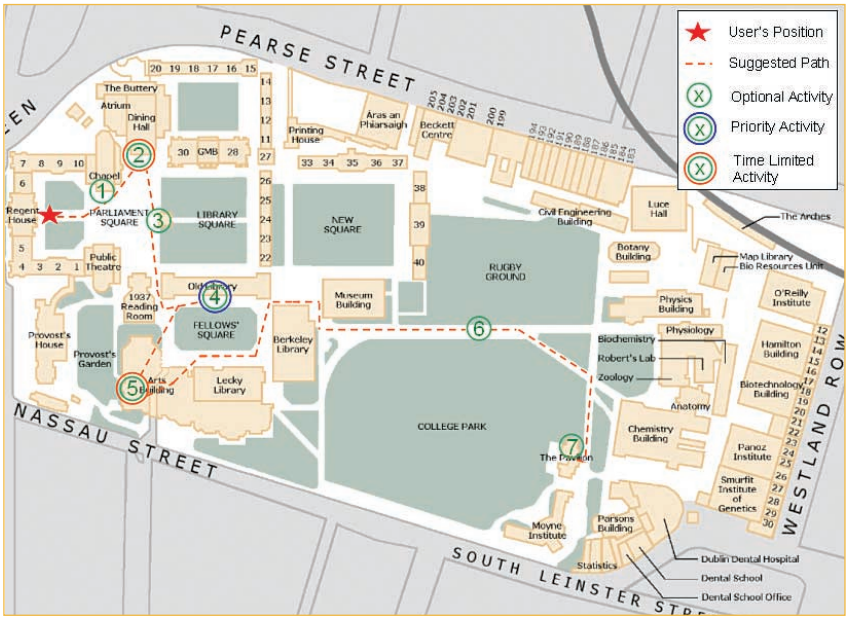


Figure 1. Context-aware trail application for mobile users. A recommended tourist trail through the Trinity Dublin Campus includes optional high-priority and time-constrained activities.

optimal route from their current location to the package pickup point and on to the drop-off point, as well as textual and audio information about the job. Dynamic route adaptation may occur based on factors such as assignment urgency and traffic conditions, and courier monitoring is also possible.

A more relaxed trail, such as that shown in Figure 1, is appropriate to support tourists' interests. For example, we are currently developing a context-based application that disseminates multimedia-based "stories" about previous inhabitants of an old part of Dublin to mobile users.

Finally, trails-based applications can be used to support multiplayer games. For example, consider a treasure-hunt game involving teams of PDA-equipped mobile players dispersed throughout a city or other large ubiquitous computing space. The game's objective is to arrive at an initially unknown destination as fast as possible by solving a set of cryptic riddles. Each clue hints at a location, which contains the next clue. Clues are tailored to team members' interests and vary in difficulty, delivery format (video, audio, or text), language, and topic.

The game uses trails to determine optimal routes between clue locations in real time, based on mode of transport, street congestion, team progress (very successful teams may be hindered slightly to keep the game interesting), and the location of other team members (if, say, two or more team members are required to reveal the clue), as well as more typical contexts such as location, weather, and time.

CHALLENGES

There have been significant advances in the understanding of human-computer interaction as users perform tasks of greater variety and complexity (J.M. Carroll, *Human-Computer Interaction in the New Millennium*, Addison-Wesley, 2002). However, the form factor and limited interface capabilities of mobile devices such as PDAs inhibit the representation of the dynamic adaptation of trails-based applications.

In addition, poor data quality complicates context acquisition. Contextual information may be stale or uncertain in terms of fuzziness and trust, while power and signal strength limitations, together with frequent user mobility, also contribute to its unreliability. Frequent re-adaptation to undo previous bad decisions is unacceptable. Sensor data fusion, a large research field, addresses these and related issues.

Further, mobile networks require new protocols for reliable and consistent sharing of information among groups of users who come together in an ad hoc fashion (J. Wu and I. Stojmenovic, "Ad Hoc Networks," *Computer*, Feb. 2004, pp. 29-31).

Moreover, the limited processing and power capabilities of PDAs challenge algorithm design for dynamic adaptation. A balance must be found between the time it takes to reconfigure the current trail and application responsiveness. The Lancaster GUIDE project (www.guide.lancs.ac.uk) provides a good example of a mobile, context-sensitive system experiencing this problem.

Determining whether the user is following a similar or previous path can enhance trail reconfiguration. Pattern recognition may be used to discover such situations, though the inevitable inexactness of repeated user behavior presents challenges for recognition and matching algorithms. The journal *IEEE Transactions on Pattern Analysis and Machine Intelligence* explores these and related issues.

Finally, a recent Invisible Computing column (B. Schilit et al., "Wireless Location Privacy Protection," *Computer*, Dec. 2003, pp. 135-137) outlined several privacy protection challenges that apply to trails-based applications.

TECHNOLOGY

In collaboration with Intel Research, we are developing a generic software framework that addresses these challenges and provides a configurable platform for developers of mobile, context-aware applications. Building such a framework involves both exploring and creating appropriate technologies.

Sensor fusion

Consider, for example, classical sensor fusion technology, which has difficulty satisfying context-sensing requirements because

- source configuration is highly dynamic,
- inexpensive sensors are inferior to human perceptual capabilities, and
- sensor numerical parameters must be converted to meaningful semantics.

Among the proposed solutions to this problem include those based on Dempster-Schafer theory, a generalized form of Bayesian theory. In addition, many projects have addressed how to fuse the output of numerous, low-cost, multimodal sensors for context awareness, including Technology Enabling Awareness (www.teco.edu/tea), MediaCup (http://mediacup.teco.edu), and Smart-Its (www.smart-its.org).

Trail generation

The trail-generation process involves two conflicting requirements:

- generating a highly customized optimal trail, and
- application responsiveness.

The first requirement negates the usefulness of classic route-generation techniques, such as shortest-path hill climbing, which do not consider users' personal route-planning preferences. Current best practices in route planning are knowledge- and experience-based. These approaches generate routes most likely to conform to the user's current preference.

For unfamiliar situations, proposed collaborative case-based approaches generate a route based on similar users' experiences. Notwithstanding the semantic difficulty of identifying similarities among different users, these algorithms' recursive nature further strains system resources and thereby slows application responsiveness.

Trail reconfiguration quality can be enhanced by pattern matching recent contextual data—for example, location, user activity, and time—against a database of previous trails. This is similar to established pattern-matching techniques such as Lemple-Ziv compression as well as memory reference string prediction. A solution based on the later technology would require a probabilistic model of which trails are most likely.

Mobile, context-aware applications have far to go to realize their potential. Progress will remain slow if developers must confront the full set of technical challenges for each individual application. An application framework that provides a reusable, configurable set of solutions to such challenges will give developers a head start. ■

Siobhán Clarke *is a lecturer in the Department of Computer Science at Trinity College Dublin. Contact her at siobhan.clarke@cs.tcd.ie.*

Cormac Driver *is a PhD candidate in the Department of Computer Science at Trinity College Dublin. Contact him at cormac.driver@cs.tcd.ie.*

Editor: Bill Schilit, Intel Research, Seattle; bill.schilit@intel.com

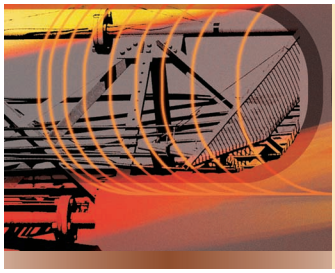
The Digital World's Midlife Crisis

Michael Macedonia, Georgia Tech Research Institute

If, like me, you've jumped from an airplane at night, you know what it feels like to be an executive at Sony, Microsoft, Nokia, Nintendo, or Kodak right now. These executives have jumped into the digital void, having no idea where they will land and scared to death their parachutes might fail.

Circumstances now force them to make choices that will result in disaster or glory. Will they all suffer the camera industry's fate, which has been eviscerated by the evaporation of its entire silver halide technology for film? By the end of this decade, the Instamatic will be Smithsonian material. Moreover, every digital device—cell phone, PDA, and even game console—will soon be a digital camera.

Or will these players strike market gold as HP did, a company that bet its fate on the lowly printer and used its color inkjets to ride the crest of the digital camera wave?



Sweeping changes in digital entertainment have pushed the industry's biggest players into freefall.

SONY AT THE CROSSROADS

Regardless, all the aforementioned companies now find themselves in a midlife crisis—wondering whether technology will be their salvation or doom. For example, Sony—a risk-taking company—has been hurting lately, recently laying off 20,000 employees (www.xbox365.com/news/news.cgi/article/EpZVuuZyAFxbbnmPnl7693). Still the world's second-largest consumer electronics company, Sony also failed to win in the mobile phone arena, where Nokia, Motorola, and Samsung now dominate, and LG is nipping at Sony's heels. Next, Sony withdrew from the US PDA market when its Clie fell prey to the cell phone integration trend (www.iht.com/articles/521467.html).

Sony has now bet its future on two technologies: the Cell microprocessor—a teraflop-capable stream processor designed by IBM and manufactured by Toshiba—and the blue-laser DVD.

The $2 billion Cell development effort represents Sony's strategic move to retain the dominant position for its most profitable product—the Playstation game console. Sony has sold more than 100 million PS2s, far outstripping the roughly 15 million game consoles Microsoft and Nintendo have each sold.

If successful, Cell will bring motion-picture-quality computer graphics to games and make each Playstation 3 a multimedia supercomputer.

Sony's Blu-ray Disc embodies the next-generation optical disc format. Jointly developed by 13 consumer electronics and PC companies—Dell, Hitachi, HP, LG, Mitsubishi, Panasonic, Philips, Pioneer, Samsung, Sharp, Sony, TDK, and Thomson—the format will enable recording, rewriting, and playback of high-definition television. Blu-ray makes it possible to record more than two hours of HDTV or 13 hours of SDTV on a 27-Gbyte disc. There are also plans for higher-capacity discs that could hold up to 54 Gbytes of data.

Sony tried to avoid a repeat of the Beta versus VHS format wars by teaming with its old adversary Matsushita, but Sony's ally on the Cell processor, Toshiba, has teamed with NEC for its own next-generation DVD, HD-DVD.

The PS3 will have Blu-ray, making the console key to Sony's strategy of proliferating the technology quickly, much as it did with DVD on the PS2. If you're wondering why DVD is so important to Sony, consider that DVD movie sales gross more than theatrical movie sales and now generate $9 billion annually.

NEXT-GENERATION CONSOLE WARS

Although IBM developer workstations with the Cell will ship this fall, the PS3 will not arrive until 2006. In a preemptive strike, Sony's two main competitors will likely beat their rival by a year with their own next-generation consoles—Microsoft's Xbox2 or Next and Nintendo's Revolution. Moreover, Microsoft may be acting as the spoiler with blue-laser DVDs.

In June, the DVD Forum steering committee (www.dvdforum.org/forum.shtml) approved version 1.0 of the

physical specifications for HD-DVD read-only discs. The forum voted to require makers of HD-DVD video playback devices to build in three video codecs, including the VC-9 technology used in Microsoft's Windows Media Video 9 (www.microsoft.com/windows/windowsmedia/9series/codecs/video.aspx). It's also a good bet that Microsoft's console will have an HD-DVD drive.

In a dramatic departure, Microsoft—whose game division has never shown a profit—will switch manufacture of its Xbox's CPU from Intel to IBM. It will also switch the Xbox's graphics chip vendor from Nvidia to ATI. Moreover, Microsoft will likely scrap the original's hard drive and offer no backward compatibility with games for the first machine.

For ultimate irony, consider that long-time rival Apple makes the Xbox2 developer's workstation, a G5 Macintosh. This radical strategy may well be short-sighted: In its race to beat Sony, Microsoft could alienate its current Xbox user base, who will soon be buying heavily hyped titles like *Halo 2*—which will not play on the Xbox2.

Nintendo already has the GameCube, manufactured by Matsushita, Sony's ally in the Blu-ray DVD effort. This console has an IBM Power PC chip and ATI graphics. Its Revolution console will also have an IBM Power PC chip and ATI graphics—factors that likely prompted Nintendo's president, Satoru Iwata, to make these stunning comments at the 2004 E3 (www.gamespy.com/articles/523/523168p1.html):

> Better technology is good, but not enough. Today's consoles already offer fairly realistic expressions so simply beefing up the graphics will not let most of us see a difference. So what should a new machine do? Much more. An unprecedented gameplay experience. Something no other machine has delivered before. ... I could give you our technical specs, as I know you'd like that, but I won't for a simple reason: they really don't matter. The time when horsepower alone made all the difference is over.

HOLDING ON WITH HANDHELDS

Although Nintendo's GameCube sales have been lagging, its highly profitable Game Boy Advance has been a leader in handhelds. Game Boy has been the platform for more than 12 million unit sales of *Pokemon* in the past two years. At $40 apiece, that's almost half a billion dollars in revenue from a single game.

Microsoft will switch manufacture of its Xbox's CPU from Intel to IBM.

To maintain dominance, Nintendo demonstrated its two-screen DS as this year's E3. But Sony counterattacked with the announcement of the Playstation Portable for 2005. The PSP has a larger screen, DVD-quality movies, and Wi-Fi 802.11.

Both the Sony and Nintendo handhelds will let users connect to the Internet, send messages to friends, or compete against one another without cables or phone lines. Both will also compete with Nokia, which is trying to build a following with its N-Gage handheld cell phone game system.

As Zelos Group senior analyst Billy Pidgeon noted (www.hollywoodreporter.com/thr/columns/tech_reporter_display.jsp?vnu_content_id=1000552950), "We expect to see $230 million in game purchases and subscription fees in the US in 2004, up from $77 million last year. And total US market revenues will reach $616 million in 2008, at which time almost 30 million consumers will pay for wireless games."

N-Gage has not, however, set the world on fire. Still, it may presage the possibility that within the next five years most of the 500 million cell phones sold each year will have 3D graphics accelerators courtesy of Nvidia and ATI, 4G broadband connectivity, GPS, and a library of games. Nokia, which releases 60 new models a year, is evolving its cell phones into anywhere multimedia machines. The challenge for Nokia is to build a market big enough and fast enough to beat Sony and Nintendo's entrance into the wireless game market through their handhelds.

We are in the midst of a convergence between high-definition television and game consoles and between handheld game machines and mobile phones. This market shift will create both big losers and big winners. Currently, IBM occupies the most favorable position, with chip designs for all three next-generation consoles. But all the contenders find themselves facing an uncertain future.

In essence, the players in this market are contending with what Clayton M. Christensen calls the "innovator's dilemma" (www.businessweek.com/chapter/christensen.htm): Leading firms are "held captive by their customers, enabling attacking entrant firms to topple the incumbent industry leaders each time a disruptive technology emerged." For Microsoft that disruptive technology might be Sony's supercomputing console running Linux; for Nintendo, it might be Nokia's N-Gage; and for Sony, it might be the Chinese high-definition format, Enhanced Versatile Disc (http://zdnet.com.com/2100-1103_2-5150373.html?tag=nl).

One thing is certain, however: Each of these firms is already immersed in an internal midlife crisis to redefine its identity so that it can triumph in the external struggle against its rivals in the game market. ■

Michael Macedonia *is a senior scientist at the Georgia Tech Research Institute, Atlanta. Contact him at macedonia@computer.org.*

An "Offshore-Resistant" Degree Program

David A. Swayne, Qusay H. Mahmoud, and Wlodek Dobosiewicz
University of Guelph

The new University of Guelph-Humber has created an innovative, combined university degree and college diploma program in computer science and communications technology. This venture between the university and the Humber Institute of Technology and Advanced Learning offers jointly an honors baccalaureate in computing science from Guelph and a diploma in wireless and telecommunications systems technology from Humber. The program, one of six joint programs between the two institutions, specializes in distributed and wireless computing.

PROGRAM DETAILS

The program, which comprises eight semesters of full-time study plus two eight-month cooperative education work placements, is divided into two distinct phases of equal duration:

- a lower division of two years, in which courses cover the basics of programming, mathematics, physics, and communications systems; and
- an upper division that consists of the two cooperative work placements and material designed to deliver a highly specialized and detailed program of study in advanced communications technologies and distributed computing.

Will proficiency in both computer science and communications give students a global edge?

Figures 1 and 2 show the curriculum's largely self-explanatory course names. Two calculus courses, an applied algebra course, and a probability and statistics course cover elementary mathematics. A course in ethics forms part of the third semester's curriculum. The first five programming courses give a fairly complete picture of the underpinnings of programming, algorithms, and elementary data structures. The requisite physics courses for communications consist of the two first-year electronics courses and the fourth-semester electromagnetics course. Physical application principles dominate the courses in communications systems.

The University of Guelph-Humber's combined-program concept offers a distinct advantage over the standard undergraduate computer science model because it integrates two different aspects of computer applications. Students will graduate with skills in both computing and communications, ensuring that a broad spectrum of positions will be available to them. We believe our program's integration of these two disciplines will produce graduates whose skills cannot be easily outsourced. The "A New University Model" sidebar describes in detail how we crafted this program.

There is nothing to prevent our program from being emulated by others, particularly from other countries. However, we feel that many of the applications we address must remain close to the source. Examples of such applications would be secure communications—applications with hardware and software dependencies. When all else fails, the need for the hardware to be situated locally also carries the natural need to maintain the software locally. That is, these sorts of students will be most useful in projects that do not travel well.

Faculty and staff

The program draws its faculty members evenly from Guelph and Humber, with three Guelph faculty members currently at the new campus. New staff members have been hired to support the program. The University of Guelph-Humber has its own staff to handle student recruiting.

Students

Most of Guelph-Humber's students came from Ontario province. While some had no exposure to programming, many were self-taught and some even had industry certificates such as those granted by the A+ and Microsoft Certified Systems Engineer (MCSE) programs. We accepted 36 students for

Continued on page 102

Continued from page 104

Year 1 **Semester 1** F2002	***Year 2*** **Semester 3** F2003
Procedural Programming 1	User-Centered Programming 3
Electronics Fundamentals	Principles of Telecommunications
Telecommunication Computations	Communication Systems
Calculus for Computing I	Data-Centered Programming 4
Elective	Ethical Issues in Info. Technology
	Workplace Prep and Tech. Writing (half course)
Semester 2 W2003	**Semester 4** W2004
Intro. to Object-Oriented Programming 2	Data Structures 5
Semiconductor Electronics	Probability & Statistics
Calculus for Computing II	Data Networks
Anatomy of a PC	Electromagnetics
Digital Systems	*Elective*

Figure 1. Lower-division program, programming courses numbered 1 through 5.

Work 1 (Spring 2004) and Work 2 (Fall 2004)	***Work 3 (Fall 2005) and Work 4 (Winter 2006)***
Year 3 **Semester 5** W2005	***Year 4*** **Semester 7** S2006
Operating Systems	Distributed Software Engineering
Distributed Programming	Internet Technologies
Digital Signal Processing	Senior Paper I (half course)
Fibre Optic Communications	***Telecom*** *Elective*
Telecom *Elective*	***DCCT*** *Elective*
	DCCT *Elective*
Semester 6 S2005	**Semester 8** F2006
Database Systems & Concepts	Distributed Info. System Architectures
System & Network Simulation	***Telecom*** *Elective*
Wireless Data Networks	***Telecom*** *Elective or CS Elective*
Cellular Technology	Technology & Society
DCCT *Elective*	Senior Paper II
Sample ***Telecom*** *Electives:* *Broadband Communications* *Telephone Switch Technology*	*Sample CS Electives:* *Cryptography and Network Security* *Artificial Intelligence*

Figure 2. Upper-division program.

the Fall 2002 semester, 30 of whom are now in their second year. We also accepted 61 students for the Fall 2003 semester, 54 of whom have successfully finished their first year. Second-year students will be away from May through December 2004 on an eight-month work placement, then they will return in January 2005 for two more semesters of study.

Staying small

Part of the Guelph-Humber equation relies on a funding formula based on a maximum class size of 65 students and a maximum laboratory size of 35 students. This improves significantly on typical university class sizes in Ontario, particularly with the double cohort—a situation created by the elimination of the 13th year of fully funded public education. It is a mixed blessing, though. From a selfish perspective, high attrition rates would lead to the early demise of the program. With this in mind, we framed retention strategies that focus on the introductory programming language and system choices and our control over the mathematics curriculum.

PROBLEM AREAS

Our experience teaching technological programs at several universities led us to believe that at least three main subjects have excessive failure rates or bad outcomes, defined as a D grade, a fail, or a drop: calculus, computing, and chemistry—the so-called three Cs. Computing science programs typically have only calculus and computing.

Apparently, the damage usually is done in the first semester. Students who survive often have their cumulative averages reduced and their careers blighted by difficulties that frequently evaporate when they catch on to the subjects. Thus, our curriculum focuses on building expertise and maintaining a sense of accomplishment, rather than discouraging and frustrating students.

The programming problem

We recognize that it is better to become proficient at one language and its environment than to risk confusion through diversification. Therefore, our first four programming-language courses focus on the use of HTML, JavaScript, and Java. We stay with Java through the third semester, introducing user interfaces, elementary data structures, and sockets programming in Java. Then, in the fifth course, we adapt the programming for data structures to C++, combining the learning of data structures and algorithms with a new but very similar language—but doing so in a semester in which the curriculum covers no other major programming component.

Our most important retention strategy is programming for fun (Qusay H. Mahmoud, Wlodek Dobosiewicz, and David Swayne, "Making Computing

Programming Fun and Flexible," *Computer*, Feb. 2004, pp. 108, 106-107) and otherwise managing student expectations in the first semester. We achieve this by encouraging and controlling collaboration in programming, facilitated by using programming tools that give good feedback to the student.

The calculus problem

We recognize that far too many students drop out of computer science programs for the wrong reasons—the transition from secondary to postsecondary school being a principal one. We decided to exercise control over the mathematics courses to ensure that whatever time is available to the students is well spent. Calculus, particularly in the first semester, involves three hours of lectures, three hours of labs and tutorials, and one hour of testing per week. Extensive training in the use of tools such as algebraic software empowers the students to solve rather dramatic problems that would only devolve into messy calculations if solved using pencil and paper.

The time has come for computer science to become interdisciplinary and for computer science departments to offer innovative programs that will produce graduates whose skills cannot be outsourced.

Our program started after the dotcom crash. Because of the Ontario Access to Opportunities Program (ATOP), which sought to have all computer science programs in the province double in size over four years, we are competing for a shrinking pool of students in a time of computing industry retrenchment. Our strategies include staying small, community building, and bolstering retention without sacrificing integrity.

The University of Guelph-Humber has been a huge success. As of this writing, more than 80 percent of the computing students who finished their second year have found co-op jobs with medium to large companies, such as CIBC, RIM, and Sybase. ■

A New University Model

From 1999 to 2001, the presidents of the University of Guelph and the Humber Institute of Technology and Advanced Learning negotiated an agreement to offer joint programs sanctioned by the Ontario Ministry of Training, Colleges, and Universities. This venture received $28.63 million in funding as part of the government of Ontario's SuperBuild initiative to meet the needs of the increasing postsecondary student population.

The Department of Computing and Information Science at the University of Guelph and the Faculty of Information Technology and Accounting at Humber thus entered into co-development of the Wireless program. Originally, Wireless provided a package for renovating existing programs at both institutions. Resistance to the program at Guelph and the constraints imposed by squeezing the joint program into eight semesters posed an immediate threat to this form of advancement by small steps, however.

Development thus began in the fall of 2000 on a completely new program, and continued until December 2001. This program subsumed earlier efforts and expanded the Wireless notion into a broader study.

The development team obtained requisite if guarded approval to proceed with the new program, whose calendar designation became its abbreviated name—Distributed Computing and Communications Technology or DCCT. The team developed a preliminary curriculum and several analyses to show how the proposed degree program would enhance the University of Guelph's learning objectives and the development of generic skills.

In December 2001, we held a workshop with participants from both institutions as well as guest presenters Ralf Denzer and Ernst-Erich Doberkat, professors who had developed similar curricula in Germany. Significant revisions to the Wireless program resulted from this workshop.

Construction of a joint Guelph-Humber building began in November 2001, and the building opened in the fall of 2003. The University of Guelph-Humber became the first in Ontario to offer both a university degree and a college diploma in just four years of study. The DCCT was one of the three inaugural programs that, from Guelph's point of view, provided a significant response to the Ontario government's double cohort, a situation created by the elimination of the 13th year of fully funded public education.

David Swayne *is a professor in the Department of Computing and Information Science, University of Guelph, and program coordinator of the Distributed Computing and Wireless & Telecommunications Systems Technology program, University of Guelph-Humber. Contact him at dswayne@cis.uoguelph.ca.*

Qusay H. Mahmoud *is an assistant professor in the Department of Computing and Information Science, University of Guelph, and associate chair of the Distributed Computing and Wireless & Telecommunications Systems Technology program, Guelph-Humber. Contact him at qmahmoud@cis.uoguelph.ca.*

Wlodek Dobosiewicz *is a professor and chair of the Department of Computing and Information Science, University of Guelph. Contact him at dobo@cis.uoguelph.ca.*

Editor: Neville Holmes, School of Computing, University of Tasmania; neville.holmes@ utas.edu.au