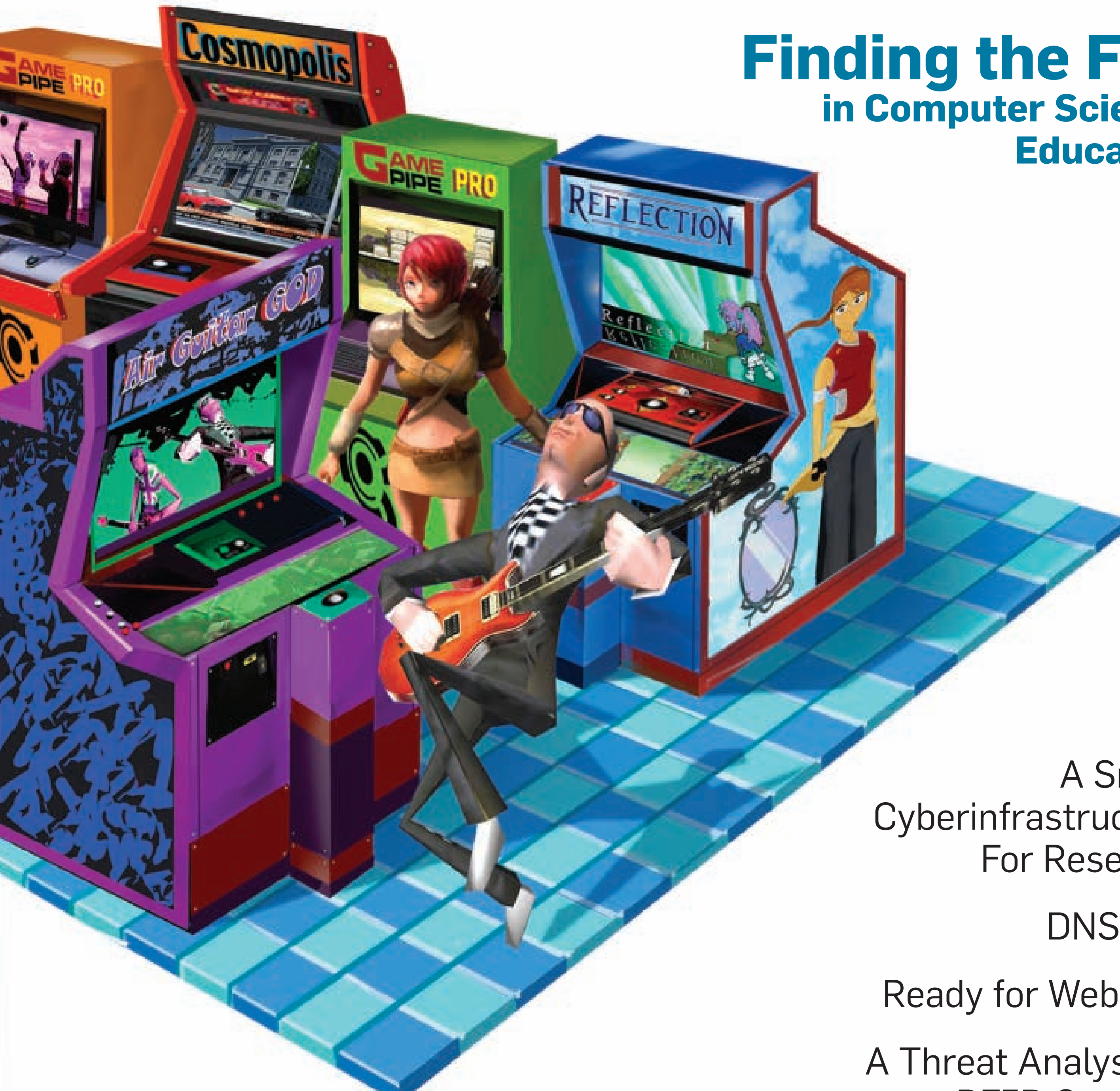
# COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

12/2009 VOL.52 NO.12


## Finding the Fun in Computer Science Education

A Smart Cyberinfrastructure For Research

DNS Lies

Ready for Web OS?

A Threat Analysis of RFID Systems


Association for Computing Machinery

The Chartered Institute for IT

Association for Computing Machinery

# International Academic Research Conference

## ACM-BCS Visions of Computer Science 2010

## Programme Committee

Michel Beaudouin-Lafon (LRI)
Mic Bowman (Intel)
Matthew Chalmers (Glasgow)
Fabrizio Gagliardi (Microsoft)
Carole Goble (Manchester)
Matt Grossglauser (Nokia)
David Hutchinson (Lancaster)
Jane Hillston (Edinburgh)
Matt Jones (Swansea)
Ross King (Aberystwyth)
Marina Kirotka (Oxford)
Mounia Lalmas (Glasgow)
Michael Luck (Kings College)
Julie McCann (Imperial)
Ursula Martin (Queen Mary)
Simon Moore (Cambridge)
Colin Runciman (York)
Nigel Shadbolt (Southampton)
Susan Stepney (York)
Alexander Wolf (Imperial)

## Programme Chairs

Derek McAuley (Nottingham)
Simon Peyton-Jones (Microsoft)

## The University of Edinburgh

### 13-16 April 2010

### First call for papers; submission 18 December 2009

The programme committee, ACM - The Association for Computing Machinery and BCS - The Chartered Institute for IT, are delighted to invite you to the joint ACM-BCS 2010 "Visions of Computer Science" conference, to be held at the Informatics Forum, Edinburgh University, 13 - 16 April 2010. This flagship event aims to energise the computing community and bring it together around some positive and inspiring visions of our discipline and follows the highly successful "Visions of Computer Science" conference in 2008.

Submissions are being solicited in all areas of research covering the broad field of Computer Science and Engineering (CSE). They include but are not limited to:· Computer Architectures and Digital Systems; Theoretical Computer Science: Algorithms and Complexity, Logic and Semantics; ·Non-standard Models of Computation; ·Programming Methods and Languages; ·Software Engineering, and System Design Tools; Quantitative Evaluation of Algorithms, Systems, and Networks; ·Artificial Intelligence, Agents, and Machine Learning; Networks; ·Distributed and Pervasive Systems; ·Grid Computing and eScience; Digital Economy; ·Databases, Information Retrieval and Data Mining, Web based Computation; ·Human Computer Interaction; ·Robotics and Computer Vision; ·Bioinformatics, Synthetic Biology and Synthetic Chemistry; ·Medical Applications.

Papers related to the current UKCRC Grand Challenges in Computing www.ukcrc.org.uk/grand-challenge offering new insights or approaches are welcome.

### The relevant dates for authors are:

- submission: 18 December 2009
- notification: 19 February 2010
- camera-ready: 5 March 2010

Instructions for authors and the required paper layout can be found at www.bcs.org/visions2010 ; *the strict limit for submissions is 12 A4-pages in 10pt two-column style, including figures and references.* Authors may include a clearly-marked appendix, which referees may or may not take into account. Submissions to be made at www.easychair.org/conferences/?conf=bcs10

*The proceedings will be published by BCS in their electronic proceedings series www.bcs.org/ewic on the ACM Digital Library, while a special issue of The Computer Journal http://comjnl.oxfordjournals.org will be dedicated to a collection of the best papers.*

Springer
the language of science

springer.com

# Noteworthy LNCS Titles

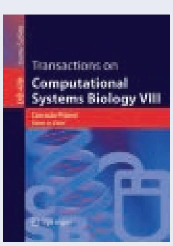

## Pattern Recognition in Bioinformatics

### Third IAPR International Conference, PRIB 2008, Melbourne, Australia, October 15-17, 2008. Proceedings

**M. Chetty**, **A. Ngom**, **S. Ahmad** (Eds.)

This book constitutes the refereed proceedings of the International Workshop on Pattern Recognition in Bioinformatics, PRIB 2008, held in Melbourne, Australia, in October 2008. The 39 revised full papers presented were carefully reviewed and selected from 121 submissions. The papers discuss the applications of pattern recognition methods in the field of bioinformatics to solve problems in life sciences.

2008. XVI, 472 p. (Lecture Notes in Computer Science / Lecture Notes in Bioinformatics, Volume 5265) Softcover
ISBN 978-3-540-88434-7 ► **$92.95**

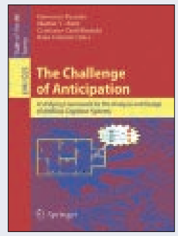



## Transactions on Computational Systems Biology VIII

**C. Priami**, University of Trento, Povo, Italy (Ed.)

The LNCS journal Transactions on Computational Systems Biology is devoted to inter- and multidisciplinary research in the fields of computer science and life sciences and supports a paradigmatic shift in the techniques from computer and information science to cope with the new challenges arising from the systems oriented point of view of biological phenomena.

2007. VII, 103 p. (Lecture Notes in Computer Science / Transactions on Computational Systems Biology, Volume 4780) Softcover
ISBN 978-3-540-76638-4 ► **$69.95**



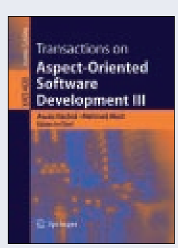

## The Challenge of Anticipation

### A Unifying Framework for the Analysis and Design of Artificial Cognitive Systems

**G. Pezzulo**, Istituto di Scienze e Tecnologie della Cognizione – CNR, Rome, Italy; **M. V. Butz**, Universität Würzburg, Germany; **C. Castelfranchi**, **R. Falcone**, Istituto di Scienze e Tecnologie della Cognizione – CNR, Rome, Italy (Eds.)

This book proposes a unifying approach for the analysis and design of artificial cognitive systems: The Anticipatory Approach. In 11 coherent chapters, the authors of this state-of-the-art survey propose a foundational view of the importance of dealing with the future, of gaining some autonomy from current environmental data, and of endogenously generating sensorimotor and abstract representations.

2008. XVI, 288 p. (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence, Volume 5225) Softcover
ISBN 978-3-540-87701-1 ► **$69.95**

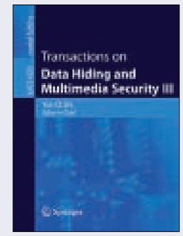



## Transactions on Aspect-Oriented Software Development III

**A. Rashid**, Lancaster University, UK; **M. Aksit**, University of Twente, The Netherlands (Eds.)

The LNCS Journal "Transactions on Aspect-Oriented Software Developmen"t is devoted to all facets of aspect-oriented software development (AOSD) techniques in the context of all phases of the software life cycle, from requirements and design to implementation, maintenance and evolution.

2007. IX, 201 p. (Lecture Notes in Computer Science / Transactions on Aspect-Oriented Software Development, Volume 4620) Softcover
ISBN 978-3-540-75161-8 ► **$69.95**

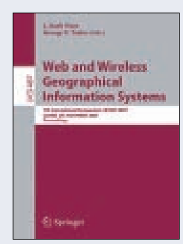



## Transactions on Data Hiding and Multimedia Security III

**Y. Q. Shi**, New Jersey Institute of Technology, Newark, NJ, USA (Ed.)

Besides, two related disciplines, steganalysis and data forensics, are increasingly attracting researchers and becoming another new research field of multimedia security. This journal, LNCS Transactions on Data Hiding and Multimedia Security, aims to be a forum for all researchers in these emerging fields, publishing both original and archival research results.

2008. IX, 91 p. (Lecture Notes in Computer Science / Transactions on Data Hiding and Multimedia Security, Volume 4920) Softcover
ISBN 978-3-540-69016-0 ► **$79.95**




## Web and Wireless Geographical Information Systems

### 7th International Symposium, W2GIS 2007, Cardiff, UK, November 28-29, 2007, Proceedings

**J. M. Ware**, **G. E. Taylor**, University of Glamorgan, UK (Eds.)

This book constitutes the refereed proceedings of the 7th International Symposium on Web and Wireless Geographical Information Systems, W2GIS 2007, held in Cardiff, UK, in November 2007. The 21 revised full papers presented were carefully reviewed and selected from 45 submissions.

2007. XI, 293 p. (Lecture Notes in Computer Science / Information Systems and Applications, incl. Internet/Web, and HCI, Volume 4857) Softcover
ISBN 978-3-540-76923-1 ► **$69.95**

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

014092x

# COMMUNICATIONS OF THE ACM

## Departments



## Last Byte



Association for Computing Machinery
*Advancing Computing as a Science & Profession*

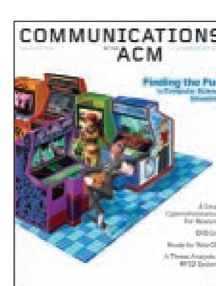
## News

**Grace Hopper Attendee Natalia Villanueva-Rosales**



## Viewpoints






**About the Cover:** Artist Lee MacLeod, who has produced illustrations for game companies including Nintendo, Sega, and Activision Value, fashioned a playful arcade reflecting real games and video characters created by students of the GamePipe Laboratory at the University of Southern California.

PHOTOGRAPH BY GAIL CARMICHAEL / HTTP://COMPSCIGAIL.BLOGSPOT.COM

12/2009
VOL. 52 NO. 12

## Practice



Article development led by acmqueue
queue.acm.org

## Contributed Articles



## Review Articles



## Research Highlights



## Virtual Extension

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

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



Association for Computing Machinery
*Advancing Computing as a Science & Profession*

ILLUSTRATION BY ANDY GILMORE

# COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

*Communications of the ACM* is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

*ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.*




**ACM U.S. Public Policy Office**
Cameron Wilson, Director

1100 Seventeenth St., NW, Suite 50
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

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

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




CONTACT POINTS

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

WEB SITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/guidelines





**Subscriptions**
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $100.

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

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


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

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

Association for Computing Machinery

PLEASE RECYCLE THIS MAGAZINE

Printed in the U.S.A.

DOI:10.1145/1610252.1610253

Chris Stephenson

# It is a Pivotal Time for K–12 Computer Science

For many, the recent news that enrollments in undergraduate computer science programs are no longer on a downward slide is reassuring, but I seriously hope that it doesn't begin to undo the good that has come out of our mutual fear, most especially that it does not diminish the commitment across all educational levels to address the serious issues in K–12 computer science education.

Creating real, sustainable change in any area of education is a frustratingly slow process. Public education is a complex bureaucracy where competing ideologies, philosophies, ontologies, and pedagogies vie for attention and control. It is subject to extreme pressures. As a profession, teaching is simultaneously professionalized and devalued. And, in most cases, the employees are overworked, underpaid, and severely under-resourced.

But change is possible. Five years ago, ACM founded the Computer Science Teachers Association (CSTA) with the goal of addressing serious concerns in K–12 computer science education, including the lack of curriculum standards, poor professional development for teachers, common misunderstandings about computer science, student and parent perceptions that there are no jobs in the computing field, and the complete mess that is computer science teacher certification.

Today, CSTA stands as an example that faith, funding, and a whole lot of volunteer support from the top to the bottom can achieve something close to miracles. The guidelines in the *ACM Model Curriculum for K–12 Computer Science* are now recognized as the defacto national computer science curriculum standards. Thanks to its partnership with colleges and universities through the JETT and TECS programs, CSTA has held more than 96 professional development workshops for teachers across the U.S. The annual Computer Science & Information Technology symposium is the closest thing we have to an annual national conference for K–12 computer science and information technology educators. CSTA's white paper *The New Educational Imperative: Improving High School Computer Science Education* provides a cogent, research-supported argument for the role that computer science must play in the K–12 academic canon. Careers in computing resources developed by CSTA and the ACM Education Board have now made their way into every school in the U.S. and several other countries as well. Finally, CSTA's new report on teacher certification (*Ensuring Exemplary Teaching in an Essential Discipline*) proposes a new framework for guaranteeing we have the teachers we require with the skills we need in our classrooms.

But something equally important came out of the enrollment crisis. Colleges and universities facing dwindling class sizes began reaching out to K–12 computer science educators and students in order to recruit students directly into their programs, and in doing so, many faculty learned that the assumptions they had been making about what it is like to teach computer science in K–12 were misguided at best and paternalistic at worst.

Many colleges and universities now have ongoing outreach and mentoring programs because they understand that waiting for students to come to them is a recipe for disaster. Interest in computer science must begin long before students sign up for their first university or college courses. It begins in K–12 where other academic disciplines first sow the seeds of interest and engagement.

This increased interest in direct outreach to K–12 has also prompted a greater level of understanding and a spirit of cooperation across educational levels that will benefit us greatly in the long run, but only if we do not lose sight of what we are doing and return to our former isolated complacency.

The next year will be a pivotal one for us. Talk (and hopes) of a new, relevant, rigorous, and more engaging sequence of high school computer science courses (including a new gold standard Advance Placement computing course) and an ambitious plan for teacher professional development could mean a real renaissance for our discipline and field.

As we begin to work toward these goals, the challenges will be enormous as will the temptation to circle the wagons and fire inward. I hope our community has the vision to rise above fragmentation and discord so we can work together to do something important and valuable. We did it when we formed CSTA and I think we can do it again to reframe computer science education.

**Chris Stephenson** (chris.stephenson@comcast.net) is the executive director of Computer Science Teachers Association.

DOI:10.1145/1610252.1610255

# In CS Education, Educate the Educators First

In their Point/Counterpoint "CS Education in the U.S.: Heading in the Wrong Direction?" (July 2009), Robert Dewar wrote that the CS curriculum lacks important fundamentals, mostly of a mathematical nature, and Owen Astrachan wrote that "Studying more mathematics will not make software bugs disappear, although both [Edsger W.] Dijkstra and Dewar seem to think so," concluding implicitly that more mathematics in the CS curriculum is not necessary.

Astrachan apparently overlooked the distinction between necessary and sufficient conditions, a routine distinction in mathematics. As far as I know, neither Dijkstra nor Dewar ever claimed that mathematics is a "silver bullet" guaranteeing reliable design, and both would likely agree with Fred Brooks that indeed there is no silver bullet.

Dijkstra's and Dewar's argument that mathematics is an essential aspect of a proper CS education mirrors the fact that education in all engineering disciplines involves a solid mathematical foundation from the start, then used in all (other) courses; see my article "Teaching and Practicing Computer Science at the University Level" in *Inroads*, *SIGCSE Bulletin 41*, 2 (June 2009), 24–30.

As long as CS educators cannot agree about the fundamentals, practice will remain below the professional standards common in classical engineering; see Allen Tucker et al.'s "Our Curriculum Has Become Math-Phobic" in *SIGCSE Bulletin 33*, 1 (Mar. 2001), 243–247. This could be the result of declining CS student enrollment, possibly leading to the replacement of mathematics with, say, trendy topics apparently more appealing to freshmen. However, even trendy topics can be combined with a solid mathematical foundation, with the trendy topics included as illustrations. In any case, the emphasis in teaching such topics must be mathematical modeling, not mere description.

Unfortunately, teachers are divided, so combining theoretical CS expertise and practical experience in the same person is rare, unlike in other engineering disciplines. Educating the educators may well be a first priority.

**Raymond Boute**, Ghent, Belgium

## Modular Programming Still a Challenge

In Leah Hoffmann's interview "Liskov on Liskov" (July 2009) Barbara Liskov recalled research being conducted at the time she was beginning her career (1970). "When I started," she said, "the main way people thought about modularization was in terms of subroutines… But they didn't have any way of linking a bunch of procedures together." She also said (in Karen A. Frenkel's news item "Liskov's Creative Joy," also July 2009) "…the major challenge still is how to build large software systems."

The definition of "module" inevitably affects the building of large software systems, prompting our own recollections and connections. For example, a counterexample to the use of subroutines as the basic module existed in the form of a project that began in 1965 at Argonne National Laboratory (ANL) and was reported at the Spring Joint Computer Conference (May 1969) to develop the Argonne Reactor Computation System (ARC) from linked computational modules running on IBM System 360 computers and OS/360. The modules were FORTRAN IV main programs where the output of one program was analyzed to become the input for other programs. Reactor designers wanted three main features:

*Superprograms.* To link programs into superprograms directed by another FORTRAN IV main program while the original programs could still run on their own. They wanted all Assembler code and any direct communication with the manufacturer's software to be isolated for porting to other hardware. Initially, the most important Assembler code was for LINKing and LOADing modules for coordinating the I/O stream and communicating with JCL;

*New algorithms.* To improve the modules through new-algorithm rewrites wherever possible; and

*New modules.* To make it possible to build modules as required.

ARC programmers each had five to 10 years of experience (rare in 1965) and advanced degrees in mathematics, science, and engineering. They coded in Absolute and Assembler on ANL-built hardware (AVIDAC and GEORGE) and in FORTRAN on the IBM 704 and the CDC 3600. Because OS/360 was so unstable (as shipped in 1965), they needed to be able to decipher dumps. Completion of the project would have been delayed without programmers capable of working close to the hardware and operating system.

The ARC system later became the platform for ANL reactor calculations and was studied and used by other laboratories worldwide, though not before it was shown to be portable.

On the first Earth Day (Apr. 22, 1970), a joint project with the Control Data Corporation aimed to port the ARC system to CDC hardware. We were aided in this effort by Richard Lee of CDC and Larry Atkins, a Northwestern University engineering student also known for writing Chess 3.6, the winner of several ACM North America Computer Chess Championships in the 1960s and 1970s. Once the modular environment was ported, the computational modules were easily ported. In fact, after one computational module was ported, the remaining work was almost automatic.

**Louis C. Just**, Lakewood, CO
**Gary K. Leaf**, Argonne, IL
**Bert J. Toppel**, Argonne, IL

## A Quart of NFE Solution for a Pint CPU Problem

In his article "Network Front-End Processors, Yet Again" (June 2009), Mike O'Dell seemed to be arguing what I say to colleagues in simpler terms, namely, that using a network front-end (NFE) processor is (and always has been) a bad idea for one basic reason: It must perform moderately complex computation at least as fast as a computer's "main" CPU, yet manufacturers insist it be cheap due to the cost "multiplier" effect when a computer includes many network links. To show that solving this problem is impossible, consider the following proof by contradiction:

Assume an NFE processor with requisite attributes—moderately general-purpose, fast, and cheap—with most computer manufacturers using it as a main processor, not as a lowly NFE and so in need of even more network bandwidth.

Many computer engineers have long understood, as O'Dell wrote, that the most efficient way to implement a network-protocol software stack is to use one or more CPUs of an *N*-way SMP, but users strongly resist this idea when they discover they've paid big for what, from an application point of view, is only an *N*-1-way computer; note, too, popular "low-end" cases in which $N = 2$ or 4. Apparently, Sun Microsystems ("the network is the computer") hasn't left much of an impression on IT managers. The result is that NFE startups continue to waste engineering talent by trying to pour a quart of technology into this particular pint jar.

**Scott Marovich**, Palo Alto, CA

## How to Address Big Data

I want to thank Adam Jacobs for cataloging the important issues in the management of large volumes of data in his article "The Pathologies of Big Data" (Aug. 2009). But please know that innovations are also being made by relational database vendors. One recently released product is the HP Oracle Database Machine (http://www.oracle.com/database/database-machine.html) that processes large volumes of data using massive horizontal parallelism across a shared-nothing storage grid. Pipeline (vertical) parallelism is enabled by offloading data processing to a storage grid, so the amount of data that must be shipped back from the storage grid to the database grid is reduced as well. The storage grid and database grid are connected through a high-speed Infiniband interconnect.

A single DM has an I/O bandwidth of 14GB/sec in the first version of the product (with uncompressed data). If the data is compressed, the effective bandwidth is much greater, depending on compression ratio. (Jacobs's experiment would have run much faster on the DM.) Multiple DMs can be connected to increase data capacity, along with corresponding network/computing capacity.

Oracle's Automatic Storage Management (ASM) is an integral part of DM, providing automatic load balancing across all nodes in the storage grid. ASM also provides fault-tolerance through dual or triple mirroring. The Oracle database provides high-availability and disaster-recovery features, and ASM enables sequential I/Os through its allocation strategies (such as large allocations).

One of the best ways to improve query performance (assuming the most optimal access method is used) is to avoid I/O altogether. The Oracle database provides rich partitioning strategies that enable skipping large chunks of data that do not qualify as a query scan.

Moving data from production (OLTP systems) to a specialized data store adds to a system's total cost of ownership, as one would otherwise be managing two different data stores with poor or no integration. DM solves the big-data problem without a special data store just for data analysis; DM provides a single view of the data.

Though I am a technical member of the Oracle Exadata development team, my aim here is not to plug the product but report that the big-data problem is indeed being tackled, particularly by relational database vendors.

**Umesh Panchaksharaiah**, Richmond, CA

## My Generation

Samuel Greengard's news story "Are We Losing Our Ability to Think Critically" (July 2009) is inspiring as a basis for future work. I am coordinating an interdisciplinary seminar on the collective construction of knowledge (http://seminario.edusol.info in Spanish), including two topics Greengard might be able to bridge: One is free software in a democratic society, inciting people to be more politically active and involved, despite being (usually) independent of political parties and other traditional means of shaping society. The other is how motivation and peer-recognition function in these communities; such free-culture communities have much in common with scientific communities, despite starting off with completely different motivations.

My generation (born in the 1970s), including many people in the free software movement, has directly experienced the great shift computing and networking have brought the world, fully embracing the technologies. The greatest difference between people who are just users of computing and those striving to make it better depends on who has the opportunity to appropriate it beyond, say, the distraction level, the blind Google syndrome, or the simple digestion of "piles of data and information [that] do not equate to greater knowledge and better decision making."

Thanks to Greengard for sparking some useful thoughts.

**Gunnar Wolf**, Mexico City

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

# BLOG@CACM

The *Communications* Web site, http://cacm.acm.org, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish excerpts from selected posts.

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

DOI:10.1145/1610252.1610257 http://cacm.acm.org/blogs/blog-cacm

# CS Woes: Deadline-Driven Research, Academic Inequality

*Jeannette M. Wing writes about the negative effects of deadline-driven research and Mark Guzdial discusses the role of computer science faculty in fostering inequality.*

**From Jeannette M. Wing's "Breaking the Cycle"**

How can we break the cycle of deadline-driven research? In computer science, there has been a growing trend in the past decade or so for researchers to publish in workshops and conferences in order to increase the length of their publication list. This situation is especially true of junior faculty, worried about getting tenure; graduate students, worried about getting job interviews; and now even undergraduates, worried about getting into graduate school. In promotions and tenure committee meetings at some schools, discussion of the number of papers can overshadow discussion of the quality and impact of the candidate's work.

We have successfully trained deans and provosts that, in computer science, papers in premier conferences count as much as or more than papers in journals. So, the pressure to publish in conferences is even that much more intense. And presumably the more the better. To accommodate the research capacity of our field, new workshops and conferences (and journals) proliferate, resulting today in extensive Web sites maintaining double-digit rankings of conferences. It is now common practice to see the conference success rate included with each publication listed on a candidate's résumé. (I will not repeat the cogent arguments that others have given on the subject of journals vs. conferences, published in *Communications*,[1–3] but they are relevant to this topic as well.)

We are now in a state where our junior faculty are mentoring graduate students with this deadline-driven approach to research. It's the only value system they know and they are passing it onto the next generation. When one of my own graduate students, after we agreed that we would submit a journal version of our conference paper, said to me, "Jeannette, the author guidelines for *Journal X* don't specify a page limit," I knew something was very wrong with our current culture in computing. We are now in a state where the common thought-chunk of research is a 12-month effort that fits in 12 pages.

We, as faculty advisors, are in a bind: Do we say to our student, "Yes, go ahead and submit to that conference [whose due date is looming]" or "No, don't waste your time writing for that conference. Your work is not ready. Spend the time developing the work"? Do we give in to the peer pressure our students feel, making them potentially less competitive when they are on the job market? We need to promote a culture that encourages faculty and student researchers to take the time needed to work out their ideas so that when they feel ready, they can submit based on the import of their contribution.

Moreover, conservatism tends to win out in program committees, when submissions are competing for a finite number of conference slots, and in panel reviews for funding agencies, when proposals are competing for finite resources. This attitude leaves less room for the bold, creative, risk-taking, visionary ideas, especially those that are not fully fleshed out with all the i's dotted and t's crossed. Note that I have nothing against conferences: they are important for the expeditious exchange of technical ideas, as well as networking among researchers and between academia and industry. I have nothing against (high-quality) incremental research: some research agendas are long-term in vision, but rely on making progress step by step, building on prior research results.

The consequences of this deadline-driven research are potentially bad for the field. Our focus should be on the quality of the research we do. Our goal should be advancing the frontiers of science and engineering.

So how can we break this cycle? One place to start is with the department heads. At hiring time, among other factors, we should look for a candidate's big idea (or two), not number of publications. In mentoring junior faculty, we need to stress the importance of quality and impact. At faculty evaluation time, we should promote and grant tenure based on quality and impact.

Hopefully, we in the community can at least start a dialogue on this topic. It is for the good of our field—to keep it healthy, exciting, and vibrant.

**References**

1. Crowcroft, J., Keshav, S. and McKeown, N. Scaling the academic publication process to Internet scale. *Commun. ACM 52*, 1 (Jan. 2009), 27–30.
2. Vardi, M. Y. Conferences vs. journals in computing research. *Commun. ACM 52*, 5 (May 2009), 5.
3. Birman, K. and Schneider, F. B. Program committee overload in systems. *Commun. ACM 52*, 5 (May 2009), 34–37.

**Reader's comment**

*A key challenge that our community needs to address is how to detect, from an ocean of papers, the key innovative ones that need to be widely distributed. The new* Communications *is a very good step in this direction, but I think that* Communications *will not be sufficient and we'll need innovative techniques to efficiently and quickly detect the most innovative papers in each CS subfield.*

*—Olivier Bonaventure*

### From Mark Guzdial's "CS Faculty Cause Inequality"

An article in the Sept. 8, 2009 edition of the *New York Times* argues that "Colleges Are Failing in Graduation Rates." Specifically, the article claims "[t]he United States does a good job enrolling teenagers in college, but only half of students who enroll end up with a bachelor's degree. Among rich countries, only Italy is worse. That's a big reason inequality has soared, and productivity growth has slowed."

That's a strong claim, that the failing rates in college are actually *causing* inequality. This isn't the first time

**"It's easy to pick the 'best and brightest' who look like us, act like us, and learn like us. The challenge is to identify the students who are *even brighter and better* than us, but don't look like us, act like us, or learn like us." —Mark Guzdial**

the *Times* has made this argument, though. The first time I read the claim that higher-education faculty are a significant cause of the widening gap between the haves and have-nots was in a column in 2005 by David Brooks of the *New York Times*. He explicitly argued that colleges, rather than being a ladder to improving one's life, are actually reducing the opportunities for the poor. Brooks wrote:

"As you doubtless know, as the information age matures, a new sort of stratification is setting in, between those with higher education and those without. College graduates earn nearly twice as much as high school graduates, and people with professional degrees earn nearly twice as much as those with college degrees. But worse, this economic stratification is translating into social stratification. Only 28% of American adults have a college degree, but most of us in this group find ourselves in workplaces in social milieus where almost everybody has been to college… The most damning indictment of our university system is that these poorer kids are graduating from high school in greater numbers. It's when they get to college that they begin failing and dropping out. Thomas Mortenson of the Pell Institute for the Study of Opportunity in Higher Education has collected a mountain of data on growing educational inequality. As he points out, universities have done a wonderful job educating affluent kids since 1980. But they 'have done a terrible job of including those from the bottom half of the family income distribution. In this respect, higher education is now causing most of the growing inequality and strengthening class structure of the United States.' "

CS faculty play a role in this phenomenon. No one who looks at CS1 failure rates could argue that CS doesn't contribute to failing grades. Richard Tapia, in his forward to Jane Margolis' *Stuck in the Shallow End*, makes the argument explicitly starting from his title, "Computer Science is Widening the Education Gap." He wrote:

"Over the years, I have developed an extreme dislike for the expression 'the best and the brightest,' so the authors' discussion of it in the concluding chapter particularly resonated with me. I have seen extremely talented and creative underrepresented minority undergraduate students aggressively excluded from this distinction. While serving on a National Science review panel years back, I learned that to be included in this category you had to have been doing science by the age of 10. Of course, because of lack of opportunities, few underrepresented minorities qualified."

No one is arguing that we should not seek the *actual* "best and the brightest." The real question is how we make that determination and how we develop those students. It's easy to pick the "best and brightest" who look like us, act like us, and learn like us. The challenge is to identify the students who are *even brighter and better* than us, but don't look like us, act like us, or learn like us. With declining enrollment and a population of computer scientists who increasingly have the same gender and represent only a few ethnic groups, we *must* look beyond the simple definitions, find those terrific students whom we might have missed in our first glance, and help them to develop in the ways that best suit them.

**Jeannette M. Wing** is a professor at Carnegie Mellon University. **Mark Guzdial** is a professor at the Georgia Institute of Technology.

DOI:10.1145/1610252.1610256

# In the Virtual Extension

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

### Power and Trust in Global Virtual Teams

*Niki Panteli and Robert Tucker*

Although the current understanding of virtual teams has advanced in significantly over the last few years, the authors contend it has not taken sufficient account of power dynamics within virtual teams nor sought to explore the nature of power within geographically distributed teams. The challenge is to be able to manage power differentials effectively in order to allow collaboration to foster within a virtual team environment. The question addressed in this article is: How is power exercised in global virtual teams and how can it effectively impact trust development and overall team performance in such distributed environments?

### Online Privacy, Government Surveillance, and National ID Cards

*Sun Sun Lim, Hichang Cho, and Milagros Rivera Sanchez*

The authors explore how the online privacy concerns of Internet users are related to their attitudes toward government surveillance and national ID cards. A survey of Internet users in five multinational cities (Bangalore, New York, Seoul, Singapore, and Sydney) found that while positive attitudes toward ID cards raise online privacy concerns, actual experiences using these cards appear to desensitize Internet users to privacy intrusions. This is possibly due to the perceived convenience of these cards, the realization that privacy intrusions are not particularly significant, or a sense of resignation that the use of ID cards will inevitably be accompanied by privacy intrusions.

### Security in Dynamic Web Content Management Systems Applications

*Ganesh Vaidyanathan and Steve Mautone*

Web Content Management Systems (WCMS) allow teams to maintain Web content in a dynamic fashion. While this "on-the-fly" content creation provides Web site authors several advantages, there are distinct disadvantages. Indeed, organizations are adopting information technology without understanding such security concerns. The authors illustrate how to evaluate open source systems and how an evaluation technique in terms of security may be used in an organization to assess a short list of possible WMCS systems. This work focuses on security issues in WCMS with the objective to understand the security issues as well as to provide a generic security framework.

### Assessing Open Source Software as a Scholarly Contribution

*Lou Hafer and Arthur E. Kirkpatrick*

In academic computer science, papers about software are considered scholarship but actual software is not. The authors propose "best practices" for the evaluation of the scholarly contribution of open source software, raising publication of code to a status comparable to publication of the ideas it embodies. Evaluating software as scholarship acknowledges the importance of the application of discoveries. The authors contend software instantiates knowledge in a form that can be applied, unlike the passive knowledge in an article. Considering software as scholarship also advances the scientific principle of reproducibility. Pseudo-code in an article is insufficient for replication; only actual code provides enough detail.

### Why Did Your Project Fail?

*Narciso Cerpa and June Verner*

Most research literature on failed software projects tends to deal with a few high-profile failures. The authors review 70 failed projects to determine what factors lead to project failure. Data was collected from a survey that considered over 80 software development practices. Projects do not fail for a single reason alone; they fail for multiple reasons, including poor estimation and schedule, poor risk management, and lack of staff rewards. The failure factors are different for in-house and outsourced projects, and while organizations do not conduct post-mortem reviews software projects will continue to fail.

### Visual Passwords: Cure-All or Snake-Oil?

*Karen Renaud and Antonella De Angeli*

In our everyday lives we're expected to remember a number of passwords and PINs. Since human memory is finite and fallible, we often forget them. As an industry, we know that the password has outlived its usefulness as an authenticator, but no one has come up with a convincing and viable alternative. Visual passwords have been researched for a decade now. The authors present the different kinds of visual passwords, their pros and cons, and debate whether they do indeed possess the potential to be a viable alternative to the hated password.

### Positive Externality, Increasing Returns, and the Rise in Cybercrimes

*Nir Kshetri*

The distinctive geography of cyberspace provides an ideal environment for engaging in opportunistic behavior. This article employs increasing returns and externalities approaches to explain the escalation of cybercrimes. The author focuses on three positive or self-reinforcing feedback systems to examine increasing returns in cybercrime-related activities. They are related to economic, sociopolitical and cognitive systems. The author also examines three mechanisms that may give positive feedback to cybercriminals: inefficiency and congestion in the law enforcement system, acceleration of the diffusion of cybercrime know-how and technology, and increase in potential criminals' predisposition toward cybercrimes.

### Technical Opinion: Are Employees Putting Your Company At Risk By Not Following Information Security Policies?

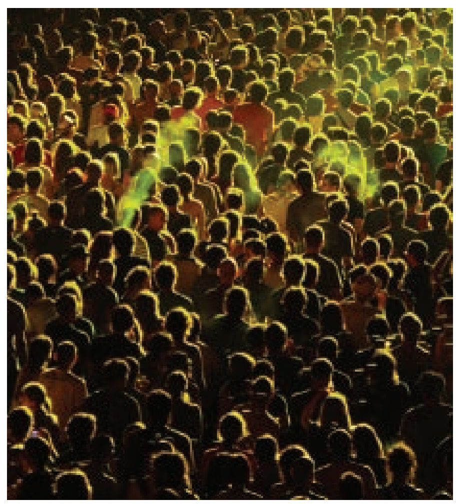
*Mikko Siponen, M. Adam Mahmood, and Seppo Pahnila*

Careless employees who do not follow information security policies constitute a serious threat to information privacy and confidentiality. A field research, conducted to determine which issues are vital toward employees' compliance with these policies, indicates that the visibility of security activities and expectations of peers have a positive impact on employees' ability to assess the severity of security threats. These precursor variables also affect employees' beliefs that the security policies compliance is an effective way to combat these threats. In turn, if employees realize the severity of security threats, they often have a strong intention to comply with security policies.

DOI:10.1145/1610252.1610258

David Roman

# Crowdsourcing and the Question of Expertise



There is an in inherent weakness to crowdsourcing that should bother computer scientists and computer users alike. It's the fact there is no clear difference between "the wisdom of the crowd" and "the mob that rules." What's missing is a measure of discernment.

The Internet is awash in information that demands selectivity, leading *Newsweek* among others to predict the rise of online experts and reliable information (http://www.newsweek.com/id/119091). The assessment seems overly optimistic. There are some efforts to rate expertise on the Internet (http://cacm.acm.org/news/42206), but most of us are left with coping strategies that limit where you go, what you see, and who you trust. It is not the kind of open investigation that promotes learning or understanding.

Crowdsourcing doesn't really help sort through or synthesize information, in fact, it might do the opposite. Research shows that it favors popular opinion and therefore reinforces homogeneity (http://cacm.acm.org/news/42525). That's not hospitable to unconventional or idiosyncratic views.

There is an upside, for sure. Luis von Ahn's GWAP (http://www.gwap.com/gwap/about/) uses computer games "to solve problems for humans all over the world." And Galaxy Zoo tapped about 250,000 visitors to classify nearly one million galaxies (http://cacm.acm.org/magazines/2009/10/42492).

Now the downside: The limitations of crowdsourcing are becoming apparent, even to its defenders. Blogger Josh Berkus summarizes key weaknesses, saying the term is "evil" and carries too much baggage (http://it.toolbox.com/blogs/database-soup/never-say-crowdsourcing-34331). In the end he concludes that the problem is mainly about improper usage. But the issue is bigger than that. The problem with crowdsourcing is that there is no verity. In fact, "correctness [is]...anathema to crowdsourced systems" (http://cacm.acm.org/magazines/2009/7/32094). That's a small concern when rating movies, but researchers and scientists need something more.

Science needs higher standards. This was illustrated by *Newsweek* when it decried science education in the U.S. and showed how "wisdom of the masses" is an oxymoron. It described how John Holdren, director of the White House Office of Science and Technology Policy, trades candor for political timidity when discussing science policy (www.newsweek.com/id/216505). "He must sell his ideas to people who couldn't pass high-school algebra—and who believe they know more than he does."

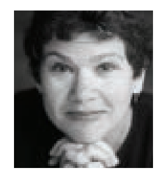
Crowdsourcing empowers followers. It risks weakening leaders.

# ACM Member News

### HALL WINS DUNCAN DAVIES MEDAL

**ACM President Dame Wendy Hall received the Duncan Davies Medal, which is awarded annually by the Research and Development Society to an individual who has made an outstanding contribution toward making the United Kingdom the world's best-performing research and development environment.**

### BERMAN HONORED WITH KEN KENNEDY AWARD



**Francine Berman was awarded the inaugural Ken Kennedy Award from ACM and IEEE Computer Society for "her influential leadership in the design, development, and deployment of national-scale cyberinfrastructure." A vice president for research at Rensselaer Polytechnic Institute, Berman was recognized for her work as a pioneer in grid computing and a leading advocate for the development of a national-scale cyberinfrastructure for the access, use, stewardship, and preservation of the digital data.**

**In an email interview, Berman discussed the current challenges and opportunities in cyberinfrastructure. "There are immense opportunities that focus on the development of cyberinfrastructure to drive innovative solutions for some of the most complex and compelling societal challenges of our age: health care, energy, the environment, safety, and economic stability," said Berman. "Some of the greatest breakthroughs we are now seeing in these areas come from the innovative use of computers, information, sensors, networks, scientific instruments, and other 21st century tools. The challenge is to develop a system to support and deploy cyberinfrastructure as *infrastructure*: sustainable business models, appropriate standards, low-barrier-to-access user interfaces, and interoperability. The development of cyberinfrastructure as infrastructure truly constitutes a grand challenge for our age."**

PHOTOGRAPH BY PABLO TWOSE VALL

news

Science | DOI:10.1145/1610252.1610259

Gary Anthes

# Blueprints for Self-Assembly

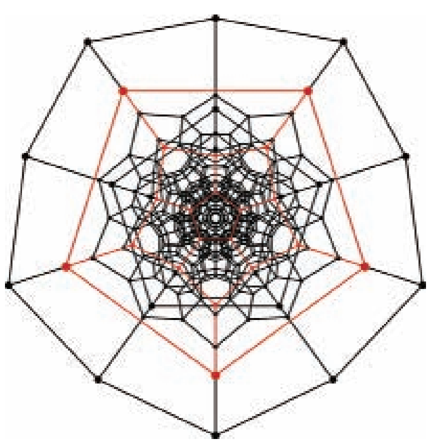
*Researchers are using tools from information theory and computer science to facilitate the automatic creation of nanoscale structures.*

SELF-ASSEMBLY, BY which atoms, molecules, or other nanoscale components spontaneously organize into something useful, sounds so simple. Just mix a few chemicals and wait for a new plastic, drug, or electronic component to form at the bottom of your test tube.

Unfortunately, it's not so easy. Coaxing tiny particles to arrange themselves in an orderly way, with desirable and repeatable properties, is enormously complex, typically involving a great deal of trial and error in the laboratory.

But now two mathematicians, using tools from information theory and computer science, have found a new and relatively simple way to orchestrate the assembly of nanostructures. And they have devised algorithms that can produce mathematical proofs that their structures are optimum.

Henry Cohn, a principal researcher at Microsoft Research New England and Abhinav Kumar, an assistant professor of mathematics at Massachusetts Institute of Technology, have employed a rich mix of techniques—including heuristic algorithms, linear programming, search optimization, and error-correction theory—to produce their results.



**The Schlegel diagram for a regular 120-cell structure (with a dodecahedral facet in red) from Henry Cohn and Abhinav Kumar's self-assembly research.**

Writing in a recent issue of *Proceedings of the National Academy of Sciences*, Cohn and Kumar describe their success in designing a system to direct 20 randomly placed particles on a sphere to form into a perfect dodecahedron with 12 pentagonal faces, a structure that minimizes potential energy and, hence, maximizes stability.

Although the methods have yet to be implemented in a lab, they may ultimately find use in such diverse fields as electronics, communications, and medicine. For example, Cohn says, a drug company might produce a time-release drug by encapsulating tiny drug droplets in structures, such as dodecahedrons, that have certain desired properties. The idea is to simplify the search for the proper materials and conditions for self-assembly of such things.

The forces between particles determine whether or not they will organize into a stable and desired configuration. Cohn and Kumar's blueprints for self-assembly specify the required inter-particle forces and distances via formulas called potential functions.

Traditional approaches to this problem use complex potential functions with multiple potential wells, local energy minima that cause the particles to settle into certain positions. But Cohn and Kumar devised a way to find potential functions that cause the particles to organize themselves more directly, without relying on local minima. The resulting formulas are simpler and, hence, would be much easier to implement in a lab or manufacturing line, they say.

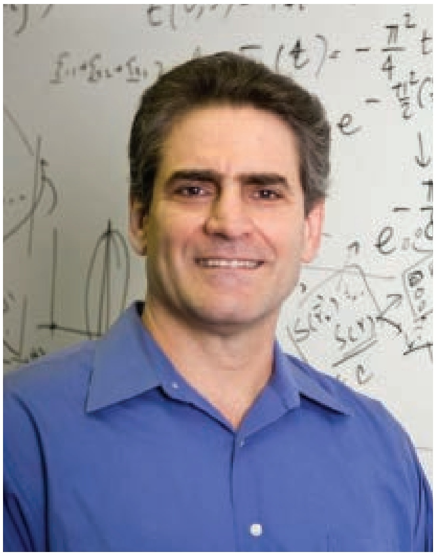
In the second part of the problem, Cohn and Kumar go on to mathematically prove that the dodecahedron and several other structures built by their methods are in fact the unique ground states, or globally energy-minimizing arrangements of particles. They do that using linear programming bounds, a tool borrowed from error-correction theory.

The researchers find their optimal potential functions using an iterative process they call "simulation-guided optimization," which alternates between molecular dynamics simulation and linear programming. A trial potential function is chosen, and a number of simulations—with random starting positions for the particles—are run, allowing the particles to interact until they settle into a structure. The result is a list of possible candidate structures for the ground state.

Then linear programming looks for a new potential function that makes all the candidates rank worse than the target configuration, and the whole process is repeated. The researchers call this process a heuristic algorithm because "it is not guaranteed to work." They tested their potential function 1,200 times with different starting positions of the 20 particles, and all but six converged to the dodecahedron.

Not surprisingly, a process that repetitively applies simulation and linear programming can be computationally very taxing. Indeed, a large part of the mathematicians' effort went into finding efficient search methods. "The problem is how not to get stuck at some sub-optimal solution," Cohn says.

Graphs of potential functions against inter-particle distances for



**Princeton University professor of chemistry Salvatore Torquato, who pioneered an inverse approach to self-assembly.**

Cohn/Kumar solutions tend to be smooth, with particle interactions decreasing monotonically with distance. But those for the more conventional approaches—those that employ potential wells—are complex and bumpy, with numerous local maxima and minima. "The problem with potential wells is they are much harder to manufacture in the lab," Kumar explains. "But the kinds of functions you see in nature, and the kind you might be able to generate, are basically functions with not too many wiggles. We ask the question, 'Can you get a potential function with essentially no wiggles?' "

### The Inverse Approach

Cohn and Kumar's work builds on earlier research by Salvatore Torquato, a professor of chemistry at Princeton University. Starting with a paper written four years ago, Torquato pioneered what he calls the inverse approach to self-assembly. In the traditional forward approach, known particle interactions are used to predict a likely resulting structure. But the inverse method starts with some desired configuration and derives the optimal inter-particle interactions that would spontaneously organize into that target structure. Torquato has used pure theoretical work as well as numeric computer simulation to find potential functions that can lead to the self-assembly of materials into squares, honeycombs, diamond shapes, and lattices.

"This is a completely different way of thinking about designing these structures, and it's tailor-made for self-assembly," he says.

Torquato says the inverse approach

---

Report from the ACM Nominating Committee

# Slate of Nominees for ACM General Election

**In accordance with the Constitution and Bylaws of the ACM, the Nominating Committee hereby submits the following slate of nominees for ACM's officers. In addition to the officers of the ACM, two Members at Large will be elected. The names of the candidates for each office are presented in random order below:**

**PRESIDENT** (7/1/10–6/30/12):

Alain Chesnais,
*SceneCaster.com*

Joseph A. Konstan,
*University of Minnesota*

**VICE PRESIDENT** (7/1/10–6/30/12):

Barbara G. Ryder,
*Virginia Tech*

Norman P. Jouppi,
*Hewlett Packard*

**SECRETARY/TREASURER** (7/1/10–6/30/12):

Alexander L. Wolf,
*Imperial College London*

Carlo Ghezzi,
*Politecnico di Milano*

**MEMBERS AT LARGE** (7/1/10–6/30/14):

Vinton G. Cerf,
*Google*

Fei-Yue Wang,
*Chinese Academy of Sciences-Institute of Automation/ University of Arizona*

Satoshi Matsuoka,
*Tokyo Institute of Technology*

Salil Vadhan,
*Harvard University*

**The Constitution and Bylaws provide that candidates for elected offices of the ACM may also be nominated by petition of one percent of the Members who as of November 1 are eligible to vote for the nominee. Such petitions must be accompanied by a written declaration that the nominee is willing to stand for election. The number of Member signatures required for the offices of President, Vice President, Secretary/Treasurer and Members at Large, is 687.**

**The Bylaws provide that such petitions must reach the Elections Committee before January 31. Original petitions for ACM offices are to be submitted to the ACM Elections Committee, c/o Pat Ryan, COO, ACM Headquarters, 2 Penn Plaza, Suite 701, New York, NY 10121, USA, by January 31, 2010. Duplicate copies of the petitions should also be sent to the Chair of the Elections Committee, Gerry Segal, c/o ACM Headquarters. All candidates nominated by petition are reminded of the requirements stated in the Policy and Procedures on Nominations and Elections that a candidate for high office must meet in order to serve with distinction. Copies of this document are available from Rosemary McGuinness, Office of Policy and Administration, ACM Headquarters. Statements and biographical sketches of all candidates will appear in the May 2010 issue of CACM.**

**The Nominating Committee would like to thank all those who helped us with their suggestions and advice.**

**Stuart Feldman, Chair, Fabrizio Gagliardi, Susan Graham, Mathai Joseph, Lucy Sanders.**

driven by theoretical and computational methods is the most direct way to design materials that can be produced by self-assembly. Some such substances have counterintuitive properties, such as materials that shrink when heated or that, when stretched, expand at right angles to the direction of stretch. The latter property might be useful in a foam designed to provide a tight seal between two surfaces, for example.

"The game is to show that in fact you are not limited to what you get from molecular interactions," Torquato says. "What kinds of interesting structures, that typically nature doesn't make, can you make computationally? We ask the fundamental questions: What's permitted? What's not? We still don't know the answers to those questions."

While researchers using the inverse method increasingly rely on concepts from information theory, those employing the forward approach are more Edisonian, relying more on hands-on, trial-and-error laboratory work, Torquato says. "They synthesize new [substances] and they get what they get at the end of the day," he says. "But the most direct way to get self-assembly is, OK, you want a particular target configuration, then let's design the interactions to get those."

But Torquato says it's not a battle between Edisonians and computer scientists. "The idea would be to combine our computational theoretical approach with experimental synthesis technology, and that, I think, is the way material science will be done in the future."

Torquato hails as a "wonderful contribution" Cohn and Kumar's methods for mathematically proving theorems about their potential functions. That was feasible, he said, because the spaces they considered, such as spheres, were finite and bounded. "Can their approach scale to Euclidian space?" he asks. "There is no reason to think one can't do that. Until they came along, there was no rigorous proof of any kind."

Cohn and Kumar found inspiration in some unlikely recesses of information science, such as the theory behind the error-correction codes used in communication systems. In a noisy communication channel, one wants to keep multiple signals as well separated as possible so they don't get confused. Similarly, in chemistry, particles often repel each other and stay as far apart as possible. Cohn says he was able to apply some of the theory of error-correction coding to the self-assembly problem by using it to tell particles how to stay far apart.

**The research of Cohn and Kumar may have applications for fields such as electronics, communications, and medicine.**

"Some of the most interesting areas for research are where there is an imbalance between two fields, where each field knows something that the other field hasn't figured out yet," Cohn says. "If you can make the connection, you can engage in arbitrage and transfer some information in each direction." ■

**Further Reading**


*Cohn, H. and Kumar, A.*
Algorithmic design of self-assembling structures, *Proceedings of the National Academy of Sciences 106*, 24, June 16, 2009.

*Cohn, H. and Kumar, A.*
Universally optimal distribution of points on spheres, *Journal of the American Mathematical Society 20*, 1, Sept. 5, 2006.

*Torquato, S.*
Inverse optimization techniques for targeted self-assembly, *Soft Matter 5*, 6, March 21, 2009.

*Rechtsman, M. C., Stillinger, F. H., Torquato, S.*
Synthetic diamond and Wurtzite structures self-assemble with isotropic pair interactions, *Physical Review E 75*, 3, id. 031403, March 26, 2007.

*Cohn, H. and Kumar, A.*
Counterintuitive ground states in soft-core models, *Physical Review E 78*, 6, id. 061113, Dec. 12, 2008.


**Gary Anthes** is a technology writer and editor based in Arlington, VA.



Milestones

# CS Awards

**IBM's Blue Gene supercomputers and members of the computer science community were honored for their innovative research.**

### NATIONAL MEDAL OF TECHNOLOGY AND INNOVATION

In a special ceremony at the White House, President Obama honored IBM's Blue Gene line of supercomputers with the National Medal of Technology and Innovation in recognition of its lasting contributions to America's competitiveness, standard of living, and quality of life through technological innovation.

### JIM GRAY eSCIENCE AWARD

University of California, Santa Barbara professor Jeff Dozier was awarded Microsoft Research's second annual Jim Gray eScience Award. Dozier was cited for "his pioneering research on remote sensing, water resources, and climate change, and his contributions to the integration of environmental science and computer science."

### SEYMOUR CRAY AWARD

IEEE Computer Society bestowed its 2009 Seymour Cray Computer Engineering Award to Kenichi Miura, a professor at the National Institute of Informatics in Tokyo, for "unique contributions to the field of computer engineering by bringing a strong background in numerical algorithms and applications to the task of designing systems that deliver high performance on real scientific applications."

### SIDNEY FERNBACH AWARD

IEEE Computer Society presented the 2009 Sidney Fernbach Award to Roberto Car and Michele Parrinello, developers of the Car Parrinello Molecular Dynamics approach. Car, a professor of chemistry at Princeton University, and Parrinello, a professor of computational science at ETH Zurich, were recognized for laying "the foundation for a modern approach to the chemistry and physics of materials. Their methodology was revolutionary, increasing the speed of simulations and propelling a major force in science."

Technology | DOI:10.1145/1610252.1610260

Alex Wright

# Ready for a Web OS?

*A new generation of browsers may finally herald the long-awaited convergence of the Web and operating system.*

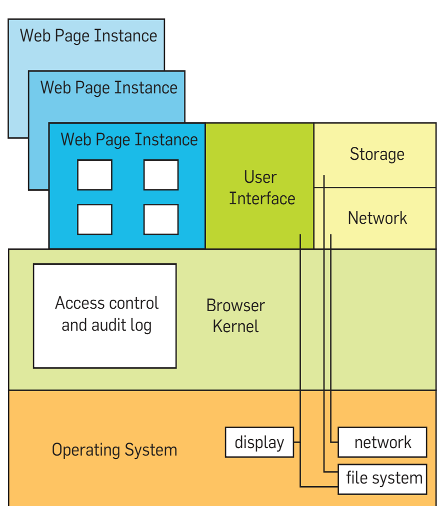
Back in 1995, Netscape cofounder Marc Andreessen predicted that his fledgling Web browser would one day render Windows obsolete. Fifteen years later, Netscape is long gone, and the traditional desktop operating system (OS) remains firmly established on most personal computers. Meanwhile, Web browsers still look a lot like they did in the mid-1990s, running inside application windows. In hindsight, Andreessen may have spoken a bit too soon. But history may yet prove him right.

The hegemony of the desktop OS is starting to fracture with the emergence of a new generation of browsers that may finally herald the long-awaited convergence of Web and OS. An enormous amount of Web OS development is currently under way, with the development of Web standards, such as HTML5, to add richer capabilities and features; new technologies like Microsoft's Xax and Google's Native Client that make browsers and their applications as capable and powerful as desktop applications; and architectural changes to browsers, making them process oriented, which increases their robustness and security.

A Web OS offers enormous promise. Potentially, it could take the best of the Web—the rapid deployment and updating of new applications, device independence, and the ease and convenience with which large communities can collaborate and share information—and combine it with the advantages of desktop applications—operating at machine speed, rich and interactive interfaces, and access to local hardware—and sidestep many of the security and compatibility issues currently plaguing desktop OSs. Before the Web OS becomes a practical reality, however, browser developers must overcome several major obstacles to security and device integration that continue to tilt the balance of power in favor of the desktop OS.

For many average computer users, the browser has become their de facto OS—a tool of choice for e-mail, personal finance, and other activities that were once the domain of desktop applications. Today's Web has come a long way from its original incarnation as a collection of passive, hyperlinked documents. Web developers now routinely use sophisticated scripting languages and other active client-side technologies to provide users with rich experiences that approximate the performance of desktop applications, including features like drag-and-drop, keyboard shortcuts, and other desktop-like affordances that have become commonplace.

The latest Web browsers include powerful features that further close the gap between Web applications and native desktop applications. Most major browsers have significantly increased the speed of their JavaScript engines, allowing more complex and computationally demanding applications to be developed. And Web applications will soon benefit from evolved Web standards, such as HTML5, featuring offline support, local storage, geolocation capabilities, graphics acceleration, and perhaps access to client devices, such as a scanner or video camera.




**To improve the performance and security of the OP Web browser, the main architecture is divided into five main subsystems: browser kernel, storage subsystem, network subsystem, user-interface subsystem, and Web page instances.**

"With HTML5, we're going to see a new generation of rich Web applications," says Adam Barth, a postdoctoral fellow at the University of California at Berkeley who focuses on privacy and browser security. "But I suspect it will take a while for application developers to realize the full potential of the various HTML5 technologies like canvas, local storage, and video."

In a similar vein, experimental technologies, such as Xax and Native Client, allow Web publishers to implement Web programs as native x86 code that executes directly and safely on the client's processor, eliminating the interpretation or compilation overhead of scripted or byte-coded languages and frameworks such as Java, JavaScript, Silverlight, and Flash.

"I think we're going to see a proliferation of different scripting languages in the browser," says Barth. "In the past, JavaScript had a mortal lock as the lingua franca of the Web, but now, with technologies like Xax and Native Client, anyone can write an interpreter for their favorite language and run the interpreter in the browser without pestering the user to install the interpreter."

Sam King, assistant professor in the computer science department at the University of Illinois at Urbana-Champaign, thinks JavaScript is too entrenched in today's Web ecosystem to cede its dominant position anytime soon. "JavaScript is tightly integrated into modern browsers and a fundamental part of the current Web," he says. "However, as technologies such as Xax and Native Client show up in

FIGURE: CHRIS GRIER, SHUO TANG, AND SAMUEL T. KING, "SECURE WEB BROWSING WITH THE OP WEB BROWSER", PROCEEDINGS OF THE 2008 IEEE SYMPOSIUM ON SECURITY AND PRIVACY (OAKLAND), MAY 2008.

browsers, Java, Silverlight, and Flash will certainly have new competition."

**Multiple Concerns**

As Web sites take advantage of improved client-side technologies, browsers will need to start coping with a growing range of performance, reliability, and security concerns. "As we make Web browsers more powerful, we need to keep in mind how malicious Web sites might abuse that additional power," says Barth. For example, a truly secure browser should let users visit sites that contain buggy or malicious code without fearing about the integrity of their OS, applications, or private data.

To address these concerns, researchers are reconsidering the underlying architecture of the Web browser. New multiprocess browsers, such as Tahoma, Google Chrome, Internet Explorer 8, and the OP browser, place separate Web applications in their own operating system processes or virtual machines, allowing the underlying host operating system or virtual machine monitor to ensure that crashes and slowdowns in one Web application do not affect the performance or robustness of other applications. "By decomposing the browser, you can separate out the security logic from the implementation," says King.

To improve security, browsers such as Chrome and Internet Explorer 8 have been refactored to run untrusted components and Web code in a low-privilege sandbox, limiting the exposure of browser vulnerabilities and the damage that can be inflicted by a malicious Web application. Mozilla is exploring a similar protection mechanism in its Electrosis project that will strengthen security features in a future version of Firefox. Meanwhile, Microsoft Research developers have taken this approach a step further with the prototype Gazelle browser, which separates different Web sites into discrete protection domains. With the OP browser, researchers at the University of Illinois at Urbana-Champaign explored applying OS principles to Web browser design by breaking the browser program into smaller subsystems.

By isolating processes within a browser, browsers can mediate interactions between Web programs, such as mashups and embedded widgets, to prevent one program from stealing information from or causing damaging side effects to other programs. However, the techniques that accomplish this often come at the expense of backward compatibility, making them more difficult to deploy.

**As Web sites take advantage of improved client-side technologies, browsers must cope with a growing range of performance, reliability, and security issues.**

Looking further ahead, questions of compatibility will continue to arise as browsers must negotiate a growing tangle of local computing resources such as offline storage, cameras, microphones, geolocation, and graphics acceleration hardware—all of which have their own security and performance issues.

Geolocation provides an instructive example of the types of challenges that browser developers will likely face in the near future. "Geolocation is an interesting case from a security point of view because we're resorting to asking users to grant Web sites additional privileges to read location data," says King. Developers must negotiate delicate trade-offs in giving users an appropriate level of control without letting them also damage their systems. "Developing a security policy for accessing local data and hardware is still an open, difficult, and important research problem," he says.

The continued proliferation of devices—each with its own OS and interface—may provide further impetus toward a consistent, predictable Web OS. With new devices like the Palm Pre smartphone, the CrunchPad Web tablet, and various netbooks running Google's forthcoming Chrome OS, all user interaction will take place through a browser or Web-based applications. While each of these devices may still have a different OS providing a scaffolding of background processes, users will increasingly experience these devices through the filter of a Web interface.

As developers take advantage of these emerging technologies to craft Web-based experiences across a growing range of devices, the OS will likely continue to recede from users' awareness—and perhaps eventually disappear altogether. But even if the traditional OS sticks around in some form for years to come, it may not matter much to anyone except developers. "I'm not sure users care that much about the computing platform," says Barth. "Users seem to care much more about what they can do with technology than how it gets done." ■

**Further Reading**


*Barth, A., Jackson, C., Reis, C., and the Google Chrome team*
**The Security Architecture of the Chromium Browser, http://seclab.stanford.edu/websec/chromium/, 2008.**

*Cox, R. S., Hansen, J. G., Gribble, S. D., and Levy, H. M.*
**A safety-oriented platform for web applications, *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2006.**

*Grier, C., Tang, S., and King, S.T.*
**Secure web browsing with the OP web browser, *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2008.**

*Reis, C. and Gribble, S. D.*
**Isolating web programs in modern browser architectures, *Proceedings of the 4th ACM European Conference on Computer Systems* (EuroSys 2009), Nuremberg, Germany, March 2009.**

*Wang, H. J., Grier, C., Moshchuk, A., King, S. T., Choudhury, P., and Vente, H.*
**The multi-principal OS construction of the Gazelle web browser, *Proceedings of the 18th USENIX Security Symposium*, Montreal, Canada, August 2009.**


**Alex Wright** is a Brooklyn-based writer and information architect. Charles Reis, Steve Gribble, and Hank Levy of Washington University contributed to the development of this article.

Society | DOI:10.1145/1610252.1610261

Samuel Greengard

# Making Automation Work

*Today's automated systems provide enormous safety and convenience. However, when glitches, problems, or breakdowns occur, the results can be catastrophic.*

The June 22, 2009 Metro subway train crash in Washington, D.C., in which nine people died and 80 were injured, is the deadliest accident in the Metro's 33-year history.

PHOTOGRAPH BY PABLO MARTINEZ MONSIVAIS/AP PHOTO

It's no secret that engineers and designers constantly seek to build safer and more convenient systems. And, over the last century, planes, trains, automobiles, and industrial machines have become far more automated and efficient. However, when a Metro subway train rammed into another train in Washington, D.C. last June, designers had to confront the unpleasant reality that automation may have been the cause. The accident, which killed nine people and injured 80, may have been rooted in a computer malfunction and the operator's inability to manually apply the brakes quickly enough.

The Metro train accident lies at the heart of what human factors experts refer to as the "automation paradox." As automated systems become increasingly reliable and efficient, the more likely it is that human operators will mentally "switch off" and rely upon the automated system. And as the automated system becomes more complex, the odds of an accident or mishap may diminish, but the severity of a failure is often amplified.

As John D. Lee, a professor of industrial and systems engineering at the University of Wisconsin at Madison told the *Washington Post*: "The better you make the automation, the more difficult it is to guard against these catastrophic failures...."

Understanding how people and machines interact is infinitely complex. Programming all the various possibilities and scenarios into a system can tax even the best design and engineering experts. What's more, as technology evolves, the entire process grows more convoluted and iterative. In some cases, experts say, it's wise to ask what purpose automation serves and when it's best to use it and eschew it.

What is the fallout from automation glitches? Where do programmers, designers, and engineers typically fall short? And what can technologists do to build better systems? There are no simple solutions. But as Donald Norman, professor of computer science at Northwestern University, co-founder of Neilson Norman Group, and author of *The Design of Future Things*, says, "Designers often make assumptions or act on incomplete information. They simply don't anticipate how systems will be used and how unanticipated events and consequences will occur."

## Human-Machine Interface

It's clear that automation has provided enormous gains to society. Safer and more efficient factories; faster police, emergency, and fire response; and more user-friendly and safer automobiles are only a few of the benefits.

Yet, at the same time, it takes little effort to find evidence of breakdowns between human and machine.

The crash of Air France Flight 447 that occurred over the Atlantic Ocean last June—killing all 228 people aboard—may have been caused by a malfunction in a speed sensor. The plane's Pitot tubes, a pressure measurement instrument used to track fluid flow velocity, may have became blocked by ice. At that point, they may have stopped emitting signals, and experts say that the pilots could have encountered false speed readings. In fact, the jet—which was coping with a series of storms, including a severe thunderstorm—reportedly relayed a signal that its computer system no longer knew the speed of the aircraft, and that automatic pilot and thrust functions were switched off. This may have forced the pilots to take over manual control during chaotic, if not impossible, flying conditions.

There are also plenty of examples of humans having trouble with automation systems in everyday life. As automobiles become more automated, new problems crop up. For instance, motorists blindly follow the incorrect directions provided by a navigation system, even though a glance at the road would indicate there's an obvious error. A few motorists have even driven off a cliff or into oncoming traffic after following directions explicitly. What's more, studies show that many motorists use automation features, such as adaptive cruise control, incorrectly. In some cases, Norman says, these automated systems cause the car to speed up as motorists exit a highway because there's suddenly no car in front. If a driver isn't paying attention, an accident can occur.

In the case of airplane pilots and train operators, one solution is regular training sessions in which the pilot or operator is required to turn off their automated system and operate everything manually. This can help them retain their skills and alertness.

But even this is not likely to eliminate breakdowns. Human-machine interface failures occur for a number of reasons, experts say. Sometimes, designers rely on a wrong set of assumptions to build a system. They simply don't understand the way people use technology or the cultural differences that occur. In some instances, thousands and sometimes millions of variables exist and capturing everything in a single algorithm is exceedingly difficult. In fact, Norman argues that machine logic doesn't necessarily jibe with the human brain. "If you look at 'human error' it almost always occurs when people are forced to think and act like machines," he says.

Worse, complex algorithms often prompt humans to relate to devices as if they were fellow human beings. As a result, the autopilot on a plane, the cruise control on a car, and automated speed-control systems in mass transit become either aids or crutches, depending the situation.

Too often, the sum of a system is not equal to the individual parts, says Sidney W. A. Dekker, director of research at the Leonardo da Vinci Center for Complexity and Systems Thinking at Lund University in Sweden. "There is often a great deal of human intuition involved in a process or activity and that's not something a machine can easily duplicate," says Dekker. "If you look at delivering babies, there's a reason we have midwives and nurses. Machines can monitor and help, but they can't detect subtle signs and they're unable to adapt to situations as seamlessly."

David D. Woods, professor of cognitive engineering at Ohio State University, says that designers can easily succumb to the trap of thinking "a little more technology will solve the problem." However, understanding variables and identifying possible exceptions and disruptions is paramount. For example, when the Metro D.C. train crashed, it may have been due to wet leaves on the tracks and a computerized system that wasn't programmed for such a scenario. "The automation system functioned as it was designed," Woods says. "The situation simply fell outside the model of what engineers envisioned."

### Beyond Failure

Make no mistake, human factors experts constantly scrutinize automation. Many believe that if human error exists, it falls on the shoulders of those engineering, designing, and programming technology. "In reality, there is no such thing as operator error. Too often, systems aren't designed as whole and those creating them overlook important factors," argues Nancy Leveson, professor of aeronautics and astronautics at Massachusetts Institute of Technology and author of the forthcoming book *Engineering a Safer World*.

Yet, progress is taking place. Consider the airline industry: In 1989, 1.4 crashes per 1 million departures occurred. By 2008, the number had dropped to 0.2 fatal accidents per 1 million departures. In fact, crashes have steadily dropped over the decades while survivability has increased. Dekker, who is a pilot and has flown various aircraft, including a Boeing 737, says that the industry has gotten serious about stamping out flaws, bugs, and oversights.

These improvements have taken place because the airline industry has moved beyond studying ergonomics and discreet processes. In fact, Leveson says that researchers have put a microscope to cognitive functions, psychology, cultural issues, and a variety of other components that comprise human factors. "They have evolved toward a system view and worked to understand how everything—hardware, software, procedures, and humans—interact. It's a model that other industries must embrace," she says.

One thing is certain: Automation disconnects won't disappear anytime soon. Leveson believes that, ultimately, the people designing systems must take a more holistic view and get past the notion that when a problem or breakdown occurs it's a result of "human error." She believes that universities must place a greater focus on human factors and that programmers and others must understand that, without a big picture view of what they are building, the end result will continually fall short.

Others, such as Dekker, argue that society must examine larger issues, including whether automation automatically translates into progress. "In reality, not every function or process is best automated," he says. "In some cases, automation simply creates new or different tasks and doesn't provide any real benefit." Automation may also change processes to the point where people are more confused and entirely new social dynamics take place. At that point, he says, designers may attempt to add new features, which only ratchet up confusion and complexity further.

To be sure, imperfect people continue to build imperfect systems. The need to focus on human-machine interfaces has never been greater. "Designers, engineers, programmers, and others must take an expansive view of automation and understand all the possibilities and variables," concludes Norman. "Only then can we build systems that improve performance and solve real-world problems." C


#### Further Reading

*Bainbridge, L.*
**Ironies of automation. *New Technology and Human Error*, J. Rasmussen, K. Duncan, J. Leplat (Eds.). Wiley, Chichester, U.K., 1987.**

*Dekker, S.*
***The Field Guide to Understanding Human Error.* Ashgate Publishing, Farnham, Surrey, U.K., 2006.**

*Dekker, S.*
***The Field Guide to Human Error Investigations.* Ashgate Publishing, Farnham, Surrey, U.K., 2002.**

*Norman, D.*
***The Design of Future Things.* Basic Books, New York, 2009.**

*Sarter, N. B., Woods, D. D., and Billings, C. E.*
**Automation surprises. *Handbook of Human Factors and Ergonomics* (3rd ed.). Wiley, New York, 2006.**


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

Conference | DOI:10.1145/1610252.1610262

Leah Hoffmann

# Problem Solvers

*This year's Grace Hopper Celebration focused on using technology for social good.*

Over the past 15 years, the Grace Hopper Celebration of Women in Computing has become one of the industry's premier forums for women in computer science. Cofounded in 1994 by Anita Borg and Telle Whitney and inspired by the legacy of Grace Murray Hopper, the conference balances a broad range of technical talks with professional and personal programs. Once a modest gathering of several hundred women, it has since grown into a large, four-day affair—held this past October in Tucson, AZ—with a tightly packed schedule and more than 1,600 attendees from 23 countries.

Yet the experience remains both powerful and intimate. "There's always a lot of excitement," says Valerie Barr, chair of the computer science department at Union College. "There's an air of joy and celebration." Barr has attended all but one of the previous nine conferences. Like many attendees, Barr cites the friends she has made as one of the primary reasons she comes back year after year. These friendships are actively encouraged by the conference leadership, who challenged women to introduce themselves to five new people each day and harnessed social media like Facebook, Twitter, and blogs to spread the message. Last year's innovative CONNECT program, which enables attendees to use special scannable bar codes to automatically exchange contact information with people, was back again to facilitate further networking.

This year's theme was "Creating Technology for Social Good," and panels, papers, and speeches showcased diverse examples of collaboration and accomplishment. Keynote speakers Megan Smith, vice president of new business development at Google.org, and Francine Berman, vice president for research at Rensselaer Polytechnic Institute, spoke about increasing Internet availability in developing countries and harnessing data to further projects in science and engineering. Other technical topics included human-centered design, communications security and policy, and wireless ad hoc networks. "Grace Hopper demonstrates that you can approach computer science in a way that's relevant to people—that addresses hunger, poverty, environmental issues, and so on," explains Barr. Because of the breadth of its scope, Barr says, the conference is also a great opportunity to learn about breaking research that's outside her specialty.

**From left, clockwise, Carleton University Ph.D. candidate Natalia Villanueva-Rosales, ACM President Dame Wendy Hall, and Google.org Vice President Megan Smith.**

PHOTOGRAPH BY GAIL CARMICHAEL / HTTP://COMPSCIGAIL.BLOGSPOT.COM

One of the year's best-received new features was a technical track that was devoted to robotics. "It spanned four sessions on a single day and seemed to work quite well," says conference chair Heidi Kvinge, an Intel software engineering manager.

Students comprise nearly half of the attendees, and much of the non-technical program revolves around professional development. "We always focus on the pipeline," says Kvinge. "We want to build the next generation of leaders." There were résumé clinics, leadership workshops, and panels on issues women face in industry and academia. The Computer Research Association's Committee on the Status of Women sponsored three career-building sessions for undergraduates, grad students, and early-career researchers.

Industry representatives were also invited to set up booths in an exhibition hall to showcase their companies and answer questions. "Grace Hopper is a great place to recruit people," says Tessa Lau, a researcher at the IBM Almaden Research Center. The women at Grace Hopper, says Lau, are motivated, smart, and enthusiastic about the field. The conference, meanwhile, gives them a unique opportunity to talk to corporate researchers rather than professional recruiters.

"I always come back energized and full of ideas," says Kvinge. "It broadens my vision."

**Leah Hoffmann** is a Brooklyn-based technology writer.

# CALL FOR PARTICIPATION

# CTS 2010

## Chicago, Illinois, USA

## The 2010 International Symposium on Collaborative Technologies and Systems

**May 17 – 21, 2010**
**The Westin Lombard Yorktown Center Hotel**
**Chicago, Illinois, USA**

**Important Dates:**

- Paper Submission Deadline ---------------------------------------- **December 20, 2009**
- Workshop/Special Session Proposal Deadline ----------------- **November 15, 2009**
- Tutorial/Demo/Panel Proposal Deadline ------------------------ **January 8, 2010**
- Notification of Acceptance ------------------------------------------ **February 1, 2010**
- Final Papers Due ---------------------------------------------------- **March 1, 2010**

**For more information, visit the CTS 2010 web site at:**
**http://cisedu.us/cis/cts/10/main/callForPapers.jsp**

**In cooperation with the ACM, IEEE, IFIP**

DOI:10.1145/1610252.1610263

Richard E. Ladner

# Broadening Participation
## Opening Remarks

*Highlighting efforts and providing the rationale to increase the participation and success of underrepresented groups in computing.*

This is the first in a new *Communications* Viewpoints Broadening Participation column series that will be edited by me. The purpose of the column is inform, invigorate conversation, and inspire action in the computing community concerning issues in broadening participation. This column will tend to be U.S.-centric because broadening participation in the U.S. is what I know best. What I mean by broadening participation in the U.S. is increasing the inclusion of individuals from underrepresented groups, such as women, Native Americans, African Americans, Hispanics, and persons with disabilities, in the computing field at all levels. In other countries and regions of the world underrepresented groups would be defined differently. I distinguish "broadening participation" from "diversity," which is a much broader concept. Diversity refers to the variety of backgrounds and life experiences of individuals including geographical, cultural, economic, and other differences, regardless of underrepresentation. The computing field has tremendous diversity. As an example, more than half of the computer science doctorate degrees in the U.S. are earned by people from countries other than the U.S.

University of Washington Computer Science Professor Richard Ladner, left, signs with some of the participants in the first Summer Academy for Advancing Deaf and Hard of Hearing in Computing, held during the summer of 2007.

PHOTOGRAPH BY MARY LEVIN, UNIVERSITY OF WASHINGTON

### My Qualifications

Readers might wonder why I am qualified to edit a column on broadening participation: I am a white male, not a minority and not disabled. Nonetheless, I have a particular sensitivity to issues involving broadening participation because of my life history as a son of deaf parents who were part of the deaf community. You might not think of deaf people as a minority group but the construct of oppressed minority group for the deaf community does fit in many ways, as brilliantly described in Harlan Lane's book *The Mask of Benevolence: Disabling the Deaf Community*. It is an often-overlooked historical fact that in 1880 the International Congress on Education of the Deaf essentially banned sign language from schools for the deaf worldwide in favor of the oral method, which stressed only lip-reading and speech. At that time there was no realization that sign language was a natural language with its own temporal/spatial grammar that was efficient and complete in its visual modality. Those who banned sign language were not deaf, but believed they knew what was best for deaf people by promoting their integration into the hearing world.

My parents were very successful as teachers of deaf children and in most other aspects of life. They were bilingual, fluent in American Sign Language and English. They were able to lip-read and their speech was distorted, but they were highly educated with a remarkable command of English in written form. Nonetheless, they always felt the sting of paternalism when others treated them as if they couldn't take care of themselves or were not qualified for jobs held by hearing people who they knew were less qualified. Future columns will explore people with disabilities in computing in more depth, but in this first column my intent is to provide readers with an understanding of background informing my perspective on broadening participation.

## Reasons for Broadening Participation

There are a number of reasons why broadening participation is important to the computing field, the most significant of which include numbers, social justice, and quality.

*Numbers.* In spite of several downturns in the economy since the invention of computers and the dot-com bust in the early 2000s the demand for computer professionals worldwide has continued to grow, sometimes extremely rapidly and other times more slowly, but always growing. The computing field cannot continue to rely on getting the vast majority of its high-tech workers from a few demographic groups. As an example, the America Competes Act of 2007 has provisions and authorizes spending to help increase the number of women and minorities in science, technology, engineering, and math (STEM) fields.

**U.S. Ph.D. demographics by gender, race and ethnicity, and disability.**

| | Computer Science | STEM | U.S. Population |
|---|---|---|---|
| Women | 21% | 40% | 50% |
| African American | 2% | 2% | 12% |
| Native American | 0% | 0% | 1% |
| Asian | 12% | 4% | 4% |
| Hispanic | 1% | 3% | 15% |
| White (Non-Hispanic) | 29% | 42% | 74% |
| Disabled | 1% | 1% | 15% |
| Non-resident Alien | 56% | 26% | 6% |

*Social Justice.* An important principle in a democratic society is equality, that is, everyone enjoys the same rights and is obligated by the same responsibilities. The vast public education system in the U.S. attempts to provide as many children as possible with a basic education to help them become contributors to the social well being and to learn their rights and responsibilities as citizens. Through laws enacted by legislatures and interpretations of laws by courts, the U.S. adheres to the principle of equal opportunity whereby citizens should not be discriminated against because of gender, minority status, or disability. Preventing discrimination does not guarantee proportional participation, but it does at least level the playing field.

There is well-documented evidence of implicit bias against women and minorities in academia.[1,2] Those of us in computer science are typically highly rational; perhaps having the belief that our reasoning power makes us immune to bias. Our reasoning power only masks our biases with a cloak of rationalization. Everyone has biases depending on their own upbringing and circumstances. Acting on those biases consciously or unconsciously may be discrimination, some of which could break anti-discrimination laws. The Project Implicit (https://implicit.harvard.edu/implicit) is a good starting point to learn about implicit bias and to take a test that demonstrates your own implicit biases.

**I distinguish "broadening participation" from "diversity," which is a much broader concept.**

*Quality.* The strongest reason for broadening participation is quality. The argument is similar for why diversity is important to quality. William W. Wulf, the former president of the American Academy of Engineers, explained this very well: "I believe that engineering is a highly creative profession. Research tells us that creativity does not spring from nothing; it is grounded in our life experiences, and hence limited by those experiences. Lacking diversity on an engineering team, we limit the set of solutions that will be considered and we may not find the best, the *elegant* solution."[3]

Better solutions are more likely to emerge if there is a diversity of points of view contributing to the solutions. This is one of the reasons most major companies try to diversify their work forces: it gives them a competitive advantage. Broadening participation helps our field because it brings in more people with a variety of backgrounds, and at the same time levels the playing field for those who are traditionally underrepresented.

## Demographics

It is important to understand the current circumstances with broadening participation, particularly at the end of the pipeline when students receive their Ph.D.s. The table here gives some approximate percentages for computer science Ph.D.s, STEM Ph.D.s, and the U.S. population. The computer science numbers, except for disabled, come from the 2008 Taulbee Survey (see http://www.cra.org/CRN/articles/may09/taulbee.html), the STEM numbers and disabled number for CS come from the 2007 Survey of Earned Doctorates, and the U.S. population numbers come from the 2000 U.S. Census. All numbers are rounded to the nearest full percent.

As indicated in the table, there is underrepresentation at the Ph.D. level by women, African Americans, Native Americans, Hispanics, and persons with disabilities. There has been a steady growth in the percentage of women earning Ph.D.s in computer science from the early 1970s from approximately 10% to now just over 20%. By contrast, during the same period the percentage of women earning Ph.D.s in biology rose from less than 10% to now just over 50%.

**The computing field cannot continue to rely on getting the vast majority of its high-tech workers from a few demographic groups.**

### Broadening Participation Efforts

The lack of women in computer science at all levels has motivated a number of efforts including ACM's Committee on Women in Computing (ACM-W), Computer Research Association's Committee on the Status of Women in Computing Research (CRA-W), the Anita Borg Foundation for Women and Technology (ABI), and the National Center for Women and Information Technology (NCWIT). These organizations provide a number of programs at all levels to promote and support the inclusion of more women in the computing field. The Grace Hopper Celebration of Women in Computing brings women in the computing field together every year.

The Coalition to Diversify Computing (CDC), which is sponsored by ACM, CRA, and IEEE, has the goal to increase the participation of underrepresented minorities in computing. The CDC organizes the Tapia conference every two years.

The ACM Special Interest Group on Accessible Computing (SIGACCESS) has as one of its missions to "educate the public to support careers for disabled persons." The AccessComputing Alliance (see http://www.washington.edu/accesscomputing) based at the University of Washington is a primary resource for students with disabilities who are interested in computing and for computing departments to make themselves more welcoming for such students. Nevertheless, there is no organization within the major computing organizations (ACM, CRA, and IEEE Computer Society) that advocates for broadening participation for persons with disabilities.

Since 1994, the CRA's A. Nico Habermann Award has been awarded to people who have "made outstanding contributions aimed at increasing the numbers and/or successes of underrepresented members in the computing research community." The Supercomputer Conference (SC) is the only mainstream conference I know of that includes a section on "Broader Engagement" that is "aimed at increasing the involvement of individuals who have been traditionally underrepresented in the high-performance computing (HPC) field."

### Broadening Participation at the National Science Foundation

In 2006, the National Science Foundation initiated a new program called Broadening Participation in Computing (BPC) within the Division of Computer and Information Science and Engineering (CISE). This program funds a small number of alliances and a larger number of demonstration projects. All the projects have the goal of trying to increase the participation and success of underrepresented groups in computing. Some alliances partner research universities with minority-serving institutions to get more undergraduates involved in research. Some alliances are regional, taking advantage of locality to strengthen the alliance, while others are national, centered around one topic such as robotics or students with disabilities. Demonstration projects are more narrowly focused on developing specific interventions to enthuse and build the capacity of students from underrepresented groups in the computing field. A new initiative in BPC that started this year are the Leveraging, Scaling, or Adapting (LSA) projects where a project does not have to be a new intervention, but can be an existing intervention that has been proven for one group and could be applied to another.

### Conclusion

I hope this premiere Broadening Participation has provided some background while suggesting future columns. In this spirit, there have been some very encouraging developments over the past few years. Frances Allen (2006 recipient) and Barbara Liskov (2008 recipient) were the first women to win the ACM Turing Award after the award had been given to 50 men since its inception. The inaugural ACM-Infosys Foundation Award in the Computing Sciences was awarded to a woman, Daphne Koller, in 2007. The Grace Hopper and Tapia conferences are growing and very successful. Broadening participation is beginning to appear as a theme in mainstream conferences, and this column will contribute to calling attention to the importance of this evolving topic in computing.


**References**

1. Greenwald, A.G. and Krieger, L.H. Implicit bias: Scientific foundations. *California Law Review 94* (2006), 945–967.
2. Nosek, B.A. et al. National differences in gender-science stereotypes predict national sex differences in science and math achievement. In *Proceedings of the National Academy of Sciences 106* (2009), 10593–10597.
3. Wulf, W.A. How shall we satisfy the long-term educational needs of engineers? In *Proceedings of the IEEE 88*, 4 (2000), 593–596.


**Richard E. Ladner** (ladner@cs.washington.edu) is a Boeing Professor in Computer Science and Engineering at the University of Washington, Seattle, WA. He is a recipient of the 2004 Presidential Award for Excellence in Science, Mathematics, and Engineering Mentoring (PAESMEM), the recipient of the 2008 Computing Research Association's A. Nico Habermann Award, and a winner of a 2008 Purpose Prize.

DOI:10.1145/1610252.1610264

Orna Berry and Yigal Grayeff

# Emerging Markets
# Israel's Technology Industry as an Economic Growth Engine

*How government-industry collaboration can have far-reaching economic influences.*

OVER THE PAST 40 years, Israel's economy has transformed from being closed and poor into one that is open, developed, and driven forward by a high-tech sector that is well regarded around the world. Much of the credit for this can be attributed to successive Israeli governments for realizing that a civilian research and development industry could become an engine of economic growth, and then implementing policies to create it and ensure its expansion. However, the government's actions vis-à-vis the high-tech sector—and by extension the wider Israeli economy—stand at a critical juncture. As strong as the industry is, concerns exist about its sustainability in the face of the global economic slowdown and a weakening of the structural components underpinning its success. The government, as the achievements of its policies over the last 40 years demonstrate, and particularly the previous 20, has the ability to help surmount these challenges and preserve Israel's technological edge in order to ensure renewed progress in the decades to come.


**Microsoft Corporation's Israel Research and Development Center was inaugurated in 2006.**

PHOTOGRAPH BY ARDEN PENNELL

## Policy Success

Israel has implemented numerous initiatives and policies to leverage the technology sector as a vehicle for economic growth, beginning in 1969 with the establishment of the Office of the Chief Scientist (OCS), which has become the main instrument for fostering innovation and manages several programs that offer grants and other assistance to high-tech companies. The state also provides tax breaks and additional benefits to attract overseas investment in technology firms and encourage foreign companies to establish operations in Israel.

The success of government R&D policy can be seen in a number of ways. On a quantitative basis, Saul Lach of the Hebrew University in Jerusalem has demonstrated that the return to the economy of state investment in high-tech ranges from 473% to more than 1,000%.[a] More broadly, the government achieved its goal of building a robust high-tech sector. From a small base even as late as 1993, there are now approximately 4,000 high-tech companies, one of the highest concentrations of such firms outside of Silicon Valley. These companies employ almost 250,000 staff, while the segment's share of business sector employment in 2007 was around 9%. An indicator of the strength of the industry is the presence of multinational corporations such as Intel, IBM, Motorola, and Microsoft, which provide experience to their staff in managing on a wide scale, completing big projects, marketing, and dealing with large customer accounts.

The growth in the high-tech sector accelerated after 1993 following the

a The research was conducted by the research firm Applied Economics (http://www.applied.co.il) with Saul Lach's academic supervision.

implementation of Yozma, a program designed to create a venture capital industry and make large-scale start-up financing readily available for the first time. Since then, VC firms have raised $13 billion, mostly from overseas, and backed companies that have accounted for $35 billion in exits. Most exits have been in the form of initial public offerings on exchanges abroad or sales to foreign acquirers, with the latter accelerating the increase in the presence of overseas companies in Israel. These successes have provided VCs with the returns that have enabled them to raise further funds and invest in more high-tech businesses, thus ensuring the continued development of the sector.

In addition, the high-tech industry, as per the government's aim, has become an engine of economic growth. Between 1995 and 2007, while Israel's gross domestic product increased two-and-a-half times to $164 billion, high-tech GDP more than tripled to approximately $18 billion, or more than 10% of the total figure. From 1995–2008, high-tech exports jumped more than fourfold from $5 billion to $23 billion, with their contribution to total exports growing from 20% to 32% respectively. Furthermore, the expansion of the technology sector has helped create of tens of thousands of jobs in ancillary fields such as law firms and accountancy practices.

### The Work Force

There are many reasons why Israel's innovation industries have become an integral part of the economy. One factor is the work force: per capita, Israel is among the leading countries in the world for the number of engineers, Ph.D.s, patents, and citizens with a tertiary education, as well as for the standard of its research institutions as ranked by the World Economic Forum. In addition, Israelis enter the work force at a higher average age than in other countries after having completed compulsory national service, where they develop problem-solving, leadership and teamwork skills, and take on significant responsibilities. These elements have fostered a culture of innovation that has led to the pioneering of widely used technology such as Voice over Internet Protocol (VoIP), Internet firewalls, voicemail, Intel's Centrino Wi-Fi chips, and the disk-on-key flash memory device. The caliber of the work force has also played an important role in drawing multinational corporations and international investment to Israel.

**A major reason why there are so few big high-tech companies in Israel is that many are sold, often to foreign acquirers, before they become large.**

### Challenges

However, the high-tech sector is vulnerable despite its apparent strength. Among the greatest concerns is the global economic crisis, which has led to a reduction in the sources of credit and equity capital. The IPO market has stagnated, with not a single Israeli company listing on a foreign exchange since 2007, a notable fact given that Israel has more firms traded on Nasdaq than any other country outside North America. With revenues falling, businesses of all sizes and types are feeling the effects of the slowdown, leading to layoffs and company closures, and stymieing R&D projects and the establishment of new firms.

Israel is particularly exposed to the slump because of its reliance on exports and foreign capital, especially with the dollar weakening against the shekel. In 2008, sales abroad accounted for 40% of GDP, while from 1999–2008, the direct share of overseas financing in total VC investments was approximately 60%, although the overall foreign participation was much greater because Israeli VCs raise the vast majority of their funds from abroad. Much of this capital goes toward early-stage companies in the software, communications, networking, and semiconductors sectors and not enough to firms at a later stage of development or in other industries. These trends have contributed to a failure to create large companies, ensure diversity and spread the benefits of Israel's high-tech success to the wider economy and society. Moreover, Israel's tertiary education system is declining, threatening the basis of the country's technological edge and economic growth—its work force.

### Addressing the Challenges

As great as the challenges facing Israel are, there is much it can do to negate the effects of these difficulties. A policy paper for the Israel Democracy Institute (IDI) formulated by a team chaired by Arnon Bentur of the Technion-Israel Institute of Technology, and comprising government economic advisors, high-tech business leaders, leading academics, Chief Scientist Eli Opper and one of the writers of this column, Orna Berry, has called on the country's ruling coalition to reinvigorate the government's involvement in the high-tech sector with the aim of helping it overcome the economic crisis. Such involvement has become static in recent years, as pointed out in the report *The Future of Growth Promotion in Israel: A Return to Boosting Avant-Garde Industries and Scientific-Technological Innovation*,[b] the OCS's 2009 budget is only two-thirds of what it was in 2000. While the authors of the paper understand the spending constraints on the government, Israel should do its utmost to reverse this trend as a strategic priority, especially given that state investment in the high-tech sector has been shown to create wealth and generate returns for the economy.

To help stave off job losses, for example, and ensure a continuation of the innovation that is so vital to Israel's prosperity, the report recommends that the government increase its assistance to companies in reducing the costs of R&D projects. To lower the dependence on foreign capital, the authorities should encourage local institutional investment in the high-tech sector, which is minimal, through the loosening of prohibitive regulation and via the provision of tax incentives.

Another goal that the policy paper identifies is the need to encourage

b See http://www.idi.org.il/sites/english/events/TheAnnualEconomicForum/Documents/

the development of large companies, defined as employing over 450 workers and generating annual sales of more than $100 million. Just 1% of high-tech businesses in Israel belong to this category, and since 1995, only four companies have been created that have remained independent and now satisfy the second criterion. This is significant, because the advantages of large companies to the economy are manifold. One of the most important advantages is that because they often locate some of their manufacturing in different regions in Israel and employ workers withall types of backgrounds, they spread the benefits of the high-tech sector socially and geographically. In addition, job creation at big corporations is higher than at small firms, with OCS figures demonstrating that a company experiences a substantial growth in the number of its employees—from dozens to hundreds—after it crosses the threshold of $100 million in sales a year. Large firms are also more likely to be profitable, contribute to the tax base, less likely to close, and have greater resources for development.

A major reason why there are so few big high-tech companies in Israel is that many are sold, often to foreign acquirers, before they become large. This is partly a consequence of VC financing being such a prominent source of capital in Israel, and is the disadvantage of a model that has otherwise brought the country huge benefits. Furthermore, the size of most Israeli VC funds means that late-stage financing is mostly beyond their scope, because much larger sums of money are needed than at earlier junctures. There is also an almost total lack of post-VC funding, and in particular, private-equity mezzanine capital. The result is that growth companies find it difficult to raise the cash required to build the operational infrastructure—such as in finance, management, manufacturing, sales, and marketing—that enable them to expand. The government should fill in this gap by providing the necessary resources and reducing the risks involved, especially in regard to the building of large plants and other advanced projects, while simplifying the prohibitive bureaucracy associated with these processes.

This fostering of links between the private and tertiary sectors should also be widened as part of an overall strategy aimed at reversing a decline in the education system that threatens the quality of Israel's future work force and hence the long-term status of its high-tech industry. Major problems include a relative decline in the number of R&D university personnel at a time that this is rising in Europe and Asia, an aging faculty that is not being adequately replaced, high staff-student ratios, a "brain drain" to the U.S., and a fall in the proportion of undergraduates in science and engineering—subjects that are vital for entry into high-tech positions.

Immediate action that Israel should take to offset these trends, in addition to that noted above, includes expanding the higher-education budget in order to recruit new faculty members and attract researchers back to Israel from abroad; increasing grants for doctorates and post-doctorate studies; promoting a return to academic study among high-tech workers who have recently lost their jobs; and widening the scope of scientific research funds. As with other areas of government policy, a strengthening of the education system would help spread the benefits of the high-tech sector by facilitating the drawing in of more of those from Israel's lower socioeconomic echelons.

As has been demonstrated, Israel's high-tech sector constitutes a major part of its economy, so despite a surfeit of problems the government faces, it should give high priority to helping solve the difficulties the industry is experiencing. The government should not wait for this growth engine of the economy to splutter to a halt, especially with international competition for investment increasing as countries such as China, India, Finland, and South Korea execute strategies to promote their own high-tech sectors. C

**Orna Berry** (orna@gemini.co.il) is a venture partner at Gemini Israel Funds, the former chair of the Israel Venture Association, and the former chief scientist of the Israeli government.

**Yigal Grayeff** (yigal@gemini.co.il) is a communications manager for Gemini Israel Funds and was formerly a journalist, working for the *Jerusalem Post* in Israel and Dow Jones Newswires in the U.K.



# Calendar of Events

## December 2009

**December 16–18**
**International Conference on Frontiers of Information Technology,
Abbottabad, Pakistan,
Contact: Muhammad, Sarfraz,
Email: prof.m.sarfrax@gmail.com**

**December 16–19**
**International Conference on High Performance Computing,
Kochi, India,
Contact: Manish Parashar,
Phone: 732-445-5388,
Email: parashar@rutgers.edu**

**December 16–19**
**SIGGRAPH Asia 2009,
Yokohama, Japan,
Sponsored: SIGGRAPH,
Contact: Masahiko Inakage,
Phone: 81-467-32-7641,
Email: inakage@media-studio.co.jp**

## January 2010

**January 3–7**
**23rd International Conference on VLSI Design & 9th International Conference on Embedded Systems,
Bangalore, India,
Contact: Srivaths Ravi,
Email: srivaths.ravi@ti.com**

**January 9–14**
**ACM SIGPLAN Principles and Practice of Parallel Computing,
Bangalore, India,
Sponsored: SIGPLAN,
Contact: David Padua,
Phone: 217-333-4223,
Email: dapadua@gmail.com**

**January 18–22**
**The Twelfth Australasian Computing Education Conference,
Brisbane, Australia,
Contact: Tony G Clear,
Phone: 64-9-917-9999,
Email: tony.clear@aut.ac.nz**

**January 20–23**
**International Conference on Biomedical Engineering Systems and Technologies,
Valencia, Spain,
Contact: Joaquim B. Filipe,
Phone: 351-91-983-3996,
Email: jfilipe@insticc.org**

DOI:10.1145/1610252.1610265

Peter J. Denning and Peter A. Freeman

# The Profession of IT
# Computing's Paradigm

*Trying to categorize computing as engineering, science, or math is fruitless; we have our own paradigm.*

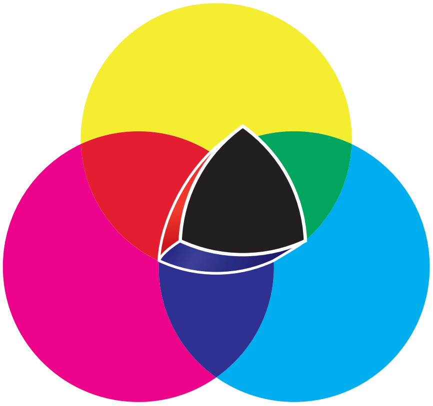
COMPUTING RIGHTFULLY COMES up in many discussions of university organization and curricula, high school courses, job qualifications, research funding, innovation, public policy, and the future of education. In repeated attempts to characterize our field in these discussions, our leaders continue to encounter sometimes contentious debate over whether computing is a field of engineering or science. Because it leaves others with a sense that we lack a clear focus, that debate negatively influences policies involving computing.

There seems to be agreement that computing *exemplifies* engineering and science, and that neither engineering nor science *characterizes* computing. What then does characterize computing? In this column, we will discuss computing's unique paradigm and offer it as a way to leave the debilitating debate behind.

The word "paradigm" for our purposes means a belief system and its associated practices, defining how a field sees the world and approaches the solutions of problems. This is the sense that Thomas Kuhn used in his famous book, *The Structure of Scientific Revolutions*. Paradigms can contain sub-paradigms: thus, engineering divides into electrical, mechanical, chemical, civil; science divides into physical, life, and social sciences, which further divide into separate fields of science.

## Roots of the Debate

Whether computing is engineering or science is a debate as old as the field itself. Some founders thought the new field a branch of science, others engineering. Because of the sheer challenge of building reliable computers, networks, and complex software, the engineering view dominated for four decades. In the mid-1980s, the science view began to assert itself again with the computational science movement, which claimed computation as a new sub-paradigm of science, and stimulated more experimental research in computing.



Along the way, there were three waves of attempts to provide a unified view. The first wave was by Alan Perlis,[9] Allen Newell,[8] and Herb Simon,[11] who argued that computing was unique among all sciences and engineering in its study of information processes. Simon went so far as to call computing a science of the artificial.

The second wave started in the late 1960s. It focused on programming, seen as the art of designing information processes. Edsger Dijkstra and Donald Knuth took strong stands favoring programming as the unifying theme. In recent times, this view has foundered because the field has expanded and the public understanding of programmer has become so narrow (a coder).

The third wave was the NSF-sponsored Computer Science and Engineering Research Study (COSERS), led by Bruce Arden in the mid-1970s. It defined computing as automation of information processes in engineering, science, and business. It produced a wonderful report that explained many exotic aspects of computing to the layperson.[1] However, it did not succeed in reconciling the engineering and science views of computing.

## Peaceful Coexistence

In the mid-1980s, the ACM Education Board was concerned about the lack of a common definition of the field. The Board charged a task force to investigate; its response was a report *Computing as a Discipline*.[4] The central argument of the report was that the computing field was a unique combination of the traditional paradigms of math, science, and engineering (see Table 1). Although all three had made substantial contributions to the field, no single one told the whole story. Programming—a practice that crossed all three paradigms—was essential but did not fully portray the depth and richness of the field.

The report in effect argued for the peaceful coexistence of the engineering, science, and math paradigms. It found a strong core of knowledge that supports all three paradigms. It called on everyone to accept the three and not try to make one of them more important than the others.

Around 1997, many of us began to think the popular label IT (information technology) would reconcile these three parts under a single umbrella unique to computing.[3,7] Time has proved us wrong. IT now connotes technological infrastructure and its financial and commercial applications, but not the core technical aspects of computing.

## A Computing Paradigm

There is something unsatisfying about thinking of computing as a "blend of three sub-paradigms." What new paradigm does the blend produce?

Recent thinking about this question has produced new insights that, taken together, reveal a computing paradigm. A hallmark of this thinking has been to shift attention from computing machines to information processes, including natural information processes such as DNA transcription.[2,6] The great principles framework interprets computing through the seven dimensions of computation, communication, coordination, recollection, automation, evaluation, and design (see http://greatprinciples.org). The relationships framework interprets computing as a dynamic field of many "implementation" and "influencing" interactions.[10] There is now a strong argument that computing is a fourth great domain of science alongside the physical, life, and social sciences.[5]

These newer frameworks all recognize that the computing field has expanded dramatically in the past decade. Computing is no longer just about algorithms, data structures, numerical methods, programming languages, operating systems, networks, databases, graphics, artificial intelligence, and software engineering, as it was prior to 1989. It now also includes exciting new subjects including Internet, Web science, mobile computing, cyberspace protection, user interface design, and information visualization. The resulting commercial applications have spawned new research challenges in social networking, endlessly evolving computation, music, video, digital photography, vision, massive multiplayer online games, user-generated content, and much more.

**There is an interesting distinction between computational expressions and the normal language of engineering, science, and mathematics.**

**Table 1. Sub-paradigms embedded in computing.**

| | Math | Science | Engineering |
|---|---|---|---|
| Initiation | Characterize objects of study (definition) | Observe a possible recurrence or pattern of phenomena (hypothesis) | Create statements about desired system actions and responses (requirements) |
| Conceptualization | Hypothesize possible relationships among objects (theorem) | Construct a model that explains the observation and enables predictions (model) | Create formal statements of system functions and interactions (specifications) |
| Realization | Deduce which relationships are true (proof) | Perform experiments and collect data (validate) | Design and implement prototypes (design) |
| Evaluation | Interpret results | Interpret results | Test the prototypes |
| Action | Act on results (apply) | Act on results (predict) | Act on results (build) |

**Table 2. The computing paradigm.**

| | Computing |
|---|---|
| Initiation | Determine if the system to be built (or observed) can be represented by information processes, either finite (terminating) or infinite (continuing interactive). |
| Conceptualization | Design (or discover) a computational model (for example, an algorithm or a set of computational agents) that generates the system's behaviors. |
| Realization | Implement designed processes in a medium capable of executing its instructions. Design simulations and models of discovered processes. Observe behaviors of information processes. |
| Evaluation | Test the implementation for logical correctness, consistency with hypotheses, performance constraints, and meeting original goals. Evolve the realization as needed. |
| Action | Put the results to action in the world. Monitor for continued evaluation. |

The newer frameworks also recognize the growing use of the scientific (experimental) method to understand computations. Heuristic algorithms, distributed data, fused data, digital forensics, distributed networks, social networks, and automated robotic systems, to name a few, are often too complex for mathematical analysis but yield to the scientific method. These scientific approaches reveal that *discovery* is as important as *construction* or *design*. Discovery and design are closely linked: the behavior of many large designed systems (such as the Web) is discovered by observation; we design simulations to imitate discovered information processes. Moreover, computing has developed search tools that are helping make scientific discoveries in many fields.

The newer frameworks also recognize natural information processes in many fields including sensing and cognition in living beings, thought processes, social interactions, economics, DNA transcription, immune systems, and quantum systems. Computing concepts enable new discoveries and understandings of these natural processes.

The central focus of the computing paradigm can be summarized as information processes—natural or constructed processes that transform information. They can be discrete or continuous.

Computing represents information processes as "expressions that do work." An expression is a description of the steps of a process in the form of an (often large) accumulation of instructions. Expressions can be artifacts, such as programs designed and created by people, or descriptions of natural occurrences, such as DNA and DNA transcription in biology. Expressions are not only representational, they are *generative*: they create actions when interpreted (executed) by appropriate machines.

Since expressions are not directly constrained by natural laws, we have evolved various methods that enable us to have confidence that the behaviors generated do useful work and do not create unwanted side effects. Some of these methods rely on formal mathematics to prove that the actions generated by an expression meet specifications. Many more rely on experiments to validate hypotheses about the behavior of actions and discover the limits of their reliable operation.

Table 2 summarizes the computing paradigm with this focus. While it contains echoes of engineering, science, and mathematics, it is distinctively different because of its central focus on information processes.[5] It allows engineering and science to be present together without having to choose.

There is an interesting distinction between computational expressions and the normal language of engineering, science, and mathematics. Engineers, scientists, and mathematicians endeavor to position themselves as outside observers of the objects or systems they build or study. Outside observers are purely representational. Thus, traditional blueprints, scientific models, and mathematical models are not executable. (However, when combined with computational systems, they give automatic fabricators, simulators of models, and mathematical software libraries.) Computational expressions are not constrained to be outside the systems they represent. The possibility of self-reference makes for very powerful computational schemes based on recursive designs and executions, and also for very powerful limitations on computing, such as the noncomputability of halting problems. Self-reference is common in natural information processes; the cell, for example, contains its own blueprint.

The interpretation "computational thinking"[12] embeds nicely into this paradigm. The paradigm describes not only a way of thinking, but a system of practice.

## Conclusion

The distinctions discussed here offer a distinctive and coherent higher-level description of what we do, permitting us to better understand and improve our work and better interact with people in other fields. The engineering-science debates present a confusing picture that adversely affects policies on innovation, science, and technology, the flow of funds into various fields for education and research, the public perception of computing, and the choices young people make about careers.

We are well aware that the computing paradigm statement needs to be discussed widely. We offer this as an opening statement in a very important and much needed discussion. **C**

**References**

1. Arden, B.W. *What Can Be Automated: Computer Science and Engineering Research Study (COSERS)*. MIT Press, 1983.
2. Denning, P. Computing is a natural science. *Commun. ACM 50*, 7 (July 2007), 15–18.
3. Denning, P. Who are we? *Commun. ACM 44*, 2 (Feb. 2001), 15–19.
4. Denning, P. et al. Computing as a discipline. *Commun. ACM 32*, 1 (Jan. 1989), 9–23.
5. Denning, P. and P.S. Rosenbloom. Computing: The fourth great domain of science. *Commun. ACM 52*, 9 (Sept. 2009), 27–29.
6. Freeman, P. Public talk "IT Trends: Impact, Expansion, Opportunity," 4th frame; www.cc.gatech.edu/staff/f/freeman/Thessaloniki
7. Freeman, P. and Aspray, W. *The Supply of Information Technology Workers in the United States.* Computing Research Association, 1999.
8. Newell, A., Perlis, A.J., and Simon, H.A. Computer science, letter in *Science 157*, 3795 (Sept. 1967), 1373–1374.
9. Perlis, A.J. The computer in the university. In *Computers and the World of the Future*, M. Greenberger, Ed. MIT Press, 1962, 180–219.
10. Rosenbloom, P.S. A new framework for computer science and engineering. *IEEE Computer* (Nov. 2004), 31–36.
11. Simon, H. *The Sciences of the Artificial.* MIT Press (1st ed. 1969, 3rd ed. 1996).
12. Wing, J. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

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

**Peter A. Freeman** (peter.freeman@mindspring.com) is Emeritus Founding Dean and Professor at Georgia Tech and Former Assistant Director of NSF for CISE.

DOI:10.1145/1610252.1610266

George V. Neville-Neil

Article development led by acmqueue
queue.acm.org

# Kode Vicious
# Broken Builds

*Frequent broken builds could be symptomatic of deeper problems within a development project.*

**Dear KV,**
Is there anything more aggravating to programmers than fellow team members checking in code that breaks a build? I find myself constantly tracking down minor mistakes in other people's code simply because they didn't check that their changes didn't break the build. The worst part is when someone has broken the build and they get indignant about my pointing it out. Are there any better ways to protect against these types of problems?

**Made to be Broken**

**Dear Made,**
I know you, and everyone else, are expecting me simply to rant about how you should cut off the tips of the pinkies of the offending parties as a lesson to them and a warning to others about carelessness. While that might be satisfying, it's illegal in most places and, I'm told, morally wrong.

A frequently broken build is a symptom of a disease, but it is not the disease itself. It indicates problems in any of the following three areas: management, infrastructure, or software architecture.

Management is the area that most quickly comes to mind when there is a team- or project-wide problem. The belief of most of the workers on a project—those tasked with writing and verifying code and systems—is that project-wide problems need to be solved by Mommy (aka the project

PHOTOGRAPH BY ROB BRUCKER

lead or the manager). Unfortunately, Mommy can remind people only so often to clean up their rooms, to tie their shoes, and not to check in broken code.

One of the best solutions to the problem of people not checking their code before they check it in is peer pressure. Anyone who checks in code without compiling it first ought to feel embarrassed by such a mistake, and if not, the other people around them should strongly encourage them to feel embarrassed. Shame, it turns out, is a strong motivator for avoiding antisocial behavior. Like many—or perhaps all—of KV's suggestions, shaming can be taken too far, but I suggest you try it and see how it works.

Depending on Mommy to tell off the misbehaving kids becomes tiresome both for you and the project management after a while. What you want to see is a good working culture develop, one in which people know that breaking the build is like taking a nap in the middle of the break room; funny once, but usually unacceptable.

Poor infrastructure can also lead to suffering with frequently broken builds. One thing that continues to amaze me is how computer hardware gets cheaper, and yet companies continue to coast along without a nightly, or more frequent, build system. For the price of a single desktop computer and a few days of scripting, most teams can have a system that periodically updates a test build of their code, builds it, and sends email to the team if the build fails. The amount of time saved by such a system is easily measurable: subtract 1 from the number of programmers on a team and multiply the resulting number by the number of hours it usually takes to figure out who broke the build, find them, shame them, and have them fix the build. Now multiply that number by the average hourly wage of each person on the team, and you have an approximate idea of how much time and money was wasted by not having peri-

**If all the components of a software system are too interdependent, then a change to one can result in an injury to all.**

odic builds. I won't get into periodic testing, which can save even more time and money, because if your build is always broken, you clearly have not achieved a sufficient level of sophistication to move on to nightly tests.

Even though the broken code will still get into the system, with a periodic build system the offending person will find out fairly quickly that he or she broke the build and hopefully will admit it in an email message ("I broke the build, hang on a second)" and then repair the error. While this is still suboptimal, it is far better than what you had before.

Sometimes it is the build system itself that is the source of the problem. Many modern build systems depend heavily on caching derived objects, as well as the parallelization of the build process. While a parallel build process can provide you results more quickly, it can often lead to build failures that are false positives. Trying to build an object that requires another object to be created first, such as an automatically created `include` file, always leads to trouble. Maintaining the list of dependencies by hand is an error-prone, but often necessary, process. If you are using a build system that depends on caching and uses parallel builds, then your problems may lie here.

Now we come to the final area that is the cause of build problems. The way in which a piece of software is put together, frequently referred to as its architecture, often impacts not only how the software performs when it runs, but also how it is built. I hesitate to use the word architecture since overuse of the term has led to the unfortunate proliferation of the job title software architect, which is far too often a misnomer.

If all the components of a software system are too interdependent, then a change to one can result in an injury to all. A lack of sufficient modularization is often a problem when software ships, but it is definitely a problem when the software is being compiled. When a change to an `include` file in one area leads to the build breaking in another area, then your software is probably too heavily interlinked, and the team should look at breaking the pieces apart. Often such links come from careless reuse of some part of the system. Careless reuse is when you look at a large abstraction and think, "Oh, I really want this version of method X," where X is a small part of the overall abstraction, and then you wind up making your code depend not just on the small part you want, but on all of the parts that X is associated with. If you get to the point where you know it's neither carelessness nor poor infrastructure that is leading to frequent build failures, then it's time to look at the software architecture.

Now you know the three most basic ways to alleviate frequent build breakage: shaming your teammates, adding some basic infrastructure, and finally improving the software architecture. That ought to keep you out of jail, for now.

**KV**

**Related articles on queue.acm.org**

**Kode Vicious: Facing the Strain**
http://queue.acm.org/detail.cfm?id=1160442

**Making Sense of Revision-Control Systems**
*Bryan O'Sullivan*
http://queue.acm.org/detail.cfm?id=1595636

**A Conversation with Steve Bourne, Eric Allman, and Bryan Cantrill**
http://queue.acm.org/detail.cfm?id=1454460

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

DOI:10.1145/1610252.1610267

Savas Parastatidis, Evelyne Viegas, and Tony Hey

# Viewpoint
# A "Smart" Cyberinfrastructure for Research

*A view of semantic computing and its role in research.*

THE WEB HAS emerged as the largest distributed information repository on the planet. Human knowledge is captured on the Web in various digital forms: Web pages, news articles, blog posts, digitized books, scanned paintings, videos, podcasts, lyrics, speech transcripts, and so forth. Over the years, services have emerged to aggregate, index, and enable the rapid searching of all this digital data but the full meaning of that data may only be interpretable by humans. In the common case, machines are incapable of understanding or reasoning about the vast amounts of data available on the Web. They are not able to interpret or infer new information from the data and this has been a topic of active research interest for decades within the artificial intelligence community. While the dream of artificial intelligence—machines capable of human-level reasoning and understanding—may still not be within our grasp, we believe semantic technologies hold the promise that machines will be able to meaningfully process, combine, and infer information from the world's data in the not-too-distant future (see Figure 1).

The Web ecosystem of simple formats and protocols is an example of how we can effectively manage, share, access, and represent large amounts of data. Companies like Microsoft and Google are building large-scale services (such as search and cloud services) leveraging the existing hardware and software infrastructures. Schema languages, XML, Entity Data Models, Microformats, RSS, Atom, RDF (see http://www.w3.org/RDF/), OWL (see http://www.w3.org/2007/OWL/), and other technologies are being used to capture the information in data while machine learning, entity extraction, neural networks, clustering, and latent semantics are approaches to extracting information from that data and help reason about it. The field is an active area of research and experimentation and is still rapidly evolving (see the sidebar "Semantic Computing" vs. "Semantic Web").

## Data Mesh

At the center of our discussion is the concept of a "data mesh," a term we use to refer to the various information and knowledge representation techniques/technologies that have been developed over the years (see Figure 2). In its simplest form, a data mesh looks like a directed graph in which the nodes represent data/information captured in well-known formats and the edges capture a relationship, characterized by a predicate and perhaps other information, between the linked data. For example, "Jane listens to Santana every day" is a relationship, in which "Jane" (the subject) and "Santana" (the object) are the nodes, "listens to" is the edge (the predicate), and "every day" is an attribute of the edge. Other tuples could add further information to the data mesh (for example, "Santana is an artist," "Santana plays the guitar," "Santana makes music," "Jane met Santana in 1995" and so forth). Semantic Web's RDF is one, but not the only, technology that can be used to represent such graphs or knowledge bases. Indeed, Cyc,[4] Semantic Networks, WordNet,[7] MultiNet[6] are examples of other such technologies/approaches. Scaling to the same level as the Web still remains a challenge for these approaches.

We believe there is an opportunity to involve users, who are now equally producers as they are consumers of information on the Web, and not just the very few experts in producing

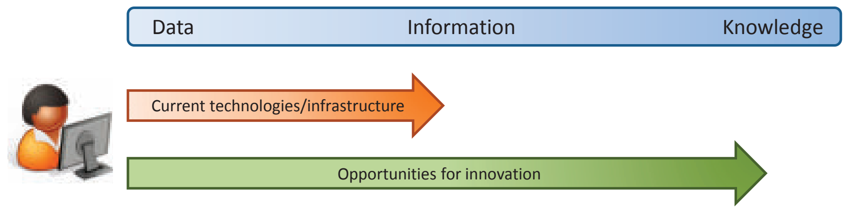



**Figure 1. Data, information, knowledge: While we are good at data management at scale (for example, Google, Amazon) we are still far away from supporting information representation and reasoning. Knowledge management at scale is a great opportunity for innovation.**

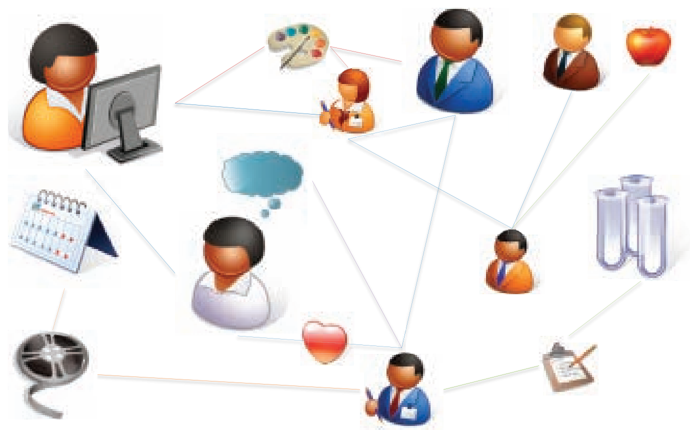


**Figure 2. All data/information can be linked in a mesh through relationships. Common, machine-processable formats are used to represent every aspect of a data mesh.**

structured data at Web scale. Recent success stories in the application of knowledge representation to specific domains, as in the myGrid (see http://www.mygrid.org.uk/) work in BioInformatics research,[10] demonstrate the potential benefits of semantic computing technologies. Here, we use the term "data mesh" to encompass the various concepts and approaches that could be used or combined to support a semantics-rich ecosystem of research tools and services. It is not our intention to suggest there would be one single data mesh that would represent all human knowledge.

We expect a great number of vocabularies to emerge, many of which will overlap, for representing every aspect of a data mesh (such as geo-location, mood, reviews, personal information, domain-specific concepts and terms). Ontologies will support an evolving ecosystem of facts, vocabularies, and relationships in specific domains. We are already witnessing a plethora of emerging efforts to standardize on such vocabularies, such as microformats (http://www.microformats.org/), data portability (http://www.dataportability.org/), gene ontology (http://www.geneontology.org/), and others.

Programs will consume, combine, and correlate everything in the universe of structured information and help users reason over it. They will allow them to ask questions against this (global) collection of facts—information access policies permitting—such as "Which is the most popular book among my friends today?," "Who is the expert on aspect A of my business workflow inside my organization?," "Have Evelyne and Savas been at the same conference, at the same time in any point in time?," "What's the degree of separation in terms of citations between my paper and the seminal work by Jim Gray?" and so on.

While data mesh instances can be built in isolation (as in many of today's social networks), we believe the potential value of aggregating all of them and combining them in one huge network of facts is tremendous. This idea is similar to Tim Berners-Lee's more recent rhetoric around the 'Giant Global Graph of Facts' (see http://dig.csail.mit.edu/breadcrumbs/node/215). Please note that we are not suggesting there would be a single repository of facts or that there would even be universal agreement on what is represented. We do expect, however, to see machine-based technologies that would be able to reason, many times using probabilistic-based techniques, over the diverse set of facts.

**We are already witnessing the emergence of data mesh instances on the Web, especially as they relate to social networks.**

We are already witnessing the emergence of data mesh instances on the Web, especially as they relate to social networks. The Zune Social (http://social.zune.net/) is an example of how a social network can be combined with information about music preferences, recommendations, and an online marketplace. Facebook (http://www.facebook.com/) is another example of how connections between identities can help in aggregating user-oriented preferences and then inferring behavior and preference statistics. Finally, Powerset (http://www.powerset.com/) is an example of a search service that leverages existing structured information, for example, Freebase (http://www.freebase.com/) or generates it from unstructured sources (such as by applying natural language processing technologies on Wikipedia content) to improve the quality of the query results.

We believe that over time, a huge ecosystem of services and tools will emerge around data mesh instances. Such tools and services will allow us to move beyond current practice of information management by incorporating more automation. Recommendation engines will be the norm and our interactions with computers will always be context-aware (for example, "since the topic of the paper being written is about botany, a query about 'bush' is unlikely to be about a person's name" or "the search about papers on orchid will take into consideration the opinion of people in the user's professional social network"). While today we can search for information over the global graph of linked Web pages consisting of predominately unstructured data, in the future we will be able to search over all types of semantically enriched information, which will in turn enable a wide range of new applications to emerge such as recommendation services, information management automation, information inferencing, and so forth.

### Tools and Services to Support Research

We believe the research community will play a central role in supporting and further evolving the semantic computing vision. We should not only become early adopters of semantic computing technologies and infrastructure in our research projects but we should also actively develop and evolve them. In Microsoft Research we are taking some first steps toward this vision, as we are investing in projects that can demonstrate the benefits of semantic computing technologies in research. We are therefore attempting to build an ecosystem of research tools and services as demonstrations of these ideas and concepts.

We focus here on the role of the researcher as an "extreme information worker" meaning a technology user with expectations and requirements at a scale not yet required by the business community. We believe information representation, management, and processing tools in combination with automation technologies will greatly help them in their research. We are therefore taking small steps toward developing semantics-aware tools and services. Here, we describe some of the work we are doing in supporting the scholarly communications life cycle through semantic computing technologies.

*Semantic Annotations and Metadata in Word.* The authoring stage is perhaps the best time to capture an author's intentions and to record the meaning of the words as they are being written. Natural language may not always be adequate to convey the meaning of a word or an expression, especially in the scientific world. In many disciplines domain-specific ontologies are therefore being created by experts to address this issue but they have not so far been incorporated with productivity tools like Microsoft Office.

**Natural language may not always be adequate to convey the meaning of a word or an expression, especially in the scientific world.**

In collaboration with Phil Bourne and Lynn Fink at the University of California, San Diego, we worked toward a plug-in for Word 2007 (part of the BioLit project; http://biolit.ucsd.edu/) that allows authors to annotate words or sentences with terms from an ontology (for example, Gene Ontology; http://www.geneontology.org/). The annotations are stored as part of the Office Open XML (OOXML) representation of the document (OOXML has been accepted as an ISO standard. More information can be found at http://openxmldeveloper.org/). Tools and services can now extract the annotations by just opening the OOXML package without human intervention and there is not even a need for Word to be installed. As a result, the documents will be able to be better categorized, indexed, and searched with the author's intent always closely associated with the text.

The ability to easily annotate terms from within Word is a first step in producing documents that semantically relate to the body of knowledge in a domain. In this way, information can easily become part of a data mesh as it is being generated (see Figure 3). The source code for the plugin is now available as open source (see http://ucsdbiolit.codeplex.com/) for the community to further extend or just use as the basis for a new generation of semantics-oriented authoring tools.

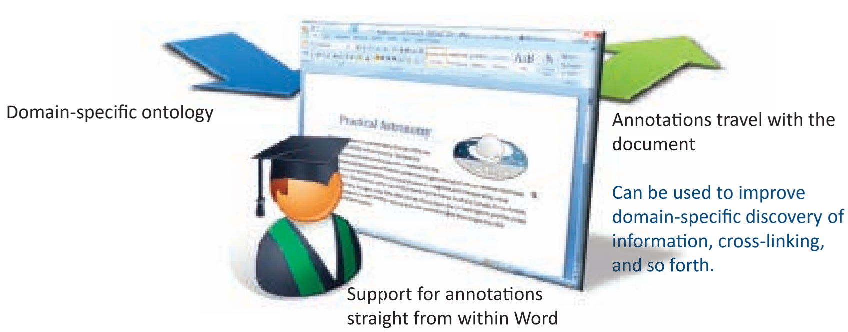
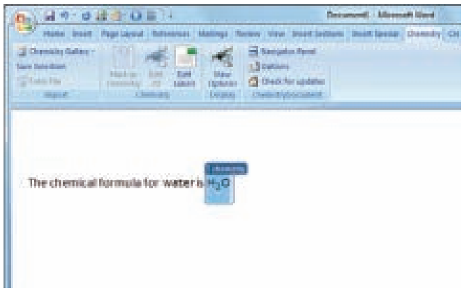



**Figure 3. Semantic annotations in Word.**



```
<cml ...>
  <molecule ...>
    <atomArray>
      <atom elementType="C" ... />
      <atom elementType="H" ... />
      ...
    </atomArray>
    <bondArray>
      <bond ... />
      <bond ... />
      ...
    </bondArray>
  </molecule>
</cml>
```

**Figure 4. A simple "Chemistry Zone" in a Word document and the CML representation (in pseudo-XML) stored inside the OOXML document.**

*Chemistry in Word.* We are investigating, in collaboration with Peter Murray-Rust, Jim Downing, and Joe Townsend from the University of Cambridge, the introduction of chemistry drawing functionality into Word documents (see http://research.microsoft.com/en-us/projects/chem4word/). Rather than just having images of chemical structures, we would like to preserve the chemistry-related semantics in a machine-processable manner. For that reason, we are using the Chemistry Markup Language (CML) in our investigations; instances of CML would be embedded inside OOXML documents. We believe an ecosystem of chemistry-related tools and services can then emerge to enable the automatic processing of documents, making the authoring process an easy but increasingly valuable part of the research life cycle.

As an example, consider the water molecule ($H_2O$). In a Word document, it appears as a series of characters, one of which is a subscript. Through this project, it will be possible to also store the structured representation of water so that programs can discover it. Figure 4 shows how some part of a document can be identified as chemistry. The tool will automatically save the CML representation of the identified region (1D and 2D representations and authoring functionality will also be supported; see Figure 5). The use of a semantically rich data format to represent domain-specific information is another step in producing structured data that automatically becomes part of the data mesh.

*Zentity—A Repository Platform.* The need for quality, well-engineered, and documented software infrastructures to support institutional repositories, archives, and digital libraries is increasing, especially in the context of the global initiative toward Open Access.[2,5] We have developed a platform, called Zentity, to support repository systems based on product-quality technologies like SQL Server, .NET 3.5, and the Entity Framework (see http://research.microsoft.com/en-us/projects/zentity/).

The Zentity platform supports a graph-based representation of the data in a repository. It provides an easy-to-use application programming interface that abstracts the use of the underlying relational system to manage digital resources and the relationships between them and creates a data mesh. Initially, the platform will be targeted toward the "research output repository" domain, offering a data model capable of capturing the research-related resources (for example, papers, reports, theses, presentations, and data sets) of an organization. However, Zentity has been designed to support the data models of arbitrary domains (for example, museums, art collections, research data, and so forth).

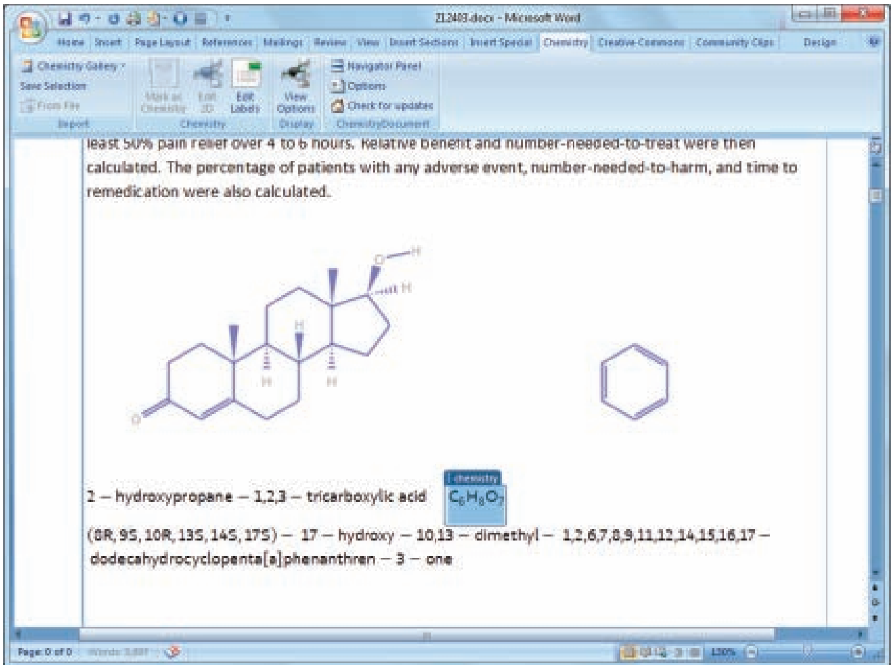



**Figure 5. A more complicated chemical structure from an early prototype of our tool.**

Interoperability is a major focus of the project and we are implementing support for popular Semantic Web technologies, like RDF and RDFS. We are also building a number of tools and services to operate against the data mesh created and we hope that more will be developed by the community as we make the platform freely available.

In addition to supporting the repository community through product-quality technologies, our work on Zentity attempts to demonstrate some of the principles of data meshes. The data model employed for the implementation promotes the graph representation principles discussed earlier and illustrated in Figure 2.

*Social Networking and Data Meshes.* As a final example, we examine the relationship between social networks and other structured information. We consider the former a special case of a data mesh (in which the nodes are people and the edges between them represent human relationships, such as "friend" or "colleague"). A social network can provide context for the interactions between people (for example, "the botany domain-specific social network"); it can be used to infer information about a community (for example, "the botany community has been actively looking at orchids over the last week"); it can be used to provide recommendations (for example, "the most read article about orchids can be found at location X"); it can be used to change or supplement the way peer reviewing is done and merit is given (for example, "the article posted online on orchids last week has received great reviews from the experts of the botany community").

One can therefore imagine an ecosystem of tools and services that takes advantage of the relationships between researchers in the context of a particular research discipline, collaboration or research project and their activities, documents, and opin-

## "Semantic Computing" vs. "Semantic Web"

**We make a distinction between the general approach of computing based on semantic technologies (machine learning, neural networks, ontologies, inference, and so forth) and the "Semantic Web" as described in Berners-Lee et al.[1] and Shadbolt et al.[9] and which is the term used to refer to a specific ecosystem of technologies, like RDF and OWL. The Semantic Web has gained a lot of attention lately, bringing more awareness of the importance of semantics. However, we consider the Semantic Web technologies to be just some of the many tools at our disposal when building semantics-based solutions.**

ions. Typical research-related processes could be augmented or even completely supplanted. For example, researchers could automatically get recommendations of papers and contacts based on what they are currently doing; experts might be automatically identified in a domain based on discussions around their papers and blog entries; peer reviewing could evolve to take into consideration the new social media and Web-based interactions; and even 'impact factors' for institutions might incorporate electronic analysis of all types of information and not just citations to publications and research grants.

Our support of the myExperiment (http://www.myexperiment.org/) project is a demonstration of our belief that scientific collaboration and information sharing can be supported through social networking. The myExperiment project brings together social networks and workflows in a single information graph—a data mesh—that can be browsed, analyzed, and searched.

## Conclusion

As researchers and scientific instruments can now produce and publish large amounts of data and information more easily than at any other point in history, there is an increasing requirement for automation tools to help manage and navigate the deluge of research data. For example, projects like Pan-STARRS (http://pan-starrs.ifa.hawaii.edu/) and the HLC (http://lhc.web.cern.ch/) will generate many petabytes of data. The emergence of folksonomies on the Web is one example of how user-driven categorization can help with information discovery. The need to deal with meaningful and relevant information within the context of one's actions is growing. There is an immense opportunity for the research community to bring its expertise and experience together in accelerating the development of semantic computing technologies. We need to invest significant resources to making the semantic computing vision a reality by:

- investing in semantics-aware infrastructure;
- increasing awareness of the potential of semantics-based computing; and
- training more researchers on semantics-based computing and related technologies.

The discussion on data meshes shows the potential value of aggregating information in a (semi-)structured, machine-interpretable manner. We believe an ecosystem of desktop tools, cloud services, and data formats will emerge to support "information and knowledge management," namely, the (automatic) acquisition, representation, aggregation, indexing, discovery, consumption, correlation, management, and inference of information. Doing so at scale would significantly improve the way we discover and share information and how we collaborate.

We have described a representative set of investments we are making to ease the transition of researchers toward a world where information is produced and consumed in a structured and semantics-rich manner (more information about the work and research tools offered by Microsoft Research for scientists can be found at http://research.microsoft.com/en-us/collaboration/about/). However, this will not happen instantly. There is a lot of unstructured data out there already. Data-mining technologies are necessary to automatically extract as much semantically rich information as possible. For example, Microsoft's Live Labs has worked on machine learning-based technologies to extract entities from the unstructured Web (see http://livelabs.com/projects/entity-extraction/). The research world needs similar technologies to be deployed at scale that can aggregate, index, and mine research-related information.

We believe such an ecosystem of semantics-aware tools and services will ultimately become the norm in our day-to-day interactions with computers, constituting a global "smart cyberinfrastructure." However, if the big companies are to invest in implementing these ideas and technologies in their offerings (products and services), the research community must test and demonstrate their potential as part of the community's attempt to build a smart cyberinfrastructure for research. Ultimately, this vision of a data mesh and smart cyberinfrastructure will go some way toward realizing the visions of the early pioneers like Vannevar Bush[3] and J.C.R. Licklider.[8]

**We need to invest significant resources to making the semantic computing vision a reality.**

**There is an increasing requirement for automation tools to help manage and navigate the deluge of research data.**


**References**

1. Berners-Lee, T., Hendler, J.A., and Lasilla, O. *The Semantic Web.* Scientific American (May 2001).
2. Borgman, C.L. *Scholarship in the Digital Age: Information, Infrastructure, and the Internet.* MIT Press, 2007.
3. Bush, V. As we may think. The Atlantic Monthly (1945); www.theatlantic.com/doc/194507/bush
4. Cycorp. Cyc Knowledge Base; www.cyc.com/
5. Dirks, L. and Hey, T., Eds. *CTWatch Quarterly: The Coming Revolution in Scholarly Communications & Cyberinfrastructure 3*, 3 (Aug. 2007).
6. Fellbaum, C., Ed. *WordNet.* The MIT Press, 1998.
7. Helbig, H. *Knowledge Representation and the Semantics of Natural Language.* Springer, Berlin, 2006.
8. Licklider, J.C.R. *Libraries of the Future.* MIT Press, 1965.
9. Shadbolt, N., Berners-Lee, T., and Hall, W. The Semantic Web revisited. *IEEE Intelligent Systems 21*, 3 (Mar. 2006), 96–101.
10. Wroe, C. et al. A suite of DAML+ OIL ontologies to describe bioinformatics Web services and data. *International Journal of Cooperative Information Systems 12* (2003), 197–224.



**Savas Parastatidis** (Savas.Parastatidis@microsoft.com) is a principal developer with Microsoft Technical Computing and a visiting fellow at the University of Newcastle upon Tyne, U.K.

**Evelyne Viegas** (evelynev@microsoft.com) is a senior program manager responsible for the Online Technologies and Web Cultures initiative at Microsoft Research in Redmond, WA.

**Tony Hey** (Tony.Hey@microsoft.com) is a corporate vice president of the External Research Division of Microsoft Research in Redmond, WA.

# practice

DOI:10.1145/1610252.1610268

Article development led by acmqueue
queue.acm.org

**Do RFID passports make us vulnerable to identity theft?**

BY ALAN RAMOS, WEINA SCOTT, WILLIAM SCOTT, DOUG LLOYD, KATHERINE O'LEARY, AND JIM WALDO

# A Threat Analysis of RFID Passports

IT'S A BEAUTIFUL day when your plane touches down at the airport. After a long vacation, you feel rejuvenated, refreshed, and relaxed. When you get home, everything is how you left it—the tables, the chairs, even the now-moldy sandwich you forgot on the counter. Everything, that is, but a pile of envelopes on the floor that jammed the door as you tried to swing it open.

You notice a blinking light on your answering machine and realize you've missed dozens of messages. As you click on the machine and pick up the envelopes, you find that most of the messages and letters are from debt collectors. Most of the envelopes are stamped "urgent," and as you sift through the pile you can hear the messages from angry creditors demanding that you call them immediately. Reading the bank statements, you suddenly realize that someone has been charging large amounts of money to an account in your name from a credit card company you've never heard of. You've lost thousands of dollars, and suddenly you aren't feeling quite so relaxed anymore.

How could someone have been stealing money from you like this while you were away on vacation? The thievery actually began months before you even left home. Several months ago, as you were casually walking through the airport en route to a business meeting in Europe, someone was lingering close behind. As you approached a security agent to have your passport


PHOTOGRAPH BY BRENT HARDINGE

checked, this individual used a small antenna connected to a computer in his backpack to *eavesdrop* on the radio communication between the security agent's reader, which has the capacity to decrypt the highly sensitive and secured data on the passport, and the RFID-enabled passport itself.

If the attacker had tried to *skim* the information off your passport by imitating a legitimate reader, the chip would never have provided the personal data within, as the correct access key would not have been given. Since the attacker was merely intercepting the communication with an antenna, however, he was able to collect all of the data, albeit in an encoded form. Private information, including not only basic information about your identity but even a digitized photograph, had been stolen from you at a moment when you thought your passport was safely in the hands of a government official. You moved on without any clue as to how deeply your privacy had been violated in an attack that you had no idea was occurring.

At that point, all the perpetrator needed to do was use the data to create a new passport, use that passport to get a U.S. Social Security number (http://www.ssa.gov/pubs/10002.html), and then create credit card accounts in your name, with your identity, and run amok with your finances.

An RFID-passport attack of this nature is more plausible than other methods, such as skimming the RFID information. Although simple to do, skimming will not yield the information needed to enable identity theft because of preventive measures integrated into the system. The first of these measures is encryption. According to the U.S. Department of State: "When a reader attempts to scan the passport, it engages in a challenge-response protocol that proves knowledge of the pair of keys and derives a session key. If authentication is successful, the passport releases its data contents; otherwise, the reader is deemed unauthorized and the passport refuses read access."[6]

Additionally, newer passport covers are being lined with materials that block RFID signals from being transmitted when the passport is closed, exposing the document to attack only when it is opened and displayed for a security agent. Relatively inexpensive signal-blocking sleeves (http://www.rfid-shield.com/products.php) are also available for RFID passports.

### What Information is Compromised?

Six pieces of information can be stolen from the RFID chip on a U.S. passport: your name, nationality, gender, date of birth, place of birth, and a digitized photograph.[1] Numerous problems of identity theft could arise from someone taking that information, but this article focuses on the financial risk.

Banks in the U.S. require that applicants for credit cards submit their Social Security numbers to be used for background credit checks. Although the passport RFID tag does not carry your Social Security number, a perpetrator can use the information it does contain to obtain your number.

The Social Security Administration's Web site (http://www.ssa.gov/pubs/10002.html) requires one of three proofs of identity for a U.S. citizen to be issued a new Social Security card: a driver's license, state-issued non-driver identity card, or passport. With the data stolen from your passport's RFID chip, someone could create a copy of the passport, then use this counterfeit one to access a real copy of your Social Security card. With this card, the perpetrator is free to apply for a real copy of your credit card, not to mention opening new accounts in your name. This puts you at a serious financial risk, all because someone was able to eavesdrop on your passport's RFID communication.

**Six pieces of information can be stolen from the RFID chip on a U.S. passport: your name, nationality, gender, date of birth, place of birth, and a digitized photograph.**

### Technology Requirements

To eavesdrop on your passport information, a perpetrator needs hardware to capture the signal as it is being scanned by a legitimate RFID reader, such as those used by government officials at airports. He or she would then need the time and technical capacity to decrypt the signal into a usable form. Finally, to reap any real benefits from the stolen information, the attacker must have all the materials necessary to reproduce a passport. We can view this as a series of hurdles that the perpetrator must overcome, starting with data capture, moving onto data recovery, and finally data reproduction.

Let us first focus on capturing the information from your passport, since it is at that point in the event chain that the vulnerabilities of the RFID technology are exploited. For successful data retrieval the perpetrator's antenna must catch two different interactions: the forward channel, which is the signal being sent from the RFID reader to the RFID token; and the backward channel, which is the data being sent back from the RFID token to the RFID reader. Lab demonstrations[3] have shown that a successful eavesdrop (a capture of both channels) on an RFID tag can occur at a distance of one meter with the use of an H-field antenna, a radio frequency receiver, an oscilloscope to monitor the signals, and a computer to store, analyze, and manipulate the data.

In the lab this was done as a proof of concept, but in the real world a perpetrator could use smaller, more discrete hardware. In our airport scenario, the perpetrator would need only an antenna and an amplifier to boost the signal capture, a radio-frequency mixer and filter, and a computer to store the data. The amplifier itself would not even need to be that powerful, since it would need to boost the signal over only a short distance of three to five meters. The antenna, mixer, and filter can be homemade with cheap materials or purchased as a set online. Some Web sites (for example, http://www.openpcd.org/openpicc.0.html) contain schematics, lists of materials, and steps on how to build your own RFID reader the size of a matchbox. These RFID "sniffers" can then be plugged into a laptop via a USB port.

Once the perpetrator has successfully eavesdropped on the communication between the RFID token and the RFID reader, the next step is data recovery. This requires two separate steps. The first is recovering the actual signal between the RFID chip in the passport and the RFID reader. This is a signal-processing problem, essentially separating the actual signal from the noise of the background. Proof-of-concept experiments[3] have shown that data recovery is a brute-force problem that can be solved with current hardware. A perpetrator would need only to record the data passed between the RFID and receiver on location, and then could perform the time-consuming signal-processing operations at home. A large part of data recovery is extracting the data from the electrical noise of the environment, which is simplified by taking a noise profile of the environment. The same Web sites that provide schematics for readers also provide code for decoding the data, although the effectiveness of their programs on new passports has yet to be tested.

Once the signal has been recovered, it must be interpreted as data. The difficulty of this step depends entirely on whether and how well the data is encrypted. The encryption key is generated from information on the passport—specifically, the name, date of birth, and passport number. There are reports that this key can be easily cracked (for example, http://www.mobilemag.com/2006/02/03/global-rfid-passport-encryption-standard-cracked-in-2-hours/) because the algorithm used to produce the key is predictable. An analysis published by the International Association of Cryptologic Research indicates that the entropy of the resulting key is on the order of 52 bits, which, while something of a challenge, is not impossible to crack.[4] We assume here that decryption is practical; if it is not, then the possibility of these attacks is minimized.

After recovering the data, the perpetrator would have everything necessary to make a new passport with the captured information. The steps required for this are beyond the scope of this article, but since counterfeiting of passports has been demonstrated and documented, it is enough to say that this is feasible.

## Costs to the Perpetrator

What we have shown so far is that with the right equipment and skill, a perpetrator can intercept the signal between a passport and RFID reader, then forge the passport to use for identity theft. The more important question, however, is whether the cost of doing this can be justified by the return.

This question is predicated on the assumption that the encryption of the information held in the passport's RFID tag can be broken. While there is some evidence this has been true in the past, stronger encryption could increase the cost of the attack considerably, to the point of making it either economically unattractive or technically impossible.

In our airport scenario, a perpetrator would have to cover several costs before reaching the ultimate goal of financial gain. To begin with, there are the hardware costs. The combined cost of the antenna, amplifier, radio mixer, filter, USB connection, and laptop would be on the order of $1,000. These are all fixed costs, and the perpetrator would presumably amortize these by using the hardware to execute numerous attacks over a period of time.

There is also cost associated with access to the passport reader. It is reasonable to assume that the perpetrator would have to purchase an airline ticket to enter the area where passports are scanned.

The cost of being caught must be factored in. Compared with other technologically intensive (for example, online) fraudulent attacks, theft of passport RFID data might involve greater risk because of the physical proximity required to eavesdrop on the RFID communication. The risk-adjusted cost of being caught is quite significant when you consider the prevalence of security officers within airports and the severity of the crime.

Presuming that the attacker manages to escape with the raw data from an eavesdropping operation, it still


PHOTOGRAPH BY ALEX GRIMM/GETTY IMAGES

**EasyPass, a new automated border control system at Frankfurt International Airport, scans passenger biometric data and compares it to data from the person's e-passport.**

has to be interpreted at home. The software costs are negligible (open source code for this specific function is available on the Internet) as are the costs of the processing time. In one example, it took less than an hour to recover the passport signal, and this process can be automated.[3] Although we have not verified this (since verification would require snooping a passport in a noisy environment such as an airport), the approach presented seemed plausible.

Jeroen van Beek of the University of Amsterdam managed to forge a passport RFID chip for $120.[5] This cost is not always necessary because a U.S. passport remains valid even if it is not fitted with an RFID chip or if the chip has failed. (Since all passports issued after 2007 have an embedded RFID chip and are valid for a maximum of 10 years, the ability to use a passport without such a chip will end after 2017.) Rather, the most significant cost is in obtaining or producing a realistic-looking passport in which to print the information. The cost of a blank passport book is difficult to determine, but there are some indications that it is not an insubstantial part of the cost of this form of identity theft. In 2008, for example, 3,000 blank U.K. passports were stolen, and officials valued each one at approximately $3,000.

Estimating the revenues that could be generated also requires some inference. In the U.S., the mean fraud amount per victim for identity theft-related crimes in 2008 was $4,849.[2] The potential revenue from the passport identity theft example, however, could conceivably be higher because of the relative ease with which a passport can be used to open new accounts and prove identity, in comparison with the most common current forms of fraud using stolen credit cards, checks, or mail. Nevertheless, comparing this figure to the $3,000 cost of a blank passport (which is just one of the many costs of creating a fake passport) reveals that the operation may not be as profitable as one might have thought.

## Countermeasures

A number of countermeasures have been suggested to protect against RFID privacy risks (not specific to the passport example), including permanent tag deactivation ("killing"), temporary tag deactivation (such as using Faraday cages or sleep/wake modes), and access-control mechanisms (hash locks, pseudonyms, blocker tags). You could "kill" the RFID tag (hitting the chip with a hammer does the trick), since, according to the State Department's Web site, if the chip fails, the passport remains valid; however, most "killing" methods leave evidence of intentional damage. The other solutions would not prevent the interception of communications between tag and authorized reader, particularly at an airport.

More effective countermeasures require changes to current government policy. The government can take steps to improve the security and privacy of passports. The basic access-control system of a U.S passport encrypts communication between it and the RFID reader with a key generated from information written on the passport; the key containing the holder's information is susceptible to brute-force attacks, however, since it has low entropy.[4] One countermeasure would be to add a 128-bit secret, printed on the passport and unique to each passport, to the key derivation algorithm.

The interception of communications between RFID tag and reader is possible because no material capable of blocking RF signals surrounds the passport-control area. Thus, another countermeasure would be to install an enclosure to block RFID transmission outside of the immediate area. Increased security around the passport-control area could also minimize the possibility of intrusion on the communication between tag and reader.

## The Final Analysis

Having looked at the potential attack, the costs of that attack, and the returns, we can now ask how concerned we should be about such an exploit. Should you really be worried as you walk through the airport that someone behind you might be stripping you of your passport information in a grand scheme to rob you?

The technical hurdles are surmountable, at least in proof-of-concept demonstrations. It is possible that such an attack could occur, but this possibility must be balanced against the complexity of the attack, the difficulty of obtaining the required high-priced blank passport, and the limited return the attack is likely to produce.

It seems much more likely that most perpetrators would resort to old-fashioned means of stealing your passport information, by stealing your physical passport itself. We recommend that it is more important to be careful about keeping your physical passport safely in hand than to be wary of perpetrators lurking behind you in line at the airport attempting to exploit the RFID tag in your passport. C

**Related articles on queue.acm.org**

**Communications Surveillance: Privacy and Security at Risk**
*Whitfield Diffie and Susan Landau*
http://queue.acm.org/detail.cfm?id=1613130

**Four Billion Little Brothers? Privacy, mobile phones, and ubiquitous data collection**
*Katie Shilton*
http://queue.acm.org/detail.cfm?id=1597790

**The Magic of RFID**
*Roy Want*
http://queue.acm.org/detail.cfm?id=1035619

**References**


1. Broache, A. RFID passports arrive for Americans. CNET News (Aug.14, 2006); http://news.cnet.com/RFID-passports-arrive-for-Americans/2100-1028_3-6105534.html.
2. Claburn, T. Identity thieves face pay cut. *Information Week* (Feb. 11, 2009); http://www.informationweek.com/news/security/privacy/showArticle.jhtml?articleID=213403976.
3. Hancke, G. Eavesdropping attacks on high-frequency RFID tokens. RFIDBlog (July 11, 2008); http://www.rfidblog.org.uk/Hancke-RFIDSec2008-Talk.pdf.
4. Juels, A., Molnar, D., Wagner, D. Security and privacy issues using e-passports. *International Association of Cryptologic Research Cryptology*, ePrint Archive; http://eprint.iacr.org/2005/095.pdf.
5. Timmer, J. Faking passport RFID chips for $120. *Ars Technica.* (Aug. 7, 2008); http://arstechnica.com/security/news/2008/08/faking-passport-rfid-chips-for-120.ars.
6. U.S. Department of State. *The U.S. Electronic Passport Frequently Asked Questions.* (Feb. 27, 2009); http://travel.state.gov/passport/eppt/eppt_2788.html#Twelve.



**Alan Ramos** graduated from Harvard University in June 2009 and is an independent media consultant.

**Doug Lloyd** graduated from Harvard University in June 2009 and is starting law school at Northeastern University, Evanston, IL.

**Katherine O'Leary**, **William Scott**, and **Weina Scott** are undergraduates at Harvard University, Cambridge, MA.

**Jim Waldo** is Professor of the Practice at Harvard University, Cambridge, MA, where he teaches distributed computing and topics in the intersection of policy and technology in the department of computer science. He is also a Distinguished Engineer with Sun Microsystems Laboratories, where he investigates next-generation large-scale distributed systems.

DOI:10.1145/1610252.1610269

Article development led by acmqueue
queue.acm.org

**DNS is many things to many people—perhaps too many things to too many people.**

BY PAUL VIXIE

# What DNS Is Not

A DOMAIN NAME SYSTEM (DNS) is a hierarchical, distributed, autonomous, reliable database. The first and only of its kind, it offers real-time performance levels to a global audience with global contributors. Every TCP/IP traffic flow including every Web page view begins with at least one DNS transaction. DNS is, in a word, glorious.

To underline our understanding of what DNS *is*, we must differentiate it from what it *is not*. The Internet economy rewards unlimited creativity in the monetization of human action, and fairly often this takes the form of some kind of intermediation. For DNS, monetized intermediation means lying. The innovators who bring us such monetized intermediation do not call what they sell lies, but in this case it walks like a duck and quacks like one, too.

Not all misuses of DNS take the form of lying. Another frequently seen abuse is to treat DNS as a directory system, which it is not. In a directory system one can ask approximate questions and get approximate answers. Think of a printed telephone white pages directory here: users often find what they want in the printed directory not by knowing exactly what the listing is but by starting with a guess or a general idea. DNS has nothing like that: all questions and all answers are exact. But DNS has at least two mechanisms that can be misused to support approximate matching at some considerable cost to everybody else, and a lot of that goes on.

## Stupid DNS Tricks

The first widespread form of a DNS

lie was to treat DNS lookups as mapping requests. Content distribution networks (CDNs), such as Akamai, and Web optimizer products, such as Cisco Distributed Director, treat incoming DNS lookups as opportunities to direct the activities of Web browsers. Using the IP source address of a DNS request, these products and services try to guess the proximity of the requester to each of many replicated content servers. Based on the measured load of each content server's system and network, and on an estimate of each content server's proximity to that requester, a DNS response is crafted to direct that requester to the closest or best content server for that URI domain.

Problems abound from this approach, but none affects the CDN operator's revenue. First and foremost it is necessary to defeat or severely limit caching and reuse of this policy-based data ("DNS lies"). Caching and reuse, which once were considered essential to the performance and scalability of DNS, would allow a policy-based response intended for requester A also to be seen by requester B, which might not otherwise receive the same answer—for example, when server loads have changed and there's a new balance. The effects of this noncaching are a higher DNS request rate (perhaps leading to higher revenue for CDNs that charge by the transaction) and more network load for access-side networks and a slightly higher floor for average transaction time.

Furthermore, it has never been wise to assume that a DNS request's IP source address gives any hint of an end-system Web browser's network location. This is because DNS requests heard by a CDN come from recursive DNS servers as a result of cache misses; they do not come from end systems themselves. Some ISPs regionalize their recursive name servers, allowing CDNs to encode rules improving the quality of their estimates. Many recursive name servers are per-country or per-continent or even per-hemisphere, however, so it's always necessary for a CDN to deploy well-connected supernodes, and these always end up hearing a lot of out-of-region requests.

The primary benefit of a CDN is the same as gimmick-free outsourcing: it gives a content owner somebody to sue if things don't go well. That DNS system performance and stability must pay the price for such liability shielding is at best unfortunate. Given a CDN still requires supernodes that will hear many out-of-region requests, a gimmick-free approach here would be to answer DNS truthfully and let existing pseudorandom distribution mechanisms do their work. Noting that there is no patent on the existing pseudorandomization technologies and that nobody ever got fired for buying a CDN, we can expect to see more content distributed this way in the decades to come.

### NXDOMAIN Remapping

Fairly often, as in millions of times per second worldwide, somebody looks up a domain name in DNS that isn't there. Maybe this is a user at a Web browser making a typographical error, or maybe there's a broken link on a Web site, or maybe a hardware or software error is causing nonexistent names to go into DNS requests. One way or another, the answer is generally supposed to be NXDOMAIN (sometimes written as RCODE=3). These negative answers are cacheable, as is any other kind of DNS information, since DNS is designed to express truth, not policy. A network application (perhaps a Web browser, or mail server, or indeed anything at all that uses TCP/IP flows to do its business) that gets back one of these negative responses is supposed to treat it as an error and reject its own underlying work item that led to this lookup. For a Web browser, rejection takes the form of an "error page." For a mail server, rejection takes the place of "bounced email." Every TCP/IP application, large or small, new or old, knows how to cope with NXDOMAIN.

The Web has changed the rules. Though the Web is young—and though the Internet was here before the Web and will be here after the Web and is much larger than the Web—the fact remains that the Web is what end users are looking at. Advertisers have a whole language to describe the value of end users, with words such as "impressions," "click-throughs," and "eyeballs." Why on earth, these advertisers ask, would you ever send back an NXDOMAIN if an impression was possible? So it is, increasingly, that in place of the NXDOMAIN your applica-

**Author Paul Vixie, noted by *Wired* magazine as "the godfather of DNS," has been solving DNS errors and mysteries for over 20 years.**

tion knows how to handle, if you ask for a name that does not exist, you'll get a positive (deceptive; false; lying) answer that your application also knows how to handle.

For example, if I ask my own recursive name server for a name that does not exist, it will tell me NXDOMAIN. If I ask OpenDNS's recursive name server for a name that does not exist, it will send me a NOERROR response with an answer pointing at an advertising server. Note that I'm using OpenDNS as a convenient example; it did not invent this technique. Indeed, Nominum and other DNS vendors now sell an add-on to their recursive name service products to allow any ISP in the world to do this, and a growing number of ISPs are doing it. Why so many? The answer is


PHOTOGRAPH BY JOHN KEATLEY

simply whoever remaps these NXDOMAIN responses gets the impression revenue. There are unverified claims that some ISPs are blocking access to OpenDNS and/or all non-ISP name servers in order to force their customers to use the ISP's own name server. I say unverified, but I find the claims credible—ISPs have wafer-thin margins and if they see this kind of manna going out the door, they can't just let it happen.

To demonstrate the extreme desire to capture this revenue, a true story: A few years ago VeriSign, which operates the .COM domain under contract to ICANN (Internet Corporation for Assigned Names and Numbers), added a wild card to the top of the .COM zone (*.COM) so that its authoritative name servers would no longer generate NXDOMAIN responses. Instead they generated responses containing the address of SiteFinder's Web site—an advertising server. The outcry from the community (including your humble narrator) was loud and long, and before ICANN had a chance to file a lawsuit to stop this nonsense, many people had patched their recursive name servers to remap any response from a .COM name server that was not a delegation (for example, telling how to find the Google.com name servers) back into an NXDOMAIN. Some ISPs put logic into their policy-based routers to turn SiteFinder responses into pointers to the ISP's own advertising server instead.

## Damage Control

NXDOMAIN wasn't designed to be a revenue hook—many applications depend on accurate error signals from DNS. For example, consider the "same origin trust model" used for Web cookies. If you're holding a cookie for Google.com and you can be fooled into following a link to KJHSDFKJHSKJHMJHER.GOOGLE.COM, and the resulting NXDOMAIN response is remapped into a positive answer to some advertising server, then you're going to send your Google.com cookie to that advertising server when you send your HTTP GET request there. Not such a bad thing for a Google.com cookie, but a real problem for a BANKOFAMERICA.COM cookie. (Thanks to Dan Kaminsky for telling me about the "same origin trust model" problem.)

Remapping could also cause email to be captured if a mail exchanger (MX) request is captured in this way. Many NXDOMAIN remappers try to avoid this by triggering only on A (address) requests, but to make this work they have to turn off caching, since NXDOMAINs are not type specific and since an SMTP initiator will fall back to type=A if it gets no answer from type=MX. Similar protections (designed to keep lawsuits away while still attracting revenue) include the idea of triggering the remapping logic only if the query domain begins with WWW.—but as far as I know there are a lot of typographical errors beginning WW., or ending with .CM, so I do not hold out a lot of hope for it long term. Too much money is involved, and nobody wants to leave it on the table (where, in this case, it belongs).

### Standard Bad Practices

There is at the time of this writing an IETF draft (think: proto-RFC) on the recommended configuration and use of DNS redirect by service providers (http://tools.ietf.org/html/draft-livingood-dns-redirect-00). The goal of this document is to present some rules for how DNS lies should be delivered in order to give all the vendors and operators in this growing market a common frame of service. Some Luddites may feel that the "standard best practice" in this area is simply not to do it at all, but this being unrealistic, we now face standards action. As a standard feature of DNS technology we can expect a day to come when all DNS services are delivered this way and our kids think of end-to-end DNS the way they think of eight-track tapes.

This document makes a substantial contribution to the debate around this feature area by suggesting that opt-out for this service should be a network layer attribute (in other words, associated with one's Ethernet [MAC] address or equipment port number) and not a transport layer attribute. Noting that as with any other kind of information, spam opt-out is not as good for the economy as opt-in, it's valuable to the debate that this IETF draft's authors have said that Web cookies aren't good enough. Others may disagree but at least this point is now on the table. This document also talks about "legally mandated" DNS redirection, which is exactly the nightmare it sounds like it is and that we can all hope becomes a historical curiosity as rapidly as possible.

The absolutely best part of this IETF draft is Section 10—DNSSEC Considerations—that ends as follows: "So the only case where DNS security extensions cause problems for DNS Redirect is with a validating stub resolver. This case doesn't have widespread deployment now and could be mitigated by using trust anchor, configured by the applicable ISP or DNS ASP, that could be used to sign the redirected answers. As noted above in Section 9.7, such improper redirection of valid responses may also cause DNSSEC trust verification problems.

### A Rescue Being Thought Of

Fifteen years ago a bunch of ivory tower theoreticians got together at IETF and said, "Let's secure DNS." The threat model has evolved over time, and now this set of protocol enhancements (DNSSEC) is more or less ready for deployment and more or less allows for the possibility that DNS liars will be caught and ignored. Noting that DNSSEC has taken too long and is a committee-based horror in its inelegance and complexity, here's how it's supposed to work and how it may help curtail the current market in DNS lies.

In DNS, data producers are the authoritative name servers, each of which is the delegated authority for one or more zones. For today, think of a DNS zone as everything at or below a certain name, so, for example, www.google.com is in the Google.com zone. DNSSEC allows these zones to be signed and verified using public-key cryptography. The private (signing) key is used by the editor of the zone to generate signature records for each set of real records. The public key is used by recursive name servers to verify that the data they receive was signed by the holder of the corresponding private key. Public (verification) keys are published using DNS itself, by including each zone's key in the zone's parent zone—so the public key for the Google.com zone is published in the COM zone and so on. I'm deliberately skipping a long and unpleasant story about where the public key for COM is supposed to be published, since it's not germane to this article.

In theory, an end-system owner who does not like being lied to can work cooperatively with zone editors who don't like their zones getting lied about if each of them deploys DNSSEC. An application that is supposed to receive an NXDOMAIN but that today receives a pointer to an advertising server would in a DNSSEC world receive a "signature not present" error. This would be an error because DNSSEC has a way to inform a validator that a signature should have been present. Note that in the vast majority of cases zone editors don't care whether their zones are being lied about, and, therefore, DNSSEC will remain silent most of the time. Consider also that this was not the original DNSSEC threat model; we really thought we had to stop on-the-wire corruption such as that discovered by Dan Kaminsky in 2008, and the idea of stopping in-the-middlebox corruption such as NXDOMAIN remapping really is just gravy.

DNSSEC will complicate life for CDN providers using Stupid DNS Tricks, but it won't end that war since it's still possible to sign every policy-based answer and keep all the answers and signatures available, and still send different


**Looking for CACM? Should an inquiry inadvertently ask for cacm.com rather than cacm.org, the response will be redirected to this advertising scene as enabled by NXDOMAIN remapping.**

answers to the same question based on requester identity and policy, and have the signatures all be perfectly valid.

DNSSEC will also complicate life for sys admins and application developers. We (ISC—the BIND people) are doing what we can to improve on that in BIND 9.7, and there are plenty of other service and technology providers in the space as well. The killer app for DNSSEC will be a Web browser and Web server that can authenticate to each other without using X.509 (volunteers are hereby encouraged to get together and try to make that happen).

### Directory Services

Browser implementers including Microsoft and Mozilla have begun doing DNS queries while collecting URIs from their graphical front end in order to do fancy "autocompletion." This means that during the typing time of a URI such as http://www.cnn.com/, the browser will have asked questions such as W, WW, WWW, WWW.C, WWW.CN, WWW.CNN, and so on. It's not quite that bad, since the browsers have a precompiled idea of what the top-level domains are. They won't actually ask for WWW.C, for example, but they are now asking for WWW.CN, which is in China, and WWW.CNN.CO, which is in Colombia.

Although one simple-sounding solution is for Microsoft and Firefox to buy some name-server hardware and network links for China and Colombia (and no doubt many other affected top-level domain operators), that won't stop the information leak or remove this stupid and useless traffic from the rest of the network. Since the truly best solution is, as usual, stop doing this stupid thing—and we all know that isn't going to happen—perhaps this behavior can be made optional, and then we can just argue about what the default (opt-in vs. opt-out) should be. This is the first time in the history of DNS that someone has used it prospectively, to find out if what has been typed is or isn't a valid domain name, in order to support something like autocompletion. As in so many other novel uses of DNS, this is not what it was designed for.

Had DNS been designed with this in mind, one of the ways we would be able to tell is that domain names would be written from highest- to lowest-order term (COM.CNN.WWW). This would allow partial name completion just as happens in graphical file system browsers. Absent a complete redesign, which won't happen in our lifetime because of the size and usefulness of the installed base, all we can do is ask browser implementers to be smarter and prepare for more DNS traffic on our networks.

**DNSSEC will complicate life for CDN providers using Stupid DNS Tricks, but it won't end that war since it's still possible to sign every policy-based answer and keep all the answers and signatures available, and still send different answers to the same question based on requester identity and policy, and have the signatures all be perfectly valid.**

### Conclusion

What DNS is not is a mapping service or a mechanism for delivering policy-based information. DNS was designed to express facts, not policies. Because it works so well and is ubiquitous, however, it's all too common for entrepreneurs to see it as a greenfield opportunity. Those of us who work to implement, enhance, and deploy DNS and to keep the global system of name servers operating will continue to find ways to keep the thing alive even with all these innovators taking their little bites out of it.

These are unhappy observations and there is no solution within reach because of the extraordinary size of the installed base. The tasks where DNS falls short, but that people nevertheless want it to be able to do, are in most cases fundamental to the current design. What will play out now will be an information war in which innovators who muscle in early enough and gain enough market share will prevent others from doing likewise—DNS lies vs. DNS security is only one example. C

**Related articles on queue.acm.org**

**DNS Complexity**
*Paul Vixie*
http://queue.acm.org/detail.cfm?id=1242499

**Improving Performance on the Internet**
*Tom Leighton*
http://queue.acm.org/detail.cfm?id=1466449

**Paul Vixie** is president of Internet Systems Consortium (ISC), a nonprofit company that operates the DNS F root name server and publishes the BIND software used by 80% of the Internet for DNS publication. He is also chairman of American Registry for Internet Numbers (ARIN), a nonprofit company that allocates Internet number resources in the North American and Caribbean regions. Previously, he was a founder and president of PAIX, the first neutral commercial Internet exchange; SVP/CTO of AboveNet; and founder of the first anti-spam company (MAPS LLC) in 1996.

DOI:10.1145/1610252.1610270

Article development led by acmqueue
queue.acm.org

**How do we develop software to make the most of the promise that asymmetric multicore systems use a lot less energy?**

BY ALEXANDRA FEDOROVA, JUAN CARLOS SAEZ, DANIEL SHELEPOV, AND MANUEL PRIETO

# Maximizing Power Efficiency with Asymmetric Multicore Systems

IN COMPUTING SYSTEMS, a CPU is usually one of the largest consumers of energy. For this reason reducing CPU power consumption has been a hot topic in the past few years in both the academic community and the industry. In the quest to create more power-efficient CPUs, several researchers proposed an asymmetric multicore architecture that promises to save a significant amount of power while delivering similar performance as conventional symmetric multicore processors.

An asymmetric multicore processor (AMP) consists of cores that use the same instruction set architecture (ISA) but deliver different performance and have different power characteristics. Using the same ISA on all cores means running the same binary on all cores without the need to compile code with a different compiler for each core type. This stands in contrast to heterogeneous-ISA systems, such as IBM's Cell or Intel's Larrabee, where the cores expose different ISAs, so the code must be compiled separately for each core type. Heterogeneous-ISA systems are not the focus of this article.

A typical AMP consists of several fast and powerful cores (high clock frequency, complex out-of-order pipeline, and high power consumption) and a large number of slower low-power cores (low clock frequency, simple pipeline, and low power consumption). Complex and powerful cores are good for running single-threaded sequential applications because these applications cannot accelerate their performance by spreading the computation across multiple simple cores. Abundant simple cores, on the other hand, are good for running highly scalable parallel applications.

Because of performance/power trade-offs between complex and simple cores, it turns out to be much more efficient to run a parallel application on a large number of simple cores than on a smaller number of complex cores that consume the same power or fit into the same area. In a similar vein, complex and powerful cores are good for running CPU-intensive applications that effectively use those processors' advanced microarchitectural features, such as out-of-order super-scalar pipelines, advanced branch prediction facilities, and replicated functional units. At the same time, simple and slow cores deliver a better trade-off between energy consumption and performance for memory-intensive applications that spend a majority of their execution time fetching data from off-chip memory and stalling the processor.

A symmetric multicore processor (SMP) includes the cores of only one type: either the complex and powerful ones, as in the Intel Xeon or AMD Opteron processors, or the simple and

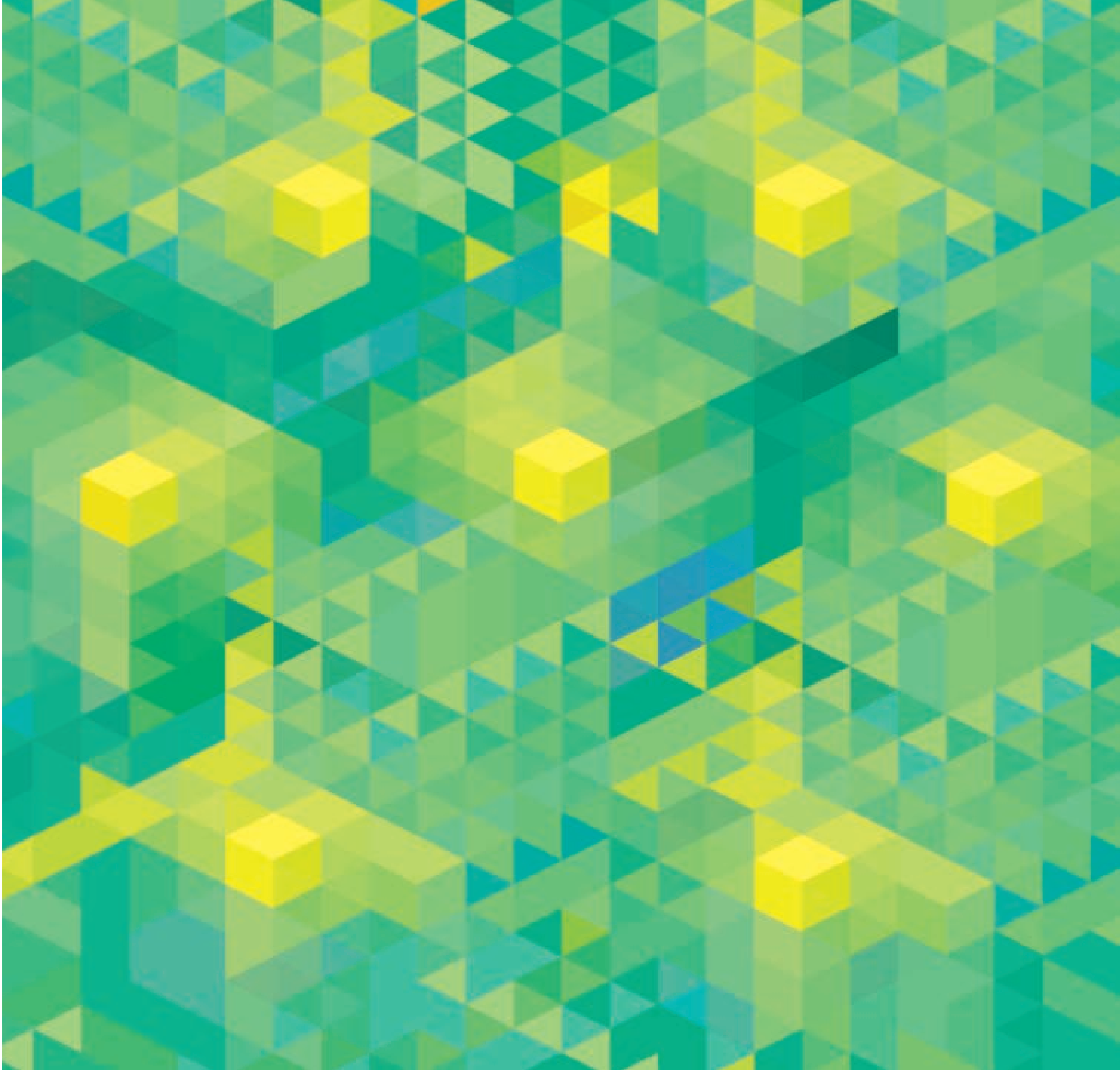


ILLUSTRATION BY ANDY GILMORE

lower-power ones, as in Sun's Niagara. So in terms of providing the optimal performance/power, an SMP is ideally suited for some applications but not all. Having cores of different types in a single processor enables optimizing performance per watt for a wider range of workloads. Having cores of different types on an AMP enables us to employ specialization (that is, we can use each type of core for the type of computation where it delivers the best performance/energy trade-off). Specialization enables maximizing the overall efficiency of the system and as a result delivers better performance per watt and per area.

Although single ISA AMP systems are not yet being built, much of the literature investigating the properties of these systems has originated from major hardware players such as Intel[1,7] and HP,[5,6] indicating that within the industry there is interest in this architecture. Furthermore, existing SMP systems can be configured to be asymmetric if one wants to reduce the amount of power consumed by the processor. For example, configuring some of the cores of an SMP to run at a lower-than-maximum voltage and frequency (via the dynamic voltage and frequency scaling facilities available on most modern CPUs) makes the CPU consume less power and makes the system asymmetric. In that case it is crucial to understand how to get the maximum performance on this asymmetric system in order to minimize performance losses associated with running some cores at a lower Hertz.

Given the emergence of AMPs, how can you best exploit them to maximize power efficiency (performance per watt)? Our goal is to shed light on some of the challenges software developers will likely face while trying to achieve this, and to provide some practical advice on how to maximize exploitation of AMPs as a result. To that end, we provide several examples demonstrating how to employ specialization on AMP systems in order to maximize performance. One area that will be of particular importance is the design of scheduling algorithms for operating systems that wish to take full advantage of AMPs. We also discuss our experience in investigating the design of such algorithms. We discuss a few surprising findings that we have discovered about the implementation of these scheduling strategies, which we think will be important to those who are developing or adapting an operating system for this upcoming class of processor and platform.

### Specialization on AMPs

Efficiency of AMP systems could be improved using two kinds of specialization: the first caters to diversity in thread-level parallelism; the second caters to microarchitectural diversity of the workload.

*Catering to diversity in thread-level parallelism.* Diversity in thread-level parallelism refers to the two broad categories into which applications can be classified: scalable parallel applications and sequential applications. Scalable parallel applications use multiple threads of execution, and increasing the number of threads typically leads to reduced execution time or increased amount of work performed in a unit of time. Sequential applications, on the other hand, typically use only one or a small number of threads and it is difficult to structure the computation such that it runs efficiently in a multithreaded environment. In addition to purely parallel or purely sequential applications, there is a hybrid type, where an application might have phases of highly parallel execution intermixed with sequential phases.

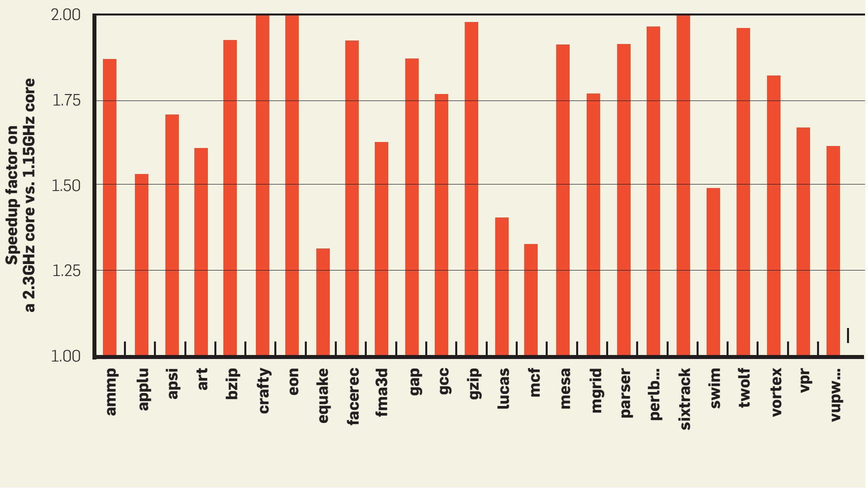
These two types of applications require different types of processing cores to achieve the best trade-off in performance and energy consumption. Suppose we have a scalable parallel application with a choice of running it on a processor either with a few complex and powerful cores or with many simple low-power cores. For example, suppose we have a processor with four complex and powerful cores and another area-equivalent and power-budget-equivalent processor consisting of 16 simple/low-power cores. Suppose further that each simple core delivers roughly half the performance of one complex core. (The numbers to estimate the conversion ratios of performance and power in complex vs. simple cores were obtained from Hill and Marty.[4]) We configure the number of threads in the application to equal the number of cores, which is a standard practice for compute-intensive applications. If we run this parallel application on the processor with complex cores, then each thread will run roughly twice as fast as the thread running on the processor with simple cores (assuming that threads are CPU-intensive and that synchronization and other overhead is negligible), but we can use only four threads on the complex-core processor vs. 16 threads on the simple-core processor. Since using additional threads results in a proportional performance improvement in this application, we get twice as much performance running on a simple-core processor as on a complex-core processor. Recalling that these two processors use the same power budget, we achieve twice as much performance per watt.

Contrast this to running a sequential application, which cannot increase its performance by using additional threads. Therefore, using a single thread, it will run twice as slow on a simple-core processor than on a complex-core processor, meaning we get twice as much performance per watt running on the complex-core system. An experienced reader will observe that power consumption on a simple-core system for this single-application workload could be reduced by turning off unused cores. Unfortunately, it is not always possible to turn off unused cores completely, especially if they are located in the same power domain as the active cores. Furthermore, an operating-system power manager may be configured to avoid putting the unused cores in a deep sleep state, because bringing the cores up from this state takes time. Thus, if a new application begins running or if the operating system needs a core for execution, then additional latency will be incurred while the dormant core is being brought up in the active power state.

This example demonstrates that applications with different levels of parallelism require different types of cores to achieve the optimal performance-per-watt ratio. AMP systems offer the potential to resolve this dilemma by providing the cores of both types. Another advantage of having both "fast" and "slow" cores in the system is that the fast ones can be used to accelerate sequential phases of parallel applications, mitigating the effect of sequential bottlenecks and reducing effective serialization. Hill and Marty demonstrated that for parallel applications with sequential phases AMPs can potentially offer performance significantly better than SMPs, as long as sequential phases constitute as little as 5% of the code.[4]

*Catering to microarchitectural diversity of the workload.* The relative benefit that an application derives from running on a fast core rather than a slow

**Figure 1. Relative speedup experienced by applications from the SPEC CPU2000 benchmark suite from running on a fast core (2.3GHz) vs. a slow core (1.15GHz) of an emulated AMP system. The maximum achievable speedup is a factor of 2. The more memory-intensive the application is the less speedup it experiences. More details on the experimental setup can be found in Shelepov.[11]**



one depends on the microarchitectural properties of the program.[2,5,6,11] Some programs are very efficient at using the CPU pipeline: they have a high amount of instruction-level parallelism, meaning that a processor can issue many instructions in parallel without running out of work. These programs show good locality of memory accesses. As a result they rarely access the main memory and thus rarely stall the processor. We refer to these programs as CPU-intensive.

At the other extreme are programs that use the CPU pipeline very inefficiently. They typically have a high processor cache-miss rate and thus stall the CPU pipeline, because they have to wait while their data is being fetched from main memory. We refer to these programs as memory-intensive . (Note that this is not the same as an I/O-bound application, which often relinquishes the CPU when it must perform device I/O. A memory-intensive application might run on the CPU 100% of its allotted time, but it would use the CPU inefficiently.)

CPU-intensive programs use the hardware of fast cores very efficiently; thus, they derive relatively large benefits from running on fast cores relative to slow cores. Memory-intensive applications, on the other hand, derive relatively little benefit from running on fast cores. Figure 1 shows some example speedup ratios of applications in the SPEC CPU2000 suite on an emulated AMP system. An SMP was used to emulate this AMP system using dynamic frequency scaling. The fast core was emulated by running a core at 2.3GHz; the slow core was emulated by using the frequency of 1.15GHz. Note that some applications experience a 2× speedup, which is proportional to the difference in the CPU frequency between the two processors. These are the CPU-intensive applications that have a high utilization of the processor's pipeline functional units. Other applications experience only a fraction of the achievable speedup. These are the memory-intensive applications that often stall the CPU as they wait for data to arrive from the main memory, so increasing the frequency of the CPU does not directly translate into better performance for them. For example, a memory-intensive application equake speeds up by only 25% when running on the fast core.

**Having cores of different types in a single processor enables optimizing performance per watt for a wider range of workloads. Having cores of different types on an AMP enables us to employ specialization.**

For systemwide efficiency, it is more profitable to run CPU-intensive programs on fast cores and memory-intensive programs on slow cores. This is what catering to microarchitectural diversity of the workload is all about. Recent work from the University of California, San Diego and HP demonstrated that AMP systems can offer up to 63% better performance than can an SMP that is comparable in area and power, provided that the operating system employs a scheduling policy that caters to the microarchitectural diversity of the workload.[6]

### Asymmetry-Aware Scheduling

Employing specialization is the key to realizing the potential of AMP systems. Specialization on AMP systems will not be delivered by the hardware; it is up to the software to employ asymmetry-aware scheduling policies that tailor asymmetric cores to the instruction streams that use them most efficiently. A thread scheduler must be aware of the asymmetric properties of the system and assign threads to cores in consideration of the characteristics of both. In this section we report on our experience in designing and implementing such asymmetry-aware scheduling algorithms in a real operating system. We first describe an algorithm that caters to diversity in thread-level parallelism and then an algorithm that caters to diversity in the workload's microarchitectural properties.

**A Scheduler Catering to Diversity in Thread-Level Parallelism.** The idea behind our parallelism-aware (PA) scheduler is simple: it assigns threads running sequential applications or sequential phases of parallel applications to run on fast cores and threads running highly scalable parallel code to run on slow cores. The following example demonstrates that the PA scheduling policy can achieve a much better system efficiency than an asymmetry-unaware policy. We emphasize that the goal in using the PA policy is to maximize systemwide efficiency, not to improve performance of particular applications. As a result this policy will be inherently unfair: some threads will have a higher priority than others in running on fast cores. Implications of a policy that equally shares fast cores among all threads are demonstrated in our earlier study.[9]

Consider an asymmetric processor with one fast core and nine slow cores, where the fast core delivers approximately twice as much single-threaded performance as the slow core. Let's say we run a workload of one sequential application and one parallel application, and the parallel application has nine threads. Under a naïve scheduling policy that equally shares fast and slow cores among threads, each thread will run on the fast core 10% of the time and on the slow core 90% of the time. (Note that existing operating-system schedulers would not share complex and simple cores equally. Distribution of time on different core types would be arbitrary. We assume a policy that shares cores equally to simplify the example.) To simplify comparison of different scheduling policies, we use as our performance measure the overall workload speedup relative to running all threads on slow cores the entire time. Under this naïve policy, each thread will speed up by 1.1× relative to running on a slow core (to work this out, consider that each thread runs at a speed of 2× for 10% of the time and at a speed of 1× for 90% of the time), and the workload-wide speedup will also be 1.1×. Note that when computing the speedup for a parallel application we assume that the speedup for the entire application is close to the average speedup of its threads rather than the aggregate speedup—this assumption is reasonable if threads of a parallel computation are working on a common task as opposed to performing unrelated tasks requiring no inter-thread communication.

Under a PA policy, the single-threaded application will run on the fast core for the entire time, and the threads of the parallel application will run on slow cores. As a result, the single-threaded application will speed up by a factor of 2×, and the parallel application will have no speedup (1×). The average speedup for the two applications will be 1.5×, or 40% better than under the naïve policy.

As another example, consider a parallel application with 50% of its code executing sequentially. An asymmetry-unaware scheduler may fail to assign the bottleneck sequential phase to run on a fast core, but a PA scheduler would make sure to accelerate it. Suppose the fast core runs twice as fast as the slow core: the PA scheduler would deliver up to 25% performance improvement to that application (see Figure 2).

Figure 3 shows the performance of a number of parallel applications on an emulated AMP system with our implementation of a PA scheduler in OpenSolaris relative to the default asymmetry-agnostic scheduler in that operating system. To emulate AMP we use a real multicore system (AMD Opteron with 16 cores), and we emulate fast cores by

**Figure 2. An illustration of how a PA scheduler would accelerate a parallel application limited by a sequential bottleneck on an AMP processor.**

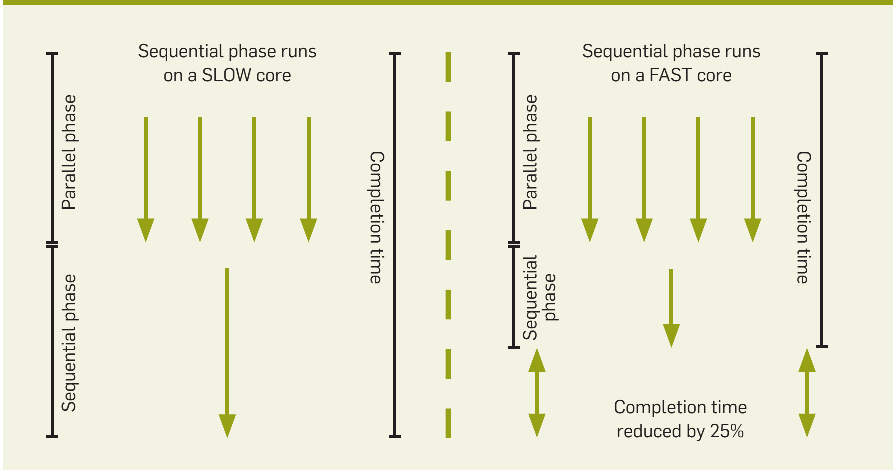



**Figure 3. Speedup achieved with a PA algorithm over the asymmetry-agnostic default scheduler on an emulated AMP system.**

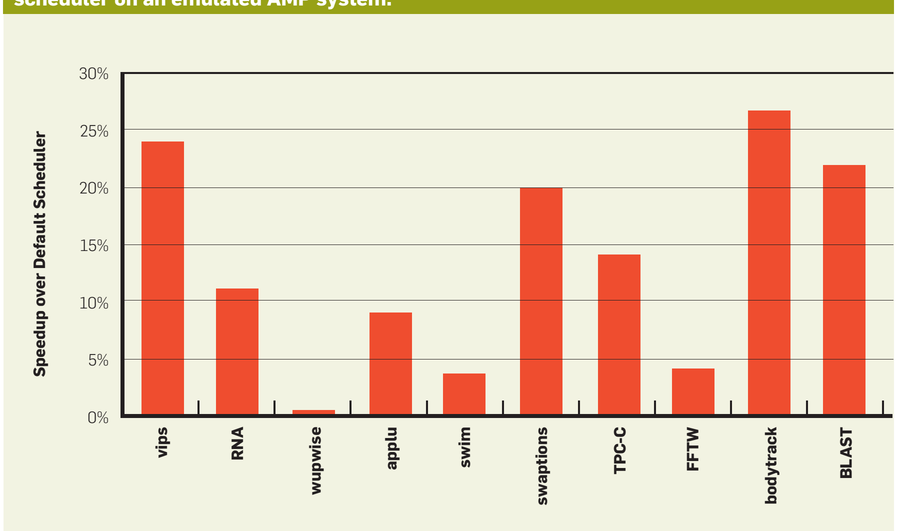
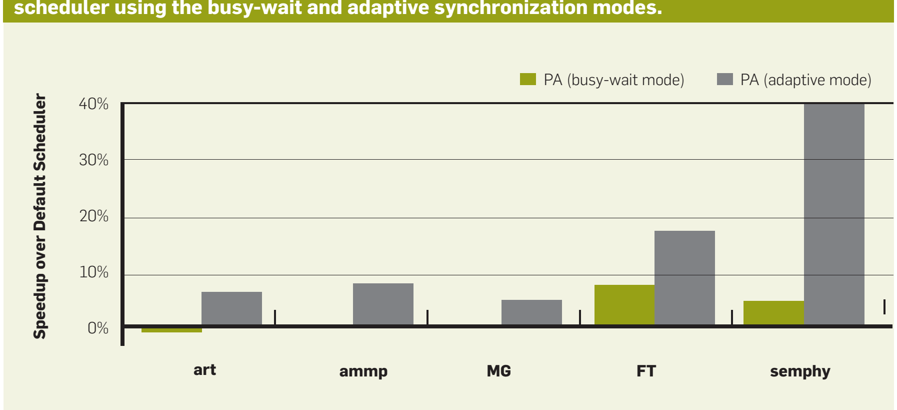



**Figure 4. Speedup achieved with a PA algorithm over the asymmetry-agnostic default scheduler using the busy-wait and adaptive synchronization modes.**



using a high clock frequency (2.3GHz) and slow cores by using a low frequency (1.15GHz). The implementation of the algorithm and the experimental platform are described in more detail in a previous work.[10] In this experiment we use four fast and 12 slow cores. The applications used in Figure 3 are drawn from several benchmark suites, such as SPEC OpenMP 2001, PARSEC, MineBench, and NAS. RNA is a bioinformatics application performing RNA sequence matching.

The figure shows a variety of speedup values. Applications with significant sequential phases (40%–60%) experience a performance improvement of up to 26% relative to an asymmetry-unaware scheduler. Applications with small sequential phases (wupwise, for example) experience no speedup from the PA policy, because they do not stand to benefit from acceleration of sequential phases on fast cores.

Here, we discuss two main challenges involved in delivering the benefits of the PA scheduling policy to applications: effectively detecting sequential and parallel phases and avoiding performance overhead that may result from cross-core migrations.

*Detecting sequential and parallel phases.* The simplest way to detect parallel and sequential phases in an application is to use the runnable thread count as a heuristic. If an application uses a large number of threads, then its runnable thread count will be high, and this application would be in a parallel phase. Conversely, an application with a single runnable thread would be in a sequential phase. The great property of the runnable thread count is that in modern multithreading environments it is visible to the operating system, because these systems map application-level threads to kernel-level threads. Therefore, by monitoring the runnable thread count an operating system can distinguish between parallel and sequential phases in applications.

Unfortunately, in some situations using the runnable thread count for detection of sequential phases might not work. In particular, an application could be running nonscalable code while still using a large number of runnable threads. We describe two scenarios where this might happen and discuss potential remedies.

In one scenario, an application might be susceptible to an external scalability bottleneck—for example, as a result of memory bandwidth contention. In this case the system memory bus is saturated, and using additional threads does not speed up the application because those threads do not contribute to useful computation. A sensible way to solve this problem is to reduce the number of threads used in an application to the point where the application operates at its peak efficiency. Essentially, this boils down to configuring the number of threads properly in a parallel application. Suleman et al. describe a technique called feedback-driven threading, which allows you to dynamically determine the optimal thread count for parallel applications.[12]

In another scenario, an application might be limited by internal scalability bottlenecks: for example, there might be a load imbalance where some threads do more work than others or necks where one thread executes the code in a critical section while other threads wait. When threads wait they may either block, relinquishing the CPU, or busy-wait, spinning on the CPU in a tight loop. If threads block, then the runnable thread count is reduced and any such reduction is visible to the operating system; but if threads busy-wait, sequential phases might be hidden from the operating system.

Whether the application uses blocking or busy-waiting depends on the implementation of the synchronization primitives, which are used to construct critical sections or barriers. Busy-waiting makes sense on a multiprocessor when the wait times are expected to be short. Blocking a thread is an expensive operation that should be avoided during short waiting periods. If the wait time is long, however, blocking is usually preferred so as to avoid wasting CPU resources. One of the most popular strategies used in synchronization libraries is the adap-

**Figure 5. Two configurations of an AMP system. Large squares represent memory domains with one or more cores and a last-level cache (LLC) inside the domain. "Fast" cores are denoted by large red boxes, "slow" cores are denoted by small blue boxes.**

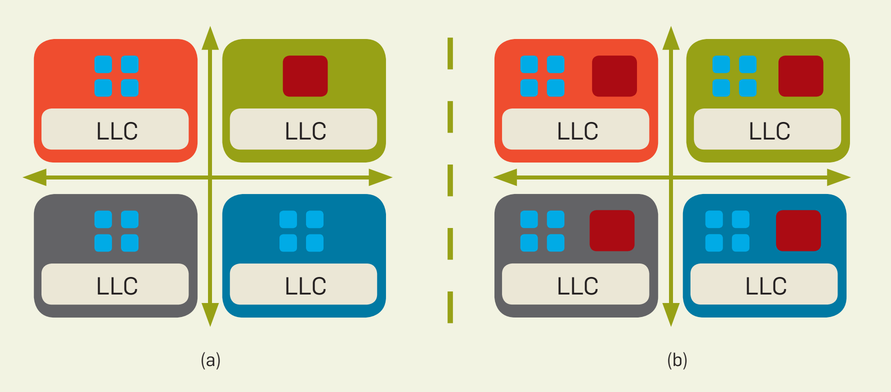
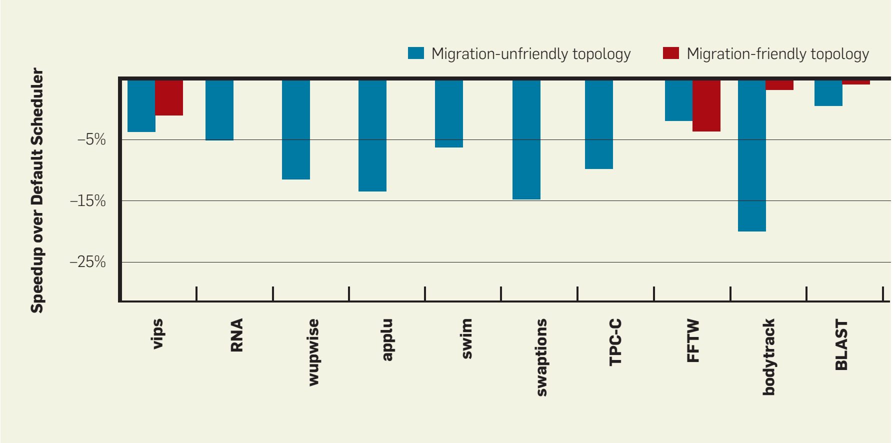



**Figure 6. Performance overhead relative to the default scheduler on a migration-unfriendly topology (like in Figure 5a) and on a migration-friendly topology (like in Figure 5b, but with three slow cores in each memory domain). Lower numbers denote higher overhead.**



for a while and then blocks. On an AMP system, using the right synchronization waiting mode is crucially important. If busy-waiting is used, a PA scheduler would not be able to detect and accelerate the bottleneck serial phases.

Figure 4 shows the performance of selected OpenMP applications that could be configured to use either the adaptive or the busy-wait mode in the synchronization library. We show the performance under the PA scheduler relative to the default OpenSolaris scheduler. We use the same emulated AMP setup as described for Figure 3, but in this case we configure the system with one fast core and 12 slow cores. For each application, we show the performance under the busy-wait and the adaptive modes.

As the figure shows, the PA scheduler can deliver significant performance improvements (up to 40%) for applications with large sequential phases but only if sequential phases are exposed to the scheduler via the adaptive synchronization mode. When the busy-wait mode is used, the scheduler is unable to detect sequential phases and accelerate them on the fast core of the AMP.

An alternative to using the adaptive mode would be to implement a new synchronization mode where the synchronization primitive explicitly notifies the scheduler when a thread begins to spin.[10] Using this waiting mode will further ensure that spinning threads do not waste the resources of fast cores. When making changes to the synchronization library is not an option, however, using adaptive synchronization mode will do the job.

*Avoiding the overhead.* A more difficult challenge in implementing PA or other asymmetry-aware algorithms is to avoid the overhead associated with migrating threads across the cores. Any asymmetry-aware algorithm relies on cross-core migrations to deliver the benefits of its policy. For example, the PA algorithm must migrate a thread from a slow core to a fast core if it detects that the thread is executing a sequential phase.

Migrations are an essential tool of asymmetry-aware algorithms, but unfortunately they can be quite expensive. The AMP shown in Figure 5a consists of several memory domains, as is usually the case with modern multicore processors. A memory domain is defined to contain cores that share an LLC (last-level cache). LLC is the last "line of defense" on the frontier between the CPU and the main memory. Thus, if the required data is not in the LLC, then the processor has to fetch it from the main memory, which takes hundreds of CPU cycles and slows down the computation considerably. In contrast, fetching data from an LLC takes only a few tens of processor cycles. Therefore, we want to minimize the number of accesses to the main memory and try to satisfy data requests from an LLC or other CPU caches as frequently as possible.

In Figure 5a, the fast core is located in a different memory domain from the slow cores, so every time the scheduler migrates a thread to the fast core, the thread loses the data accumulated in the LLC of the slow core's memory domain and must fetch the data from the main memory. (Depending on the implementation of the processor, the thread might have to fetch the data from its old LLC, not from memory, but that is still more expensive than fetching it from the LLC in its current memory domain.) As we will show later, this can cause significant performance overhead.

Consider now Figure 5b, which depicts a different AMP system where each fast core is located in the same memory domain as several slow cores. This architecture makes it easier for the scheduler to avoid cross-memory-domain migrations. In this case, a scheduler will try to migrate a thread to a fast core that is within the same memory domain as the slow core where the thread was previously running, thus enabling the thread to reuse the data in the LLC.

Figure 6 shows performance overhead experienced by several applications on two emulated AMP systems: one is a migration-unfriendly system configured like the system in Figure 5a; and another is the migration-friendly system configured like the system in Figure 5b, except that each domain has only three slow cores because of the limitations of the experimental hardware. Our PA scheduler is designed to be topology-aware, meaning that it at-

**Figure 7. Process involved in modeling relative speedup.**

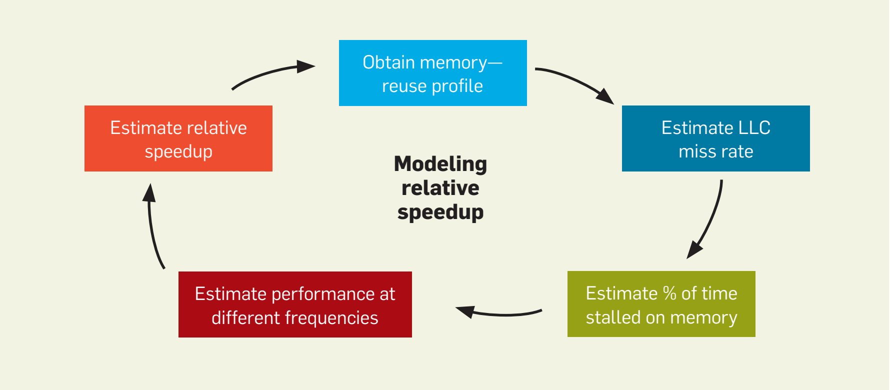
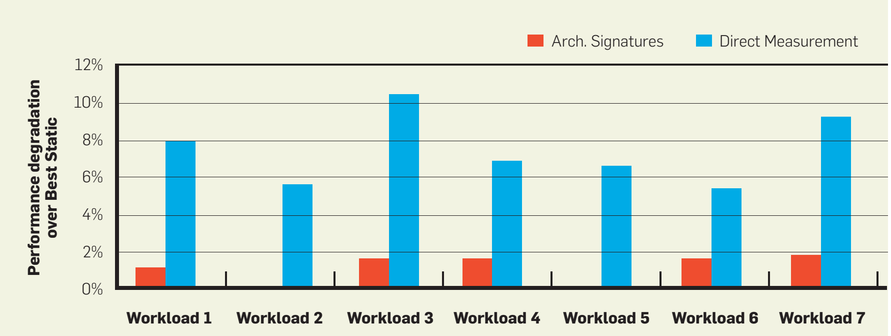



**Figure 8. Performance degradation over the best static scheduling assignment from two approaches used to determine relative speedup of threads on cores of different types. The approach based on architectural signatures outperforms the direct measurement approach due to a lower runtime overhead.**



tempts to avoid cross-memory-domain migrations whenever possible. We measured the overhead by making the designated fast cores on these two systems run at the same frequency as the slow cores—so no performance gains were to be expected from asymmetry-aware scheduling, but the overhead was still present, since our scheduler still migrated threads across cores "thinking" that the system is asymmetric.

Comparing the performance of applications under the PA scheduler and the default scheduler, we can find out the migration-related performance overhead. In this case, performance degradation under the PA scheduler is equivalent to migration overhead. As Figure 6 shows, performance overhead can be quite significant on a migration-unfriendly system, but it becomes negligible on a migration-friendly system coupled with a topology-aware scheduler.

In summary, a parallelism-aware scheduling policy can deliver real performance improvements on asymmetric hardware for parallel applications limited by sequential phases. The key is to configure the synchronization library to "reveal" the sequential phases to the scheduler. To avoid cross-memory-domain migration overhead, AMP systems should be designed such that fast cores share a memory domain with some of the slow cores and combined with a topology-aware scheduler that minimizes cross-domain migrations.

**A Scheduler Catering to Microarchitectural Diversity.** Remember that the idea of catering to microarchitectural diversity of the workload is to assign CPU-intensive threads (or phases of execution) to fast cores and memory-intensive threads (or phases) to slow cores. Recall from Figure 1 that CPU-intensive code will experience a higher relative speedup running on fast vs. slow cores than memory-intensive code, so scheduling it on fast cores is more efficient in a cost-benefit analysis. Just like the PA policy, this policy will be inherently unfair: it may improve performance of some applications at the expense of others, but it will improve the efficiency of the system as a whole.

The biggest challenge in implementing such an algorithm is to classify threads or phases of execution as CPU-intensive or memory-intensive at scheduling time. Two approaches were proposed in the research community to address this challenge. The first approach entails running each thread on cores of different types, registering the speedup obtained on a fast core relative to a slow core and using the resulting relative speedup as the measure for classifying the applications. In a scheduling algorithm, a thread with a larger relative speedup would be given preference to run on a fast core, and a thread with a lower relative speedup would be more likely to run on a slow core. Since this approach relies on direct measurement of relative speedup, we refer to it as the direct measurement approach.

**For systemwide efficiency, it is more profitable to run CPU-bound programs on fast cores and memory-intensive programs on slow cores. This is what catering to microarchitectural diversity of the workload is all about.**

A second approach, referred to as the modeling approach, is to model the speedup on a fast vs. slow core using a summary of an application's runtime properties obtained either offline or online. Modeling is less accurate than direct measurement but does not require running each thread on each type of core, thus avoiding potential load imbalance and expensive cross-core migration (we elaborate on these issues later). In an effort to build an asymmetry-aware algorithm that caters to microarchitectural diversity, we have experimented with both methods.

The direct measurement approach manifested several performance problems. Consider a scenario where each thread must be run on each core type to determine its relative speedup. Given that a running thread may switch phases of execution (that is, it may be doing different types of processing at different points in time), this measurement must be repeated periodically; otherwise, the scheduler might be operating on stale data. Since the number of threads will typically be larger than the number of fast cores, there will always be a high demand for running on fast cores for the purpose of remeasuring relative speedup. As a result, threads that are "legitimately" assigned to run on fast cores by the scheduling policy will observe undue interference from threads trying to measure their speedup there. Furthermore, having too many threads "wanting" to run on scarce fast cores may cause load imbalance, with fast cores being busy and slow cores being idle. When we used this direct measurement approach in an asymmetry-aware algorithm, we found that these problems made it difficult to deliver significant performance improvement relative to an asymmetry-agnostic scheduler.[9,11]

The modeling approach involved predicting relative speedup on different core types using certain properties of the running programs. Since we were keen on evaluating this approach on real hardware (simulators put limitations on the length and number of experiments that can be performed), we could experiment only with the asymmetry that was caused by the differences in the clock frequency of different cores. As a result, our relative speedup model was tuned to work for this specific type of asymmetric hardware. (This is the only type of AMP configuration available on today's commercial hardware.) At the same time, we do not see any fundamental reasons why our model could not be adapted to work on other single-ISA asymmetric systems.

Recall that the main factor determining how much speedup a program would obtain from running on a fast core is how memory-intensive the program is. A good way to capture memory-intensity is via a memory reuse profile, a compact histogram showing how well a program reuses its data.[3] If a program frequently reuses the memory locations it has touched in the past, then the memory reuse profile will capture the high locality of reference. If a program hardly ever touches the memory values used in the past (as would a video-streaming application, for example), the memory reuse profile will capture that as well. Memory reuse profiles are so powerful that they can be used to predict with high accuracy the cache-miss rate of a program in a cache of any size and associativity. This is precisely the feature that we relied on in evaluating memory-intensity of programs and building our estimation model.

Without going into much detail, in our scheduling system we associate a memory reuse profile with each thread. We refer to this profile as the architectural signature, since it captures how the program uses the architectural features of the hardware. The idea is that an architectural signature may contain a broad range of properties needed to model performance on asymmetric hardware, but for our target AMP system, using just a memory reuse profile was sufficient. Using that profile, the scheduler predicts each program's miss rate in the LLC, and using that miss rate, it estimates the approximate fraction of CPU cycles that this program will spend waiting on main memory. The scheduler can then trivially estimate the speedup that each program will experience running on a fast core relative to a slow core (see Figure 7). Then the scheduler simply assigns threads with higher estimated speedups to run on fast cores and threads with lower estimated speedups to run on slow cores, making sure to preserve the load balance and fairly distribute CPU cycles. The resulting scheduler is called HASS (heterogeneity-aware signature-supported), and more details about its implementation are available in our earlier publication.[11]

**The best static assignment always results in running the CPU-bound applications on the fast cores and the memory-intensive applications on the slow cores.**

To evaluate how well the approach based on architectural signatures helps the scheduler determine the optimal assignment of threads to cores, we compare the resulting performance with that under the best static assignment. A static assignment is one where the mapping of threads to cores is determined at the beginning of execution of a particular workload and never changed thereafter. The best static assignment is not known in advance, but can be obtained experimentally by trying all static assignments and picking the one with the best performance. The best static assignment is the theoretical optimum for our signature-supported algorithm, since it relies on static information to perform the assignment (the architectural signature) and does not change an assignment once it is determined.

Figure 8 shows the performance obtained using our signature-supported algorithm relative to the best static assignment. We show the overall performance for seven workloads. Each workload is constructed of four SPEC CPU2000 applications, two of which are memory-intensive and two of which are CPU-intensive. Each workload is executed on an emulated AMP system with two fast cores and two slow cores, so one single-threaded application is running on each core. The fast cores run at 2.3GHz, and the slow cores run at 1.15GHz. We used the AMD Opteron (Barcelona) system for this experiment.

The best static assignment always results in running the CPU-intensive applications on the fast cores and the memory-intensive applications on the slow cores. As Figure 8 shows, the signature-supported algorithm is always able to match the best static assignment within a couple of percentage points. In that figure we also report the performance of the direct measurement approach mentioned earlier. It struggles to do as well as the signature-supported algorithm.

Although the signature-supported scheduler performs rather well, using it in a real scheduler involves an important challenge: obtaining the memory reuse profile needed for construction of the architectural signature. We are aware of two methods of obtaining the memory reuse profile: offline (used in our experiments) and online.

Using an offline method we can obtain a profile by running the application through a profiler that is capable of monitoring and summarizing its memory-access patterns (for x86 executables we used the Pin binary instrumentation tool;[8] this approach is described in more detail in our earlier work[11]). The offline-generated profile must be somehow attached to the program binary (for example, by embedding it into the binary itself). Furthermore, it requires cooperation on the part of the developer, who must be responsible for generating the memory reuse profile for typical executions of the program.

The benefit of using offline-generated profiles is that they require no runtime overhead in the scheduling algorithm associated with their generation. The drawback of offline profiles is that it may be difficult to generate them accurately for multithreaded programs, and it may be tricky to account for different program phases. If a program changes its architectural properties according to the phase that it executes, one needs to associate multiple profiles with a single program. Furthermore, if the behavior of the program changes depending on the input, profiles obtained offline may turn out to be inaccurate.

The other method involves generating a memory reuse profile online. Although we have not experimented with the online approach ourselves, a group from the University of Toronto has recently proposed an efficient online method for generating accurate memory reuse profiles.[13] This online approach would not have the same performance overhead as the direct measurement approach, because it would not require running each thread on each core type; a profile is collected while the thread is running on a single core and thus would not create the associated load imbalance.

Finally, there may be another promising approach for online estimation of relative speedup. Recall that our architectural signature method determines relative speedup based on the estimated LLC miss rates. Instead of estimating these miss rates from memory reuse profiles, we could also measure the miss rates online using hardware performance counters. Our preliminary results indicate this is a fruitful method for determining relative speedup online. We believe this method will be an effective way of estimating the relative speedup dynamically while avoiding the drawbacks of the direct measurement technique and the complexity associated with generation of memory reuse profiles.

## Conclusion

Asymmetric multicore systems promise to deliver higher performance per watt than conventional symmetric processors. To realize these benefits, however, the workload must have sufficient diversity (in terms of either parallelism or microarchitectural properties), and the operating-system scheduler must be designed to leverage this diversity on asymmetric hardware. Our experience with designing asymmetry-aware schedulers demonstrates that these algorithms can be feasibly implemented on real systems and used to unleash the potential of AMP systems without significant overhead. C

**Related articles on queue.acm.org**

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

**The Price of Performance**
*Luis André Barroso*
http://queue.acm.org/detail.cfm?id=1095420

**Modern System Power Management**
*Andrew Grover*
http://queue.acm.org/detail.cfm?id=957774

**References**


1. Annavaram, M., Grochowski, E., Shen, J. Mitigating Amdahl's law through EPI throttling. *ISCA*, 2005.
2. Becchi, M., Crowley, P. Dynamic thread assignment on heterogeneous multiprocessor architectures. In *Proceedings of Computing Frontiers* (2006).
3. Berg, E., Hagersten, E. StatCache: A probabilistic approach to efficient and accurate data locality analysis. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software* (2004).
4. Hill, M.D., Marty, M.R. Amdahl's law in the multicore era. *IEEE Computer 41*, 7, (2008) 33–38.
5. Kumar, R., Farkas, K.I., Jouppi, N., et al.. Single-ISA heterogeneous multicore architectures: the potential for processor power reduction. In *Proceedings of the International Symposium on Microarchitecture* (2003).
6. Kumar, R., et al. Single-ISA heterogeneous multicore architectures for multithreaded workload performance. In *Proceedings of the International Symposium of Computer Architecture* (2004).
7. Li, T., Baumberger, D., Koufaty, D.A., Hahn, S. Efficient operating system scheduling for performance-asymmetric multicore architectures. In *Proceedings of the Conference on Supercomputing* (2007).
8. Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., Wallace, S., Reddi, V.J., Hazelwood, K. Pin: building customized program analysis tools with dynamic instrumentation. In *Proceedings of Programming Language Design and Implementation* (2005).
9. Saez, J.C., Shelepov, D., Fedorova, A., Prieto, M. Leveraging workload diversity through OS scheduling to maximize performance on single-ISA heterogeneous multicore systems. Submitted to *Transactions on Parallel and Distributed Systems*, 2009.
10. Saez, J.C., Fedorova, A., Prieto, M., Vegas, H. Submitted to a conference for blind review. Details omitted to preserve anonymity.
11. Shelepov, D., Saez, J.C., Jeffery, S., Fedorova, A., Perez, N., Huang, Z.F., Blagodurov, S., Kumar, V. HASS: A scheduler for heterogeneous multicore systems. *ACM Operating System Review, 43*, 2 (2009), 66-75.
12. Suleman, M.A., Qureshi, M.K., Patt, Y.N. Feedback-driven threading: power-efficient and high-performance execution of multithreaded workloads on CMPs. *SIGARCH CAM 36*, 1 (2008): 277–286.
13. Tam, D. K., Azimi, R., Soares, L.B., Stumm, M. RapidMRC: Approximating L2 miss rate curves on commodity systems for online optimizations. In *Proceedings of Architectural Support for Programming Languages and Operating Systems* (2009).



**Alexandra Fedorova** is an assistant professor of computer science at Simon Fraser University, Vancouver, Canada, where she co-founded the SYNAR (Systems, Networking and Architecture) research lab. Her research focuses on multicore systems, with a particular emphasis on operating system scheduling for improving the efficiency of multicore systems.

**Juan Carlos Saez** is a Ph.D. student in computer science at the Complutense University, Madrid. His recent research focuses on operating-system scheduling on heterogeneous multicore processors, exploring new techniques to deliver better performance per watt, and quality of service on these systems. His work is supported by the Spanish government through an FPU fellowship grant.

**Daniel Shelepov** is a member of the Internet Explorer team at Microsoft, Redmond, WA.

**Manuel Prieto-Matias** is an associate professor in the department of computer architecture at Complutense University, Madrid. His current research addresses emerging issues related to chip multiprocessors, with a special emphasis on the interaction between the system software and the underlying architecture.



This research was supported by the National Science and Engineering Research Council of Canada through a Strategic Project Grant STPSC 356815-07, Sun Microsystems, and the Spanish government through Research Contract CICYT-TIN 2008/508 and Consolider Ingenio2010 2007/2011.

DOI:10.1145/1610252.1610271

**Easing the programmer's burden does not compromise system performance or increase the complexity of hardware implementation.**

BY JOSEP TORRELLAS, LUIS CEZE, JAMES TUCK, CALIN CASCAVAL, PABLO MONTESINOS, WONSUN AHN, AND MILOS PRVULOVIC

# The Bulk Multicore Architecture for Improved Programmability

MULTICORE CHIPS AS commodity architecture for platforms ranging from handhelds to supercomputers herald an era when parallel programming and computing will be the norm. While the computer science and engineering community has periodically focused on advancing the technology for parallel processing,[8] this time around the stakes are truly high, since there is no obvious route to higher performance other than through parallelism. However, for parallel computing to become widespread, breakthroughs are needed in all layers of the computing stack, including languages, programming models, compilation and runtime software, programming and debugging tools, and hardware architectures.

At the hardware-architecture layer, we need to change the way multicore architectures are designed. In the past, architectures were designed primarily for performance or for energy efficiency. Looking ahead, one of the top priorities must be for the architecture to enable a programmable environment. In practice, programmability is a notoriously difficult metric to define and measure. At the hardware-architecture level, programmability implies two things: First, the architecture is able to attain high efficiency while relieving the programmer from having to manage low-level tasks; second, the architecture helps minimize the chance of (parallel) programming errors.

In this article, we describe a novel, general-purpose multicore architecture—the Bulk Multicore—we designed to enable a highly programmable environment. In it, the programmer and runtime system are relieved of having to manage the sharing of data thanks to novel support for scalable hardware cache coherence. Moreover, to help minimize the chance of parallel-programming errors, the Bulk Multicore provides to the software high-performance sequential memory consistency and also introduces several novel hardware primitives. These primitives can be used to build a sophisticated program-development-and-debugging environment, including low-overhead data-race detection, deterministic replay of parallel programs, and high-speed disambiguation of sets of addresses. The primitives have an overhead low enough to always be "on" during production runs.

The key idea in the Bulk Multicore is twofold: First, the hardware automatically executes all software as a series of atomic blocks of thousands of dynamic instructions called Chunks. Chunk execution is invisible to the software and, therefore, puts no restriction on the programming language or model. Second, the Bulk Multicore introduces the use of Hardware Address Signatures as a low-overhead mechanism to ensure atomic and isolated execution of chunks and help

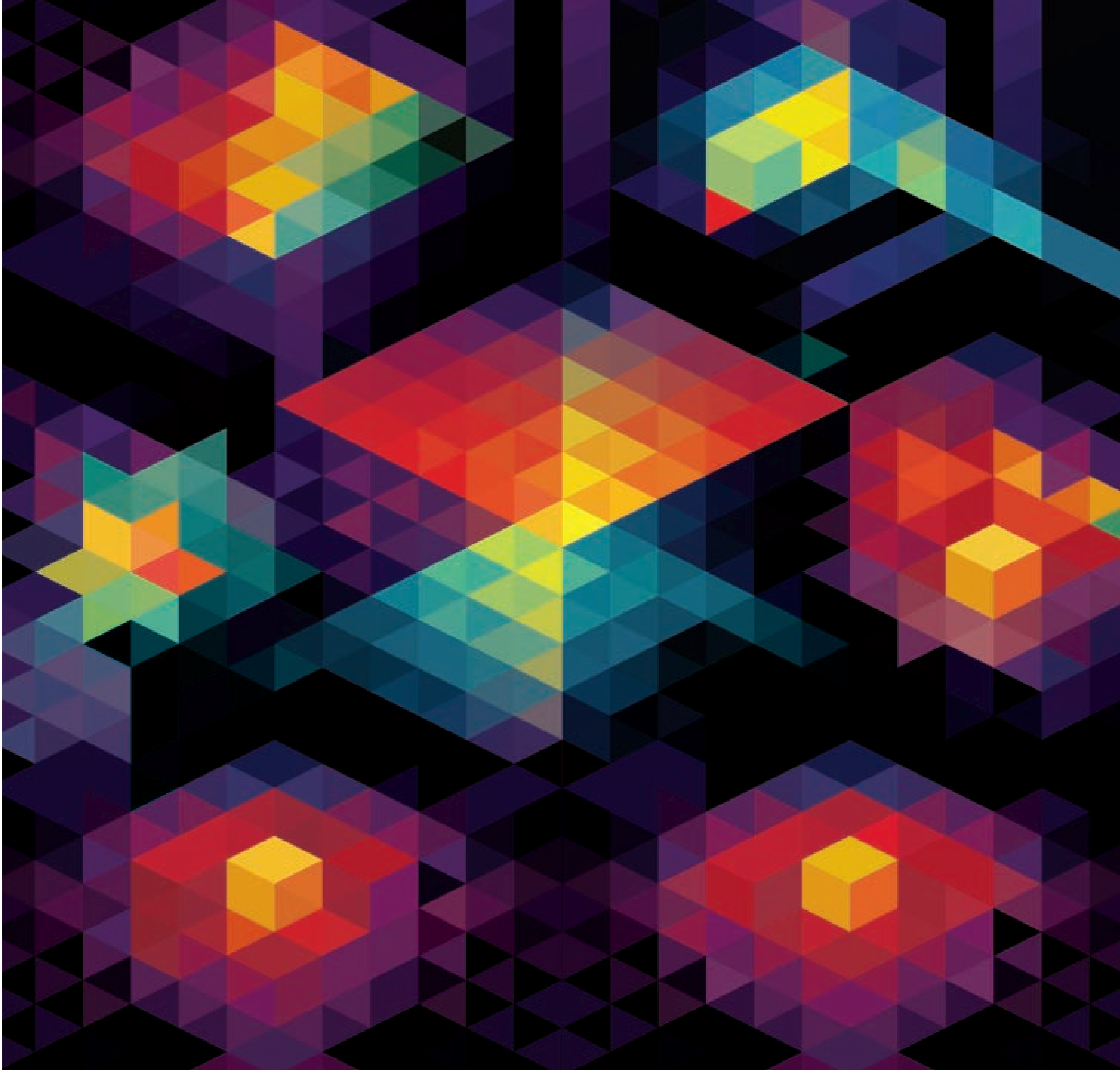


ILLUSTRATION BY ANDY GILMORE

maintain hardware cache coherence.

The programmability advantages of the Bulk Multicore do not come at the expense of performance. On the contrary, the Bulk Multicore enables high performance because the processor hardware is free to aggressively reorder and overlap the memory accesses of a program within chunks without risk of breaking their expected behavior in a multiprocessor environment. Moreover, in an advanced Bulk Multicore design where the compiler observes the chunks, the compiler can further improve performance by heavily optimizing the instructions within each chunk. Finally, the Bulk Multicore organization decreases hardware design complexity by freeing processor designers from having to worry about many corner cases that appear when designing multiprocessors.

## Architecture

The Bulk Multicore architecture eliminates one of the traditional tenets of processor architecture, namely the need to commit instructions in order, providing the architectural state of the processor after every single instruction. Having to provide such state in a multiprocessor environment—even if no other processor or unit in the machine needs it—contributes to the complexity of current system designs. This is because, in such an environment, memory-system accesses take many cycles, and multiple loads and stores from both the same and different processors overlap their execution.

In the Bulk Multicore, the default execution mode of a processor is to commit chunks of instructions at a time.[2] A chunk is a group of dynamically contiguous instructions (such as 2,000 instructions). Such a "chunked" mode of execution and commit is a hardware-only mechanism, invisible to the software running on the processor. Moreover, its purpose is not to parallelize a thread, since the chunks in a thread are not distributed to other processors. Rather, the purpose is to improve programmability and performance.

Each chunk executes on the processor atomically and in isolation. Atomic execution means that none of the chunk's actions are made visible to the rest of the system (processors or main memory) until the chunk completes and commits. Execution in isolation means that if the chunk reads a location and (before it commits) a second chunk in another processor that has written to the location commits, then the local chunk is squashed and must re-execute.

To execute chunks atomically and in isolation inexpensively, the Bulk Multicore introduces hardware address signatures.[3] A signature is a register of ≈1,024 bits that accumulates hash-encoded addresses. Figure 1 outlines a simple way to generate a signature (see the sidebar "Signatures and Signature Operations in Hardware" for a deeper discussion). A signature, therefore, represents a set of addresses.

In the Bulk Multicore, the hardware automatically accumulates the addresses read and written by a chunk into a read (R) and a write (W) signature, respectively. These signatures are kept in a module in the cache hierarchy. This module also includes simple functional units that operate on signatures, performing such operations as signature intersection (to find the addresses common to two signatures) and address membership test (to find out whether an address belongs to a signature), as detailed in the sidebar.

Atomic chunk execution is supported by buffering the state generated by the chunk in the L1 cache. No update is propagated outside the cache while the chunk is executing. When the chunk completes or when a dirty cache line with address in the W signature must be displaced from the cache, the hardware proceeds to commit the chunk. A successful commit involves sending the chunk's W signature to the subset of sharer processors indicated by the directory[2] and clearing the local R and W signatures. The latter operation erases any record of the updates made by the chunk, though the written lines remain dirty in the cache.

The W signature carries enough information to both invalidate stale lines from the other coherent caches (using the δ signature operation on W, as discussed in the sidebar) and enforce that all other processors execute their chunks in isolation. Specifically, to enforce that a processor executes a chunk in isolation when the processor receives an incoming signature $W_{inc}$, its hardware intersects $W_{inc}$ against the local $R_{loc}$ and $W_{loc}$ signatures. If any of the two intersections is not null, it means (conservatively) that the local chunk has accessed a data element written by the committing chunk. Consequently, the local chunk is squashed and then restarted.

Figure 2 outlines atomic and isolated execution. Thread *0* executes a chunk that writes variables *B* and *C*, and no invalidations are sent out. Signature $W_0$ receives the hashed addresses of *B* and *C*. At the same time, Thread *1* issues reads for *B* and *C*, which (by construction) load the non-

# Signatures and Signature Operations in Hardware

**Figure 1 in the main text shows a simple implementation of a signature. The bits of an incoming address go through a fixed permutation to reduce collisions and are then separated in bit-fields $C_i$. Each field is decoded and accumulated into a bit-field $V_j$ in the signature. Much more sophisticated implementations are also possible.**

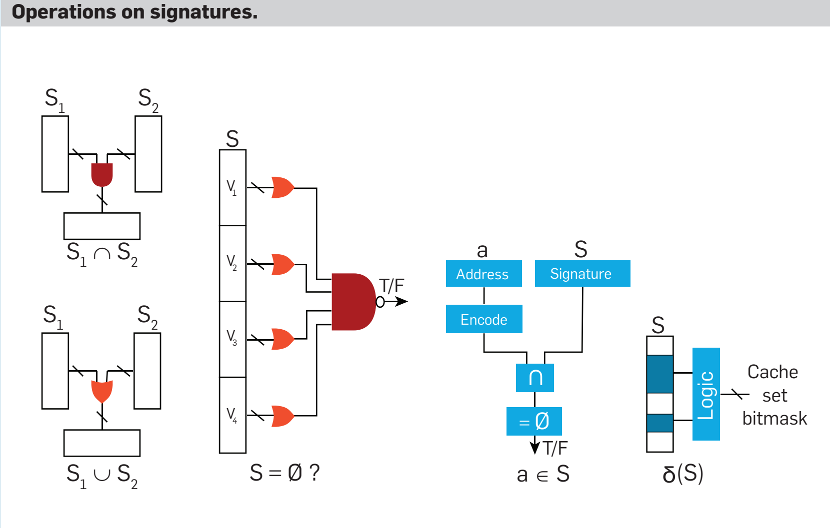
**A module called the Bulk Disambiguation Module contains several signature registers and simple functional units that operate efficiently on signatures. These functional units are invisible to the instruction-set architecture. Note that, given a signature, we can recover only a superset of the addresses originally encoded into the signature. Consequently, the operations on signatures produce conservative results.**

**The figure here outlines five signature functional units: intersection, union, test for null signature, test for address membership, and decoding (δ). Intersection finds the addresses common to two signatures by performing a bit-wise AND of the two signatures. The resulting signature is empty if, as shown in the figure, any of its bit-fields contains all zeros. Union finds all addresses present in at least one signature through a bit-wise OR of the two signatures. Testing whether an address *a* is present (conservatively) in a signature involves encoding *a* into a signature, intersecting the latter with the original signature and then testing the result for a null signature.**

**Decoding (δ) a signature determines which cache sets can contain addresses belonging to the signature. The set bitmask produced by this operation is then passed to a finite-state machine that successively reads individual lines from the sets in the bitmask and checks for membership to the signature. This process is used to identify and invalidate all the addresses in a signature that are present in the cache.**

**Overall, the support described here enables low-overhead operations on sets of addresses.[3]**

**Operations on signatures.**



speculative values of the variables—namely, the values before Thread *0*'s updates. When Thread *0*'s chunk commits, the hardware sends signature $W_0$ to Thread *1*, and $W_0$ and $R_0$ are cleared. At the processor where Thread *1* runs, the hardware intersects $W_0$ with the ongoing chunk's $R_1$ and $W_1$. Since $W_0 \cap R_1$ is not null, the chunk in Thread *1* is squashed.

The commit of chunks is serialized globally. In a bus-based machine, serialization is given by the order in which *W* signatures are placed on the bus. With a general interconnect, serialization is enforced by a (potentially distributed) arbiter module.[2] W signatures are sent to the arbiter, which quickly acknowledges whether the chunk can be considered committed.

Since chunks execute atomically and in isolation, commit in program order in each processor, and there is a global commit order of chunks, the Bulk Multicore supports sequential consistency (SC)[9] at the chunk level. As a consequence, the machine also supports SC at the instruction level. More important, it supports high-performance SC at low hardware complexity.

The performance of this SC implementation is high because (within a chunk) the Bulk Multicore allows memory access reordering and overlap and instruction optimization. As we discuss later, synchronization instructions induce no reordering constraint within a chunk.

Meanwhile, hardware-implementation complexity is low because memory-consistency enforcement is largely decoupled from processor structures. In a conventional processor that issues memory accesses out of order, supporting SC requires intrusive processor modifications. For example, from the time the processor executes a load to line L out of order until the load reaches its commit time, the hardware must check for writes to L by other processors—in case an inconsistent state was observed. Such checking typically requires sending, for each external coherence event, a signal up the cache hierarchy. The signal snoops the load queue to check for an address match. Additional modifications involve preventing cache displacements that could risk missing a coherence event. Consequently, load queues, L1 caches, and other critical processor components must be augmented with extra hardware.

**Figure 1. A simple way to generate a signature.**

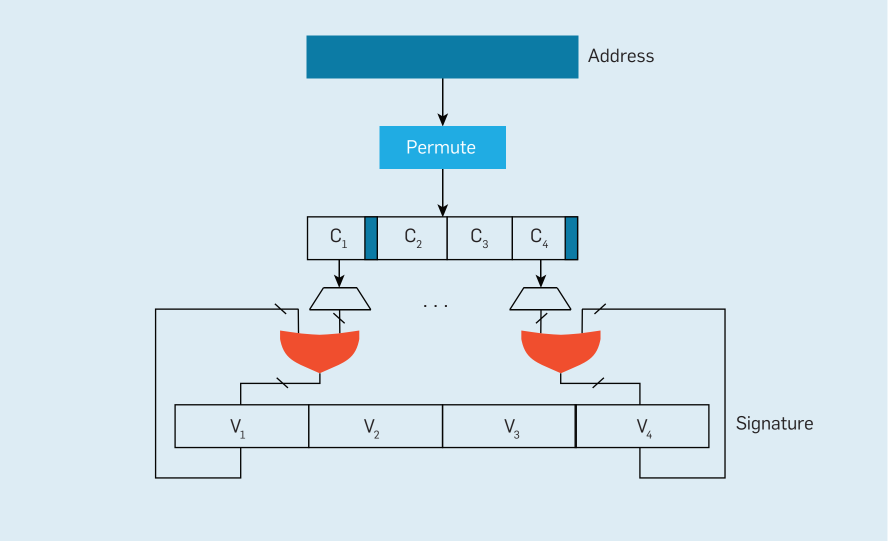
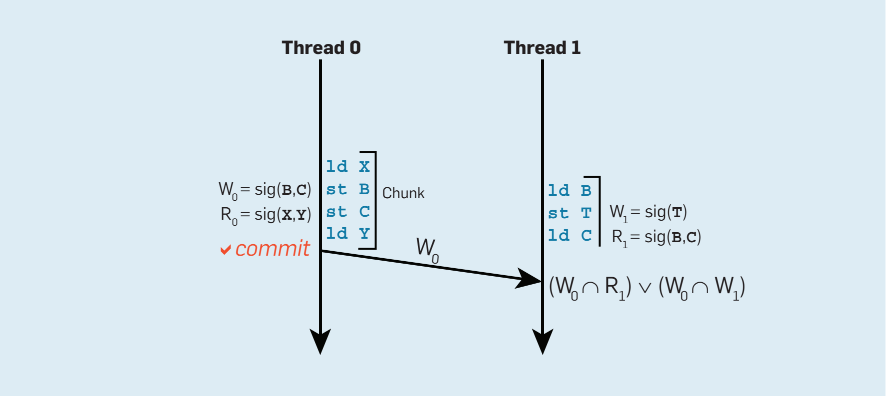



**Figure 2. Executing chunks atomically and in isolation with signatures.**




In the Bulk Multicore, SC enforcement and violation detection are performed with simple signature intersections outside the processor core. Additionally, caches are oblivious to what data is speculative, and their tag and data arrays are unmodified.

Finally, note that the Bulk Multicore's execution mode is not like transactional memory.[6] While one could intuitively view the Bulk Multicore as an environment with transactions occurring all the time, the key difference is that chunks are dynamic entities, rather than static, and invisible to the software.

## High Programmability

Since chunked execution is invisible to the software, it places no restriction on programming model, language, or runtime system. However, it does enable a highly programmable environment by virtue of providing two features: high-performance SC at the hardware level and several novel hardware primitives that can be used to build a sophisticated program-development-and-debugging environment.

Unlike current architectures, the Bulk Multicore supports high-performance SC at the hardware level. If we generate code for the Bulk Multicore using an SC compiler (such as the BulkCompiler[1]), we attain a high-performance, fully SC platform. The resulting platform is highly programmable for several reasons. The first is that debugging concurrent programs with data races would be much easier. This is because the possible outcomes of the memory accesses involved in the bug would be easier to reason about, and the debugger would in fact be able to reproduce the buggy interleaving. Second, most existing software correctness tools (such as Microsoft's CHESS[14]) assume SC. Verifying software correctness under SC is already difficult, and the state space balloons if non-SC interleavings need to be verified as well. In the next few years, we expect that correctness-verification tools will play a larger role as more parallel software is developed. Using them in combination with an SC platform would make them most effective.

A final reason for the programmability of an SC platform is that it would make the memory model of safe languages (such as Java) easier to understand and verify. The need to provide safety guarantees and enable performance at the same time has resulted in an increasingly complex and unintuitive memory model over the years. A high-performance SC memory model would trivially ensure Java's safety properties related to memory ordering, improving its security and usability.

The Bulk Multicore's second feature is a set of hardware primitives that can be used to engineer a sophisticated program-development-and-debugging environment that is always "on," even during production runs. The key insight is that chunks and signatures free development and debugging tools from having to record or be concerned with individual loads and stores. As a result, the amount of bookkeeping and state required by the tools is substantially reduced, as is the time overhead. Here, we give three examples of this benefit in the areas of deterministic replay of parallel programs, data-race detection, and high-speed disambiguation of sets of addresses.

Note, too, that chunks provide an excellent primitive for supporting popular atomic-section-based techniques for programmability (such as thread-level speculation[17] and transactional memory[6]).

*Deterministic replay of parallel programs with practically no log.* Hardware-assisted deterministic replay of parallel programs is a promising technique for debugging parallel programs. It involves a two-step process.[20] In the recording step, while the parallel program executes, special hardware records into a log the order of data dependences observed among the multiple threads. The log effectively captures the "interleaving" of the program's threads. Then, in the replay step, while the parallel program is re-executed, the system enforces the interleaving orders encoded in the log.

**The Bulk Multicore supports high-performance sequential memory consistency at low hardware complexity.**

In most proposals of deterministic replay schemes, the log stores individual data dependences between threads or groups of dependences bundled together. In the Bulk Multicore, the log must store only the total order of chunk commits, an approach we call DeLorean.[13] The logged information can be as minimalist as a list of committing-processor IDs, assuming the chunking is performed in a deterministic manner; therefore, the chunk sizes can be deterministically reproduced on replay. This design, which we call OrderOnly, reduces the log size by nearly an order of magnitude over previous proposals.

The Bulk Multicore can further reduce the log size if, during the recording step, the arbiter enforces a certain order of chunk commit interleaving among the different threads (such as by committing one chunk from each processor round robin). In this case of enforced chunk-commit order, the log practically disappears. During the replay step, the arbiter reinforces the original commit algorithm, forcing the same order of chunk commits as in the recording step. This design, which we call PicoLog, typically incurs a performance cost because it can force some processors to wait during recording.

Figure 3a outlines a parallel execution in which the boxes are chunks and the arrows are the observed cross-thread data dependences. Figure 3b shows a possible resulting execution log in OrderOnly, while Figure 3c shows the log in PicoLog.

*Data-race detection at production-run speed.* The Bulk Multicore can support an efficient data-race detector based on the "happens-before" method[10] if it cuts the chunks at synchronization points, rather than at arbitrary dynamic points. Synchronization points are easily recognized by hardware or software, since synchronization operations are executed by special instructions. This approach is described in ReEnact[16]; Figure 4 includes examples with a lock, flag, and barrier.

Each chunk is given a counter value called ChunkID following the happens-before ordering. Specifically, chunks in a given thread receive ChunkIDs that increase in program order. Moreover, a synchronization between two threads orders the ChunkIDs of the chunks involved in the synchronization. For example, in Figure 4a, the chunk in Thread *2* following the lock acquire (Chunk *5*) sets its ChunkID to be a successor of both the previous chunk in Thread *2* (Chunk *4*) and the chunk in Thread *1* that released the lock (Chunk *2*). For the other synchronization primitives, the algorithm is similar. For example, for the barrier in Figure 4c, each chunk immediately following the barrier is given a ChunkID that makes it a successor of all the chunks leading to the barrier.

Using ChunkIDs, we've given a partial ordering to the chunks. For example, in Figure 4a, Chunks *1* and *6* are ordered, but Chunks *3* and *4* are not. Such ordering helps detect data races that occur in a particular execution. Specifically, when two chunks from different threads are found to have a data-dependence at runtime, their two ChunkIDs are compared. If the ChunkIDs are ordered, this is not a data race because there is an intervening synchronization between the chunks. Otherwise, a data race has been found.

A simple way to determine when two chunks have a data-dependence is to use the Bulk Multicore signatures to tell when the data footprints of two chunks overlap. This operation, together with the comparison and maintenance of ChunkIDs, can be done with low overhead with hardware support. Consequently, the Bulk Multicore can detect data races without significantly slowing the program, making it ideal for debugging production runs.

*Enhancing programmability by making signatures visible to software.* Finally, a technique that improves programmability further is to make additional signatures visible to the software. This support enables inexpensive monitoring of memory accesses, as well as

# Making Signatures Visible to Software

**We propose that the software interact with some additional signatures through three main primitives:[18]**

**The first is to explicitly encode into a signature either one address (Figure 1a) or all addresses accessed in a code region (Figure 1b). The latter is enabled by the *bcollect* (begin collect) and *ecollect* (end collect) instructions, which can be set to collect only reads, only writes, or both.**

**The second primitive is to disambiguate the addresses accessed by the processor in a code region against a given signature. It is enabled by the *bdisamb.loc* (begin disambiguate local) and *edisamb.loc* (end disambiguate local) instructions (Figure 1c), and can disambiguate reads, writes, or both.**

**The third primitive is to disambiguate the addresses of incoming coherence messages (invalidations or downgrades) against a given local signature. It is enabled by the *bdisamb.rem* (begin disambiguate remote) and *edisamb.rem* (end disambiguate remote) instructions (Figure 1d) and can disambiguate reads, writes, or both. When disambiguation finds a match, the system can deliver an interrupt or set a bit.**

**Figure 2 includes three examples of what can be done with these primitives: Figure 2a shows how the machine inexpensively supports many watchpoints. The processor encodes into signature *Sig2* the address of variable *y* and all the addresses accessed in function *foo()*. It then watches all these addresses by executing *bdisamb.loc* on *Sig2*.**

**Figure 2b shows how a second call to a function that reads and writes memory in its body can be skipped. In the figure, the code calls function *foo()* twice with the same input value of *x*. To see if the second call can be skipped, the program first collects all addresses accessed by *foo()* in *Sig2*. It then disambiguates all subsequent accesses against *Sig2*. When execution reaches the second call to *foo()*, it can skip the call if two conditions hold: the first is that the disambiguation did not find a conflict; the second (not shown in the figure) is that the read and write footprints of the first *foo()* call do not overlap. This possible overlap is checked by separately collecting the addresses read in *foo()* and those written in *foo()* in separate signatures and intersecting the resulting signatures.**

**Finally, Figure 2c shows a way to detect data dependences between threads running on different processors. In the figure, *collect* encodes all addresses accessed in a code section into *Sig2*. Surrounding the collect instructions, the code places *disamb.rem* instructions to monitor if any remotely initiated coherence-action conflicts with addresses accessed locally. To disregard read-read conflicts, the programmer can collect the reads in a separate signature and perform remote disambiguation of only writes against that signature.**

**Figure 1. Primitives enabling software to interact with additional signatures: collection (a and b), local disambiguation (c), and remote disambiguation (d).**

(a)
```
Encode Addr, Sig1
```

(b)
```
bcollect Sig1
x = ...
... = y
ecollect Sig1
```

(c)
```
bdisamb.loc Sig1
x = ...
... = y
edisamb.loc Sig1
```

(d)
```
bdisamb.rem Sig1
x = ...
... = y
edisamb.rem Sig1
```

**Figure 2. Using signatures to support data watchpoints (a), skip execution of functions (b), and detect data dependencies between threads running on different processors (c).**

(a)
```
encode &y, Sig2
bcollect Sig2
foo()
ecollect Sig2
bdisamb.loc Sig2
  .
  .
```

(b)
```
foo(x)                bcollect Sig2
... = y               foo(x)
z = ...      →        ecollect Sig2
foo(x)                bdisamb.loc Sig2
                      ... = y
                      z = ...
                      edisamb.loc Sig2
                      if (conflict) {
                      foo(x)
                      }
```

(c)
```
bdisamb.rem Sig2
bcollect Sig2
... = y
z = ...
ecollect Sig2
edisamb.rem Sig2
```

novel compiler optimizations that require dynamic disambiguation of sets of addresses (see the sidebar "Making Signatures Visible to Software").

### Reduced Implementation Complexity

The Bulk Multicore also has advantages in performance and in hardware simplicity. It delivers high performance because the processor hardware can reorder and overlap all memory accesses within a chunk—except, of course, those that participate in single-thread dependences. In particular, in the Bulk Multicore, synchronization instructions do not constrain memory access reordering or overlap. Indeed, fences inside a chunk are transformed into null instructions. Fences' traditional functionality of delaying execution until certain references are performed is useless; by construction, no other processor observes the actual order of instruction execution within a chunk.

Moreover, a processor can concurrently execute multiple chunks from the same thread, and memory accesses from these chunks can also overlap. Each concurrently executing chunk in the processor has its own R and W signatures, and individual accesses update the corresponding chunk's signatures. As long as chunks within a processor commit in program order (if a chunk is squashed, its successors are also squashed), correctness is guaranteed. Such concurrent chunk execution in a processor hides the chunk-commit overhead.

Bulk Multicore performance increases further if the compiler generates the chunks, as in the BulkCompiler.[1] In this case, the compiler can aggressively optimize the code within each chunk, recognizing that no other processor sees intermediate states within a chunk.

Finally, the Bulk Multicore needs simpler processor hardware than current machines. As discussed earlier, much of the responsibility for memory-consistency enforcement is taken away from critical structures in the core (such as the load queue and L1 cache) and moved to the cache hierarchy where signatures detect violations of SC.[2] For example, this property could enable a new environment in which cores and accelerators are designed without concern for how to satisfy a particular set of access-ordering constraints. This ability allows hardware designers to focus on the novel aspects of their design, rather than on the interaction with the target machine's legacy memory-consistency model. It also motivates the development of commodity accelerators.

**Figure 3. Parallel execution in the Bulk Multicore (a), with a possible OrderOnly execution log (b) and PicoLog execution log (c).**

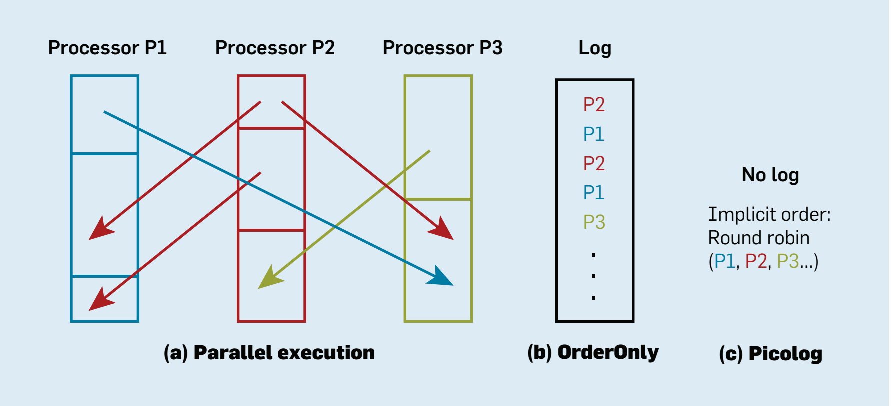
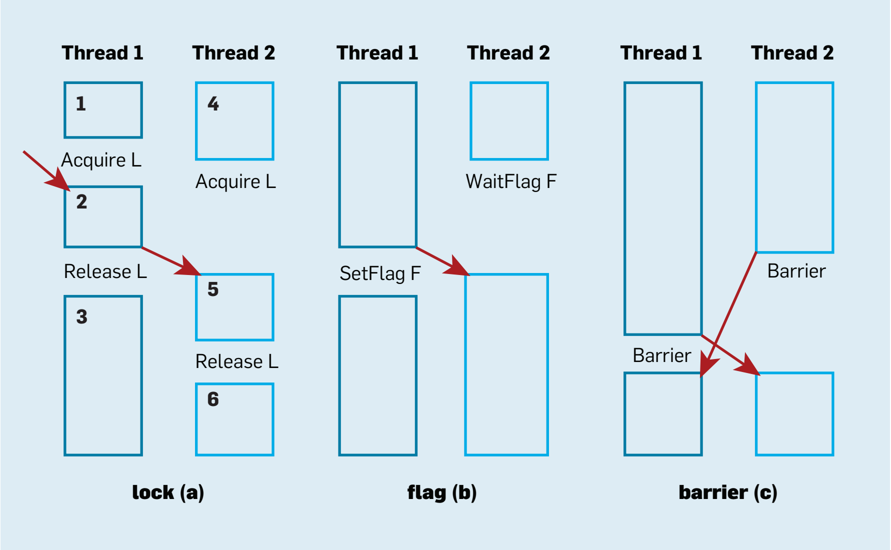



**Figure 4. Forming chunks for data-race detection in the presence of a lock (a), flag (b), and barrier (c).**




### Related Work

Numerous proposals for multiprocessor architecture designs focus on improving programmability. In particular, architectures for thread-level speculation (TLS)[17] and transactional memory (TM)[6] have received significant attention over the past 15 years. These techniques share key primitive mechanisms with the Bulk Multicore, notably speculative state buffering and undo and detection of cross-thread conflicts. However, they also have a different goal, namely simplify code parallelization by parallelizing the code transparently to the user software in TLS or by annotating the user code with constructs for mutual exclusion in TM. On the other hand, the Bulk Multicore aims to provide a broadly usable architectural platform that is easier to program for while delivering advantages in performance and hardware simplicity.

Two architecture proposals involve processors continuously executing blocks of instructions atomically and in isolation. One of them, called Transactional Memory Coherence and Consistency (TCC),[5] is a TM environment with transactions occurring all the time. TCC mainly differs from the Bulk Multicore in that its transactions are statically specified in the code, while chunks are created dynamically by the hardware. The second proposal, called Implicit Transactions,[19] is a multiprocessor environment with checkpointed processors that regularly take checkpoints. The instructions executed between checkpoints constitute the equivalent of a chunk. No detailed implementation of the scheme is presented.

Automatic Mutual Exclusion (AME)[7] is a programming model in which a program is written as a group of atomic fragments that serialize in some manner. As in TCC, atomic sections in AME are statically specified in the code, while the Bulk Multicore chunks are hardware-generated dynamic entities.

The signature hardware we've introduced here has been adapted for use in TM (such as in transaction-footprint collection and in address disambiguation[12,21]).

Several proposals implement data-race detection, deterministic replay of multiprocessor programs, and other debugging techniques discussed here without operating in chunks.[4,11,15,20] Comparing their operation to chunk operation is the subject of future work.

## Future Directions

The Bulk Multicore architecture is a novel approach to building shared-memory multiprocessors, where the whole execution operates in atomic chunks of instructions. This approach can enable significant improvements in the productivity of parallel programmers while imposing no restriction on the programming model or language used.

At the architecture level, we are examining the scalability of this organization. While chunk commit requires arbitration in a (potentially distributed) arbiter, the operation in chunks is inherently latency tolerant. At the programming level, we are examining how chunk operation enables efficient support for new program-development and debugging tools, aggressive autotuners and compilers, and even novel programming models.

## Acknowledgments

We would like to thank the many present and past members of the I-acoma group at the University of Illinois who contributed through many discussions, seminars, and brainstorming sessions. This work is supported by the U.S. National Science Foundation, Defense Advanced Research Projects Agency, and Department of Energy and by Intel and Microsoft under the Universal Parallel Computing Research Center, Sun Microsystems under the University of Illinois OpenSPARC Center of Excellence, and IBM.

**References**

1. Ahn, W., Qi, S., Lee, J.W., Nicolaides, M., Fang, X., Torrellas, J., Wong, D., and Midkiff, S. BulkCompiler: High-performance sequential consistency through cooperative compiler and hardware support. In *Proceedings of the International Symposium on Microarchitecture* (New York City, Dec. 12–16). IEEE Press, 2009.
2. Ceze, L., Tuck, J., Montesinos, P., and Torrellas, J. BulkSC: Bulk enforcement of sequential consistency. In *Proceedings of the International Symposium on Computer Architecture* (San Diego, CA, June 9–13). ACM Press, New York, 2007, 278–289.
3. Ceze, L., Tuck, J., Cascaval, C., and Torrellas, J. Bulk disambiguation of speculative threads in multiprocessors. In *Proceedings of the International Symposium on Computer Architecture* (Boston, MA, June 17–21). IEEE Press, 2006, 227–238.
4 Choi, J., Lee, K., Loginov, A., O'Callahan, R., Sarkar, V., and Sridharan, M. Efficient and precise datarace detection for multithreaded object-oriented programs. In *Proceedings of the Conference on Programming Language Design and Implementation* (Berlin, Germany, June 17-19). ACM Press, New York, 2002, 258–269.
5 Hammond, L., Wong, V., Chen, M., Carlstrom, B.D., Davis, J.D., Hertzberg, B., Prabhu, M.K., Wijaya, H., Kozyrakis, C., and Olukotun, K. Transactional memory coherence and consistency. In *Proceedings of the International Symposium on Computer Architecture* (München, Germany, June 19–23). IEEE Press, 2004, 102–113.
6. Herlihy M. and Moss, J.E.B. Transactional memory: Architectural support for lock-free data structures. In *Proceedings of the International Symposium on Computer Architecture* (San Diego, CA, May 16–19). IEEE Press, 1993, 289–300.
7 Isard, M. and Birrell, A. Automatic mutual exclusion. In *Proceedings of the Workshop on Hot Topics in Operating Systems* (San Diego, CA, May 7–9). USENIX, 2007.
8. Kuck, D. Facing up to software's greatest challenge: Practical parallel processing. *Computers in Physics 11*, 3 (1997).
9. Lamport, L. How to make a multiprocessor computer that correctly executes multiprocess programs. *IEEE Transactions on Computers C-28*, 9 (Sept. 1979), 690–691.
10. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM 21*, 7 (July 1978), 558–565.
11. Lu, S., Tucek, J., Qin, F., and Zhou, Y. AVIO: Detecting atomicity violations via access interleaving invariants. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems* (San Jose, CA, Oct. 21–25). ACM Press, New York, 2006, 37–48.
12. Minh, C., Trautmann, M., Chung, J., McDonald, A., Bronson, N., Casper, J., Kozyrakis, C., and Olukotun, K. An effective hybrid transactional memory with strong isolation guarantees. In *Proceedings of the International Symposium on Computer Architecture* (San Diego, CA, June 9–13). ACM Press, New York, 2007, 69–80.
13. Montesinos, P., Ceze, L., and Torrellas, J. DeLorean: Recording and deterministically replaying shared-memory multiprocessor execution efficiently. In *Proceedings of the International Symposium on Computer Architecture* (Beijing, June 21–25). IEEE Press, 2008, 289–300.
14. Musuvathi, M. and Qadeer, S. Iterative context bounding for systematic testing of multithreaded programs. In *Proceedings of the Conference on Programming Language Design and Implementation* (San Diego, CA, June 10–13). ACM Press, New York, 2007, 446–455.
15. Narayanasamy, S., Pereira, C., and Calder, B. Recording shared memory dependencies using strata. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems* (San Jose, CA, Oct. 21–25). ACM Press, New York, 2006, 229–240.
16. Prvulovic, M. and Torrellas, J. ReEnact: Using thread-level speculation mechanisms to debug data races in multithreaded codes. In *Proceedings of the International Symposium on Computer Architecture* (San Diego, CA, June 9–11). IEEE Press, 2003, 110–121.
17. Sohi, G., Breach, S., and Vijaykumar, T. Multiscalar processors. In *Proceedings of the International Symposium on Computer Architecture* (Santa Margherita Ligure, Italy, June 22–24). ACM Press, New York, 1995, 414–425.
18. Tuck, J., Ahn, W., Ceze, L., and Torrellas, J. SoftSig: Software-exposed hardware signatures for code analysis and optimization. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems* (Seattle, WA, Mar. 1–5). ACM Press, New York, 2008, 145–156.
19. Vallejo, E., Galluzzi, M., Cristal, A., Vallejo, F., Beivide, R., Stenstrom, P., Smith, J.E., and Valero, M. Implementing kilo-instruction multiprocessors. In *Proceedings of the International Conference on Pervasive Services* (Santorini, Greece, July 11–14). IEEE Press, 2005, 325–336.
20. Xu, M., Bodik, R., and Hill, M.D. A 'flight data recorder' for enabling full-system multiprocessor deterministic replay. In *Proceedings of the International Symposium on Computer Architecture* (San Diego, CA, June 9–11). IEEE Press, 2003, 122–133.
21. Yen, L., Bobba, J., Marty, M., Moore, K., Volos, H., Hill, M., Swift, M., and Wood, D. LogTM-SE: Decoupling hardware transactional memory from caches. In *Proceedings of the International Symposium on High Performance Computer Architecture* (Phoenix, AZ, Feb. 10–14). IEEE Press, 2007, 261–272.

**Josep Torrellas** (torrellas@cs.uiuc.edu) is a professor and Willett Faculty Scholar in the Department of Computer Science at the University of Illinois at Urbana-Champaign.

**Luis Ceze** (luisceze@cs.washington.edu) is an assistant professor in the Department of Computer Science and Engineering at the University of Washington, Seattle, WA.

**James Tuck** (jtuck@ncsu.edu) is an assistant professor in the Department of Electrical and Computer Engineering at North Carolina State University, Raleigh, NC.

**Calin Cascaval** (cascaval@us.ibm.com) is a research staff member and manager of programming models and tools for scalable systems at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

**Pablo Montesinos** (pmontesi@samsung.com) is a staff engineer in the Multicore Research Group at Samsung Information Systems America, San Jose, CA.

**Wonsun Ahn** (dahn2@uiuc.edu) is a graduate student in the Department of Computer Science at the University of Illinois at Urbana-Champaign.

**Milos Prvulovic** (milos@cc.gatech.edu) is an associate professor in the School of Computer Science, College of Computing, Georgia Institute of Technology, Atlanta, GA.

DOI:10.1145/1610252.1610272

**University CS departments are incorporating game design and development to prepare their students for the game industry's expectations.**

BY MICHAEL ZYDA

# Computer Science in the Conceptual Age

IN TECHNOLOGY, THE conceptual age is defined by cognitive or creative assets, including design, storytelling, artistry, empathy, play, and emotion. Good engineering or good computer science is no longer enough; design must be just as good.[7] The transition from information age to conceptual age has been overlooked by most academics in computer science, yet many of the consequences of the transition have been apparent for the past decade at least, including a drop in undergraduate interest, the outsourcing of U.S. programming jobs, and the decline in research laboratories focused on advanced computer science research. Today, another aspect of the transition is emerging: integration of game development into computer science curricula. Here, I discuss what it looks like, how it affects computer science departments, and how it helps drive the overall transition.

The last quarter of the 20th century saw the U.S. economy transition from the industrial age to the information age, a move characterized by major changes in profession, as many people chose to be knowledge workers, rather than manual laborers. Computer science was the driving intellectual, social, and cultural force behind the information age. Computer researchers and developers spearheaded widespread adoption of new technology and were paid well. Academic computer science departments grew in terms of numbers of students, faculty, and facilities; almost all major universities received donations, including new "computer science buildings," and enrollment was so large that standardized tests had to be created for the field. These were the fattest of times, when computer science research drove the growth in the national, as well as the global, economy.

The dot-com crash closing out the 20th century signaled the end of the era. Computer science was now viewed as an unstable career choice, with corresponding drop in interest by young people.[8] Additionally, many computer science departments heard complaints from industry that their recent graduates were unprepared to be part of the modern work force. So, following their economic self-interest, students migrated to other fields, and the leading U.S. universities saw major growth in undergraduate business programs.

We can hypothesize that the drop in interest in computer science was part of the natural ebb and flow and fashion of career choice. But doing so is to stick one's head in the sand and hope for the return of yesteryear. A more progressive view is to focus on the concurrent transition from information age to conceptual age. The old

**Scene from the *Artemis Chronicle* PC game built with Microsoft's XNA toolkit and the USC GamePipe Laboratory NitroX game engine.**

ILLUSTRATION BY ANDREA TSENG / CHIMERA GAME STUDIOS

focus on "tractable abstractions separate from real problems"[2] is no longer acceptable. In the conceptual age, we need to change the direction of the field or continue an unwelcome slide toward irrelevance.

## Cognitive and Creative Assets

The conceptual age of technology is defined by cognitive and creative assets, with the design side being just as good. We see harbingers in recent business success. For example, Apple has done well with design-driven products enabled by great engineering. On the other hand, Microsoft is a great engineering company that deemphasizes design at its own peril; the Vista operating system, despite its technical success, was not built with user experience as its ultimate goal and had great difficulty with respect to adoption. Much of the rest of the traditional computing industry is shrinking, but the game industry is a segment that continues to grow due to its focus on design backed up by great supporting engineering. The demand for computer scientists capable of building games is high, with large companies like Electronic Arts reporting that 65% of their hiring demand is for programmers skilled in building games. Unfortunately, the kind of computer scientist required by the game industry is not exactly what traditional computer science departments produce.

The game industry wants graduates who are strong programmers and system developers, skilled in game design, and capable of and experienced in game development in large, cross-disciplinary collaborative teams. Some computer science departments produce graduates who are strong programmers and system developers, but most do not produce graduates skilled in game design. Moreover, most do not produce graduates facile in large, cross-disciplinary collaborative teams. The typical computer science graduate has little experience with team software development beyond one or two projects with three to five other computer science students.

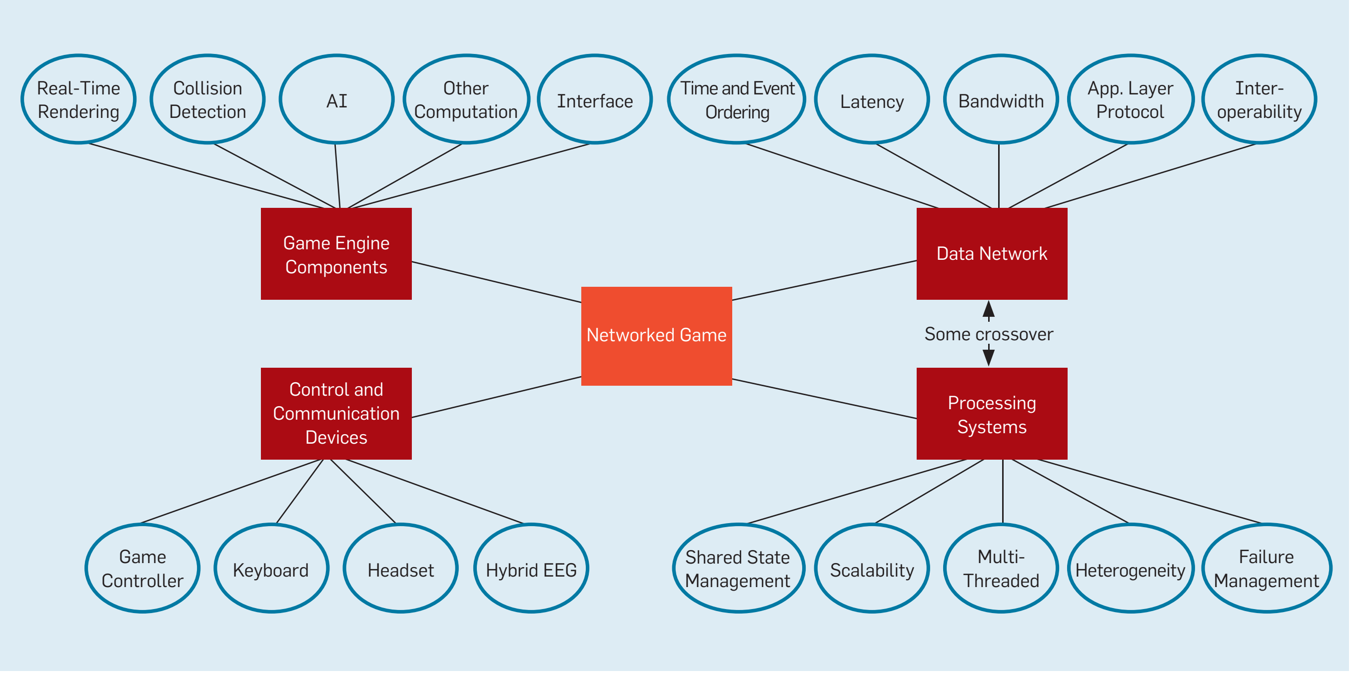
Computer science departments can retool themselves to meet these challenges, but, for game development, doing so requires a strong, experienced champion and proper resources. Here, I discuss a particular approach we take at the University of Southern California, outcomes from that program, and questions with respect to transitioning a mature field toward the conceptual age.

## Game Development and CS Education

If computer science departments are indeed to make the transition, what will they transition into? The game industry also wants graduates with a strong background in computer science. It does not want graduates with watered-down computer science degrees, but rather an enhanced set of skills. This is good news for the departments, meaning they can transition some courses to new material or new foci, create new courses, and still not abandon decades of hard-won knowledge. The best way to think about the transition to the conceptual age is to make the focus the "big idea or big concept," with a follow-on focus on how to build the concept and with what technologies.

Strong programming skills is the first item on the list, meaning an undergraduate program's first four computer science courses—CS-101 Fundamentals of Computer Programming, CS-102 Data Structures, CS-105 Object Oriented Programming, and CS-201 Principles of Software Development—must be taught in a rigorous manner by excellent practicing programmers. While computer programming languages abound, and the historic computer science attitude "You learn one language, and it's easy to pick up the next one" is not shared by industry. The game industry will also tell you that it wants the first four programming classes in C++, not Java, according to M.M. McGill[4] and

Components of a networked game.



my own private communications with directors of human resources in major game-development companies. Many universities switched to Java with the Internet boom, as it is easy to resource a programming laboratory for Java, and Java support for the user interface is extensive. But the game industry programs mainly in C++. Many game companies say they will not interview or hire someone whose first programming language is Java. Computer scientists might argue about the virtues of programming languages, but most console and PC games are built in C++. From a C++ perspective, our students can teach themselves other useful languages, learning Objective-C for iPhone game development or C# for games with XNA in a single semester. The industry is basically saying it wants strong programmers with multiple courses in C++, its primary development language. The USC experience is that 100% of its students interviewed for programming positions are given three-to-four-hour-long programming tests, with almost all companies administering the tests in C++. We had one company (NCSoft) test in C and another (Microsoft) indicate the test could be done in C++, C, or C#. Programming interns/job seekers from our program Spring 2009 (35 interviewed in the game industry) found no companies administering programming tests in Java.

Strong skills in system development is another must. The figure here outlines the components that must be juggled by a programmer developing a typical networked game, touching on much of computer science and then some. Moreover, the game software must run multithreaded in a well-balanced manner and provide an immersive and entertaining experience to the game player. Game development is viewed by some as systems design.

Programmers comfortable in cross-disciplinary groups is third on the list. Industry wants programmers who are able to generate software based on the vision of designers, work with artists to generate the right display and feel, and know how to participate in large-team development efforts. The traditional computer science student is far from this.

**Programming interns/job seekers from our program Spring 2009 (35 interviewed in the game industry) found no companies administering programming tests in Java.**

Computer science departments must provide the necessary experience so game programmers are industry-ready on graduation day. This means that graduates must have built a significant game by collaborating with other students, not all of whom have backgrounds in computer science. Grads should also be familiar with the pipeline-development process and asset-management systems, both aspects of game-industry development potentially useful throughout the program. Such experience is very different from traditional computer science-degree programs in which software-development teams are small, and there is no strong requirement for asset management or source-code base sharing or versioning.

### Big Game Program or Baby Steps?

Computer science departments changing their focus toward game development is an obvious approach toward preparing students for the conceptual age. Many universities are building game-development programs within or aligned with their computer science departments.[1,3,5,6,9,10] The annual Foundations of Digital Games conference (http://www.foundationsofdigitalgames.org/) focuses on this important transition.

How does all this directly affect the departments? Consider two separate efforts: the USC GamePipe Laboratory and the University of Washington, Bothell, the latter covered in the article "Computer Games and Computer Science" by Kelvin Sung (on page 74).

### USC GamePipe Laboratory

The USC Department of Computer Science offers both a bachelor's degree in computer science (games) and a master's degree in computer science (game development).[12] Students interested in Ph.D.-level topics are encouraged (for the moment) to apply to the traditional computer science Ph.D. program. The bachelor's in computer science (games) program includes 37 units of traditional computer science and 42 units of game-development courses. The computer science courses are the same as those taken by regular students in USC's bachelor's computer-science program, except for the following modifications:

*Programming courses.* We rewrote

three of the first four programming courses to be in C++ and created game-oriented examples and exercises. The game focus helps motivate students, getting them excited about programming. For Fall 2008, results showed a 28% increase in the number of students with letter grade A in the games-oriented CS-101 Introduction to the Fundamentals of Programming and for Spring 2009 a 49% increase (http://gamepipe.usc.edu/~zyda/GamePipe/Ghyam-Final-MS-Study-2009.pdf). Further analysis of the results is underway to determine whether they reflect superior skills, superior understanding of programming, superior motivation, or some other cause;

*Replaced EE with CS.* We replaced four electrical-engineering courses on circuit design with computer-science-focused EE-352 Computer Organization & Architecture and parallel-programming-focused EE-452 Game Hardware Architecture. The removed courses represent an older style of computer-science-degree program. We felt it more important that students learn how computer architectures affect programming rather than how to make such architectures. We also felt that parallel programming was highly relevant to both the multithreaded nature of modern game development and multicore processors;

*Added EE.* We added EE-450 Introduction to Computer Networks as a degree requirement. Networking is offered as an elective in the regular computer science curriculum. Because games need networking, all our students take it;

*Cut compiler courses.* We eliminated the two compiler courses taken by regular computer science students. ACM eliminated compilers as a requirement from the CS core in 1979. The USC Computer Science Department uses the two courses as large programming-project capstone courses, so we felt we would rather have our students build games. The replacement for the games curriculum is the two-semester CS-491A/B Final Game Projects course. An interesting result is that the Computer Science Department is weighing whether to allow general computer science students to take the Final Game Projects course instead of the compiler sequence; and

**We felt that parallel programming was highly relevant to the multithreaded nature of modern game development and multicore processors.**

*No foreign languages.* General education requirements for the bachelor's degree in computer science (games) are approximately the same as for USC's regular computer science degree. The degree lacks a four-course foreign-language requirement as in all other USC College degrees, an accident of planning rather than a recommendation.

When we began planning the degree, the dean of engineering said, "I don't want a weak degree." So we made these changes to the computer science component of the program and are confident we offer an academically strong and industry-viable undergraduate degree.

For the games-side of the bachelor's degree, we replaced 42 units of electives from the general computer science degree with a full course in game development. Hence, the degree looks more like a double major than a specialization. We also have a set of courses on game engineering, game design, and game cross-disciplinary.

Game engineering covers video-game programming, parallel programming on consoles and graphics processing units, and programming game engines, all of which are straightforward software-development courses and all important for game development.

For game design, we send our students to a three-course Game Design Workshop sequence in the USC School of Cinematic Arts Interactive Media Program. Engineers are immersed in gameplay design as taught by master game designers. The first course in the sequence—CTIN-488 Game Design Workshop—teaches students how to prototype gameplay using cardboard and hand-drawn art; they basically build board games. We get an interesting response from the engineers we send there. Some rave about it, saying it's the best thing they've ever taken. Some hate it, feeling frustrated they cannot immediately code-up a game. Some hate the first few weeks but in the end come back and say it was a great course. Ultimately, the students who understand the importance of the course and express satisfaction end up with great internships/jobs in the game industry. It's where we see the future game industry technical directors of the conceptual age.

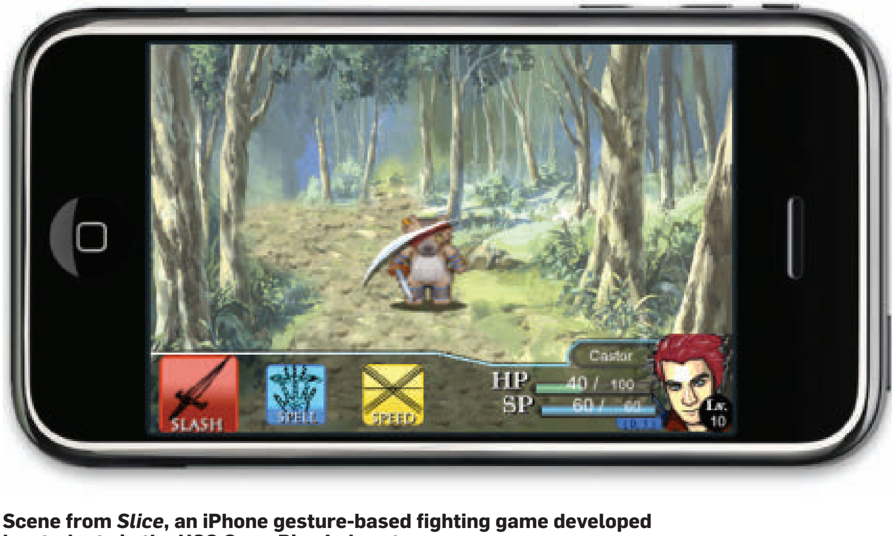
Our approach to creating a cross-disciplinary program was to design courses that could be taken by non-computer scientists, as well as by computer science majors. First-semester undergraduates survey game play, from the start of games to using (among other things) old consoles, old PC games, and emulators. Students come out of the course hooked on our degree program and wanting more. They then take a video-game production course to build individual games using GameMaker and hear from industry speakers on game development. Next is CS-281 Pipelines for Games & Interactives in which they learn how to create assets for games, including 3D models and animations. One achievement is a pipeline asset and source-code management system we designed that is shared by all game-development courses in the program. We teach our students how to use it early on, simplifying and enhancing their ability to develop games in subsequent courses. We also place all our students in a semester-long character-animation course.

In their final one-and-a-half years before graduation, our students become ready for game development. All take a course developing a serious game in a large team for an interested sponsor. Their last year before graduation is in CS-491A/B Final Game Projects building games in large cross-disciplinary teams from August to May, with game designs selected through a design playoff the previous May. The Final Game Projects course is administered jointly by the School of Cinematic Arts and has students from computer science (bachelor's and master's students), interactive media (bachelor's of fine arts and master's of fine arts), fine arts (bachelor's of fine arts), animation (bachelor's of fine arts), music composition, and film scoring. Teams in this class include from 11 to 25 students building significant games over its two-semester run, aiming for contest submission by the end of the second semester.

Strong cross-disciplinary collaboration occurs, with results presented at the end of each semester on Demo Day when game-industry executives are invited to review the students' work; the accompanying screenshots are indicative of the visual quality. *Artemis Chronicle*, a beautiful action-adventure title (see page 67), demonstrates the powerful features of the NitroX Engine, a revolutionary, complete development framework for creating XNA games. Both the game and the NitroX engine were built in the CS-491AB Final Games course over two semesters. *Air Guitar God*, a beat-matching iPhone game (below), incorporates a student-designed algorithm for automatically computing beat detection from any song imported into the game. And *Slice*, an action role-playing game (below), puts battles at your fingertips on the iPhone. Videos of the most recent Demo Day are at http://gamepipe.usc.edu/USC_GamePipe_Laboratory/R%26D/R%26D.html.

At the end of each academic year, we now routinely place large numbers of students (typically around 35) in internships/jobs in the game industry where they are nearly instantly productive. In the Fall semesters in 2007 and 2008, a team on Demo Day was offered a commercial deal to turn their game into a company for further development or prepare to ship commercially. Spring 2009 included five student-built games under commercial consideration.


**Opening screen from *Air Guitar God*, an iPhone beat-matching game developed by students in the USC GamePipe Laboratory's mobile games course.**




**Scene from *Slice*, an iPhone gesture-based fighting game developed by students in the USC GamePipe Laboratory.**


Industry wants programmers comfortable in cross-disciplinary teams.

## Effect on the USC CS Department

How has the game-degree program affected USC's Computer Science Department, nudging it along toward the conceptual age? First, I must say we have not dismantled the traditional bachelor's program in computer science, and students continue to enroll in it. Computer science and games students share almost the same core computer science curriculum.

Enrollment concerns were part of our original motivation with respect to the games program, though they have eased, as they have for many departments, according to the Computing Research Association' annual Taulbee Survey (http://www.cra.org/statistics/). For the Fall 2009 semester, 29% of the students in the USC bachelor's in computer science program are in the games program, representing an important influence on the department.

While we have not measured it, probably the greatest effect we see is an apparent "joy of computing" feeling by our students who come to class highly motivated, pour their best ideas into their projects, and produce spectacularly creative results, some of which the game industry wants to commercialize. The graduating students who move into positions in the game industry return to subsequent Demo Days, bring their bosses, and hire more of our graduates. The addition of a creative-design component and making it student-driven and student-dependent is key to this success. Students take ownership of their educational program and aspire to make everything they do shine.

Another component worth mentioning is the commitment of the program's faculty and instructors. The students are in small classes where they have much say in direction and result. The faculty is available to provide technical guidance, mentorship, motivation, and support. It is physically draining if done properly, but if the students recognize their passion for games, they cannot help but be passionate game developers as well. Faculty working directly with our program are the executive producers of some 12 to 14 games per semester. The results of these efforts are visible on the USC GamePipe Web site (http://gamepipe.usc.edu).

We have also been able to create a line of research funding based on and around games.[11] It is more than difficult to build an R&D program on games without a pipeline of students learning to build them. Today, we have both. The games program has additionally strengthened the reputation of USC's Computer Science Department in terms of increased R&D funding, improved hiring rates for graduates, attention to the games program in the press, invitations to games faculty to speak at conferences and publish in traditional venues, and other universities desiring to copy USC's success. In Fall 2008, the director of the USC GamePipe Laboratory was appointed an ACM Distinguished Speaker (http://www.dsp.acm.org/) in recognition of the program's achievement. Hiring managers and developers in the games industry now regard us as one of the "best games programs," though we are only in our fourth year of operation. Moreover, traditional USC computer science faculty not currently involved in the games program have begun to ask how they can participate, some have changed their course projects to be game-related, and many ask how we can draft proposals together. The computer science faculty realizes that something important is happening, with some beginning to also guide their own programs toward the conceptual age. C

**References**

1. Barnes, T., Powell, E., Chaffin, A., and Lipford, H. Game2Learn: improving the motivation of CS1 students. ACM Game Development in Computer Science Education (Miami, FL, Feb. 26–Mar. 3, 2008). 1–5.
2. Brooks, Jr., F.P. The computer scientist as toolsmith II. *Commun. ACM 39*, 3 (Mar. 1996), 61–68.
3. Horswill, I. and Novak, M. Evolving the artist-technologist. *IEEE Computer 39*, 6 (June 2006), 62–69.
4. McGill, M.M. Defining the expectation gap: A comparison of industry needs and existing game development curricula. ACM Foundations of Digital Games (Orlando, FL, 2009), 129–136.
5. Parberry, I., Roden, T., and Kazemzadeh, M.B. Experience with an industry-driven capstone course on game programming. ACM SIG on Computer Science Education (Houston, TX, 2005), 91–95.
6. Phelps, A., Egert, C., and Bierre, K. Games first pedagogy: Using games and virtual worlds to enhance programming education. *Journal of Game Development 1*, 4 (May 2006), 45–64.
7. Pink, D.H. *A Whole New Mind: Moving from the Information Age to the Conceptual Age.* Riverhead Books, New York, 2005.
8. Thibodeau, P. Computer science graduating class of 2007 smallest this decade. *Computerworld Online* (Mar. 5, 2008); http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9066659.
9. Whitehead, J. Introduction to game design in the large classroom. ACM Game Development in Computer Science Education (Miami, FL, 2008), 61–65.
10. Wolz, U. and Pulimood, S.M. An integrated approach to project management through classic CS III and video game development. ACM SIG on Computer Science Education (Covington, KY, 2007), 322–326.
11. Zyda, M., Spraragen, M., and Ranganathan, B. Testing behavioral models with an online game. *IEEE Computer 42*, 4 (Apr. 2009), 103–105.
12. Zyda, M., Lacour, V., and Swain, C. Operating a computer science game degree program. ACM Game Development in Computer Science Education (Miami, FL, 2008), 71–75.

**Michael Zyda** (zyda@usc.edu) is the director of the USC GamePipe Laboratory and a professor of engineering practice in the Department of Computer Science at the University of Southern California, Los Angeles, CA.



PHOTOGRAPH BY MICHAEL ZYDA/USC GAMEPIPE LABORATORY

DOI:10.1145/1610252.1610273

**Integrating computer games into existing CS courses may help attract students to the field, but there are guidelines to be considered.**

BY KELVIN SUNG

# Computer Games and Traditional CS Courses

SINCE COMPUTING IS the foundation of modern society, a proficient computing work force is essential for maintaining the country's leadership and competitiveness in the global economy. The recent decline in enrollments across computer science (CS) departments and the decrease in student diversity pose significant challenges to the continuation of the nation's prominent position in the global high-technology arena. The CS education community responded to this challenge with a general critical self-reexamination where the entire traditional CS education system is being evaluated, from the outreach to K–12 education, to the fundamental philosophies behind the curriculum design. One of the emerging results from these developments is the push for presenting abstract CS concepts in the context of familiar real-world applications.

Relating abstract principles to real-world experience has become increasingly prominent in mathematics and general science education. For example, the *Calculus Reform* movement of the 1990s included both pedagogical changes and foci on real-world problems, while the Carl Wieman Science Education Initiative at the University of British Columbia has redesigned its freshmen introductory physics course such that:[a]

*"As much as possible, the standard introductory physics material will be presented in connection with real-world situations and issues such as home heating, transportation, and electricity generation."*

In the CS education arena, the Media Computation of Georgia Tech[18] is an excellent example where foundational programming concepts are presented in the context of popular digital multimedia applications. This contextualization of computing education[18] is an ongoing effort and interactive computer video games, being one of the most familiar application areas for our students, is a context favored by many CS educators.

This article presents the USC GamePipe Laboratory effort where the entire CS curriculum is redesigned in the context of game development (Please refer to the USC GamePipe Laboratory effort by Michael Zyda on page XXX where the CS curriculum is designed in the context of game development). This article examines the ongoing efforts to integrate computer video games in existing traditional CS courses. The discussion is divided into introductory programming courses and elective CS courses, and concludes with guidelines for considering integrating computer game content into existing CS classes.

## Games and CS Classes

There are many types of games that

a http://www.cwsei.ubc.ca/departments/physics-astro courses.htm (Nov. 2007 update).

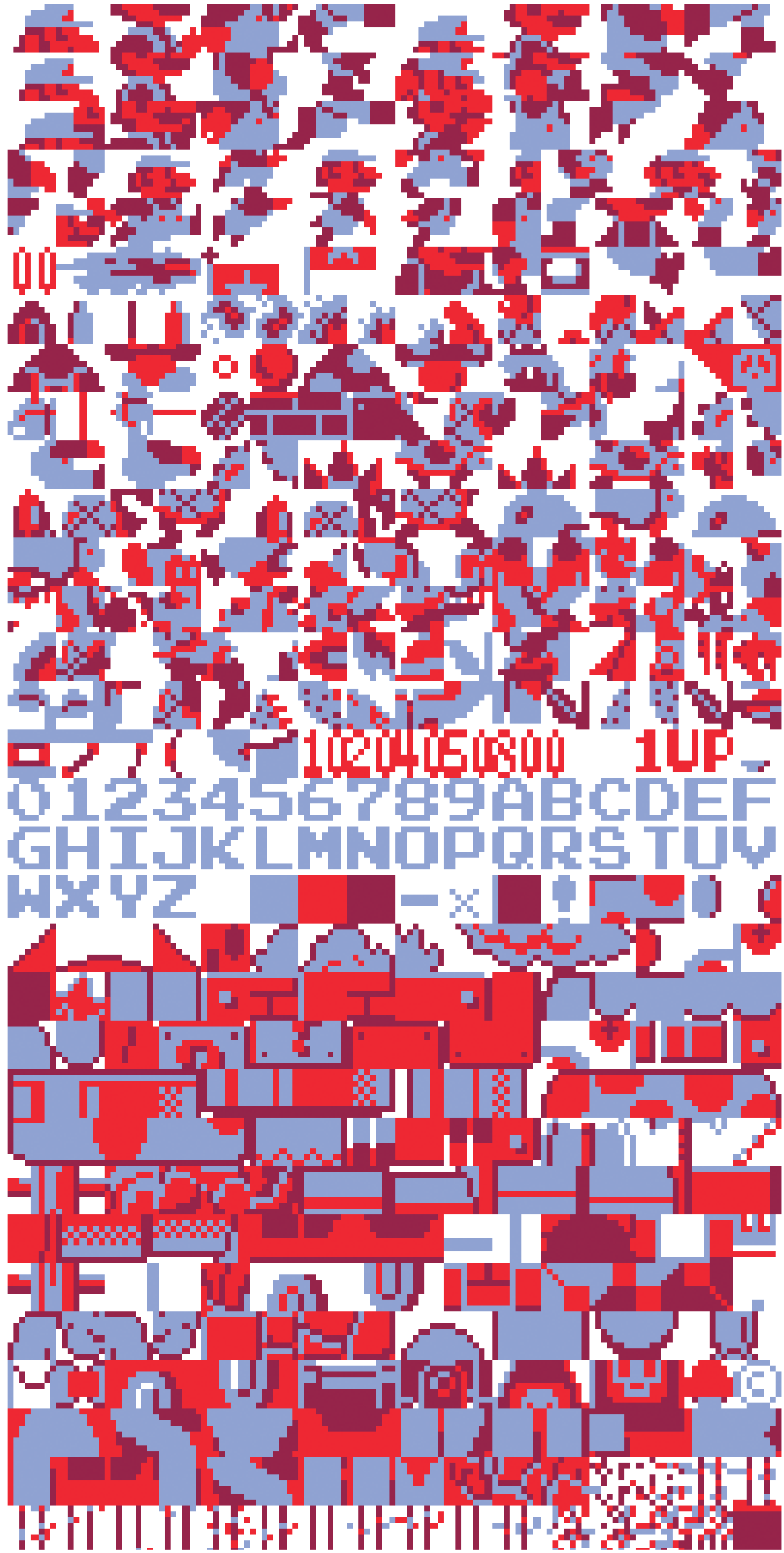
are suitable for teaching CS subjects including many noncomputer games or games that are based on dedicated devices (for example, Lego robots). Our focus is on interactive graphical computer games. It is important to recognize that in the CS education arena the term "computer game" is often used to refer to the attempts at, and the results of, effective and meaningful integration of *animated graphical visualization and various degrees of interactivity*. Because of the unknown entertainment value, strictly speaking, much of these are interesting and innovative teaching materials and are not computer games in a commercial sense.

As discussed in Sung et al.,[33] when examining recent efforts in integrating computer games into CS classes, we observe three general categories.

**1. Game development classes.** These are entire curricula,[7,40] individual classes,[4,6,14,31,39] or capstone projects[2,29] designed specifically to develop new games as an end product. The education committee of the main professional organization for the games industry, The International Game Developer Association (IGDA), has proposed a comprehensive curriculum framework conveying the industry's articulated desires for well-trained college graduates seeking jobs in the game industry. When evaluated against the IGDA curriculum framework, we see that these classes cover all the major core topic areas. Students in these classes must be concerned with all aspects of producing a real game including entertainment value, visual quality, audio effects, physics simulations, and real-time performance.

**2. Game programming classes.** These are classes (for example, Kuffner's CMU course[37]) designed specifically to study technical aspects and issues involved in building games. For example, topics covered may include event loops, path planning algorithms, and terrain representation. These classes typically do not require building an end product and the topics covered are general and usually are applicable to different domains. These classes concentrate on covering the game programming topic area in the IGDA curriculum framework.

**3. Game development client.** These are existing CS classes that creatively in-




VISUALIZATION OF A DECODED NINTENDO GAME BY BEN FRY (HTTP://BENFRY.COM/MARIOSOUP) tegrate games into their existing curriculum. Typically, games are used as programming assignments,[1,3,5,22,25,33,34,37] or to teach abstract concepts,[11,15,30] or as an example application area to teach the concepts involved in an entire topic area.[9] These are traditional CS classes that exist independent of game programming. These classes are actually clients of game development where they use game development as a vehicle to deliver specific abstract concepts. After these classes, students are expected to understand the abstract CS concepts, and not the details of game development.

Courses in the first two categories are *new* courses designed to teach students about game development. Over time, as the game development field matures, it is expected that these courses will evolve and eventually some of the contents will become part of the standard CS curriculum. This is not unlike the early years of many existing disciplines in CS (for example, software engineering[13] or computer graphics[8]), where the syllabi of pioneering courses consolidated as the disciplines mature. Courses in the third category, the "game development clients," are traditional CS courses that can be found in existing CS curricula. The earliest work in this area[1,12] adapted games almost anecdotally without holistic considerations; most of the more recent work is structured around addressing core competency areas with reference to the ACM Curriculum. Accordingly, courses in the "game development clients" category can be divided into two broad efforts: introductory programming classes (CS1/2) and advanced/elective classes.

## Games and Introductory Programming Classes

Many CS educators recognized and took advantage of younger generations' familiarity and interests for computer video games and integrate related contents into their introductory programming courses. Because these are the first courses students encounter, they build excitement and enthusiasm for our discipline.[24, b] Based on the type of effort required by faculty, existing work done in this area can be classified into three broad approaches:

*Little or no game programming*.[9,17] In these courses students learn by playing custom games but they do not actually program the games.

*Per-assignment game development.*[3,21,32,33,38] All these classes developed games as part of individual programming assignments. In each case, isolated games are designed around technical topics being studied.

*Extensive game development.* For example, faculty must design programming assignments based on custom library,[39] general game engines,[4] dedicated game engines,[25] specialized programming environments,[22] custom object-oriented class hierarchies,[25] specific curricula,[23] or new programming languages.[11]

Much of this work reported resounding successes with drastically increased enrollments and student successes.[3,11,23] Based on these results, it is well recognized that integrating computer gaming into CS1 and CS2 (CS1/2) courses, the first programming courses students encounter, is a promising strategy for recruiting and retaining potential students. With the enrollment challenges faced by the CS discipline, it is desirable and important that this strategy can be adopted widely by all interested faculty and departments.

However, most of the existing work in this area is based on pioneering exploratory projects by faculty members with expertise in computer graphics and gaming.[3,23,28] With few exceptions, these projects are *student-centric* where the main goals of study are student engagement and various learning outcomes. Adaptability and generality of the resulting materials are usually not main concerns. For the faculty members teaching CS1/2 courses, most of which are without computer graphics or gaming background, it can be challenging to take advantage of these results.

In addition, when considering experimentation with CS1/2 courses, it is important to appreciate institutional oversight procedures. Though becoming less controversial in recent years, many CS educators continue to be unsure about integrating gaming in formal educational settings.[20] It can be challenging in departmental committees to arrive at consensus for significant modifications to CS1/2 courses, especially if the modifications involve computer games. For these reasons, to be widely adaptable, game-related CS1/2 materials should be designed with the following considerations:

1. The materials should not demand knowledge in computer games or graphics.
2. The materials should include independent modules that are limited in curriculum scope.
3. The materials should support selective experimentation by individual faculty members in small-scale pilot demonstration projects in their existing courses.

*Selective Gradual Adoption.* Results from the extensive game development approach discussed previously typically include large amounts of adoptable/adaptable courseware materials. However, using these materials often requires a significant investment of time, for example, understanding a game engine, or significant reworking of an instructor's existing curriculum. Because of the considerable overhead, results from this approach are typically not suitable for selective adoption.

In terms of suitability for selective adoption, we expect that results from the per-assignment game development approach will be most applicable. For example, one could selectively replace nongame assignments in existing classes by the corresponding games assignments. However, because of the pioneering nature of work in this area, many of the results on per-assignment game development are "anecdotal" and do not discuss the impact of such assignments on the CS1/2 curriculum holistically. For example, the results from Huang only involve turn-based strategic games,[21] Ross only discusses puzzle games,[32] and the discussion from Valentine is based on a single game.[38]

The *Game-Themed Introductory Programming Project* at the University of Washington, Bothell[c] is specifically designed to address these issues. In the first phase of our project, we have designed and built general game-themed CS1/2 programming assignment modules that demand no existing knowl-

b It is important to reiterate that, after these classes students are expected to understand abstract programming concepts rather than concepts specific to building games.

c http://depts.washington.edu/cmmr/Research/XNA Games/index.php edge of games or graphics from the faculty,[33] and have demonstrated it requires minimum changes to existing classes in order to successfully adopt these materials. Currently in the second phase of our project, we are building game-themed examples and tutorials designed to provide a pathway for interested faculty to gradually incorporate game-related materials into their existing courses. Our project is student-centric because our materials allow students to practice CS concepts in a more real-world-like context. More importantly, the materials are also *faculty-centric* because these materials are the stepping-stones for faculty to begin experimenting with a promising new approach to teaching CS1/2 courses.

### Games and Elective CS Courses

As highlighted earlier, the CS education community has a sound understanding of how to integrate visualization and interactivity in delivering CS1/2 content and has achieved impressive successes. In comparison, there is a relatively modest amount of work done in integrating computer games into existing traditional CS elective classes. This is not surprising as a successful systematic integration requires the delivery of an entire technical topic area to lend itself well in visualization and interactivity. There are anecdotal examples of using game content in delivering selective topic areas (for example, design patterns,[16,27] or spatial search algorithms[34]). These are small-scale projects not meant to address entire courses as identified in the standard CS curriculum.

Artificial intelligence (AI),[9] software engineering (SE),[5,10,37] and computer graphics (CG)[36] are examples of elective courses where published results describe attempts at systematically integrating game development. In all of these cases, the stated student learning outcomes are similar to those from the typical CS curriculum and do not include competencies involved in game development as defined by the IGDA curriculum framework. In these classes, students study the core topic areas and implement games to demonstrate their understanding of the fundamental CS concepts. This work reported high student engagement and enthusiasm, while pointing out that the faculty members involved must develop large amounts of software infrastructure to facilitate and support students' game development.

**While it is the case that proper integration of game development and game content in CS classes have the potential to further engage students resulting in higher success rates, it is not the case that any game content will result in having a positive impact.**

Notice that all three of these topic areas have significant overlaps with computer games in general: intelligent behavior (AI) is one of the most important attributes of modern games, SE methodologies are applicable in any software product development, and topics in CG are the conceptual framework for visualization in games. One can argue that for these topic areas, it is relatively straightforward to integrate game content in a consistent manner. In general, for topic areas that do not offer obvious overlaps with computer games or game development (for example, compiler or programming languages), dedication and creativity would be required to develop the elaborate infrastructure and to systematically integrate the new contents. In these cases, one should carefully examine the trade-offs between required efforts, expected benefits, and consider other perhaps more appropriate practical contexts (for example, popular applications on the Internet).

### Guidelines for Consideration

While it is the case that proper integration of game development and game content in CS classes have the potential to further engage students resulting in higher success rates, it is not the case that any game content will result in having a positive impact. In addition, when exploring the potential for development or adoption of game content, we must work within the bounds of institutional oversights and be conscious about the expertise areas of faculty members. The following are some factors for consideration:

- *Institutional oversight.* Departmental committee consensus is often required for significant changes to core courses. Because of the potential impact, it can be especially challenging to arrive at a consensus for modifications to introductory-level courses like CS1/2. A strategy is to design limited-curriculum-scope experiments to gain experience (and collect results) to assist the committee's decision-making process.
- *Faculty background.* Many faculty members did not grow up playing computer games and most are not familiar with graphics programming. When

developing or evaluating materials for adoption, it is essential to pay attention to the prerequisite knowledge. An ideal approach would be to clearly separate and hide the graphical and user interactivity functionality. In this way, faculty and students only need to concentrate on the core CS concepts.

▸ *Gender and expertise neutrality.* As with any powerful tool, inappropriate use of games can backfire and result in further alienation of underrepresented groups.[20] It is important that the gaming materials are gender and expertise neutral. For example, it is important to avoid violence and unnecessary competitions.[26] The materials used should discourage the addition of superfluous "eye-candy" graphical enhancements, or user interaction programming by students with extensive prior programming experience. Doing so will help to avoid intimidating other less-experienced students.

▸ *Infrastructure support.* Free and simple are the keywords here. Given the financial reality of most schools, all materials must be freely available; the associated institutional infrastructure requirements must be modest and straightforward.

▸ *Conceptual integrity.* Our discussion focuses on the traditional CS courses. It is important to remember that ultimately, the goal is to facilitate students' learning about the core CS concepts. Any dilution, even in favor of acquiescing to some students' desire and motivation to become game developers, would do the students a disservice.

▸ *Textbook availability.* As in all pioneering work, mature and well-organized materials are mostly under development. Although there are some textbooks available for specific approaches (for example, CS1/2,[11] computer graphics[35]), mostly, one must be ready to develop custom reading material to guide students along.

**Acknowledgments**

This work is supported in part by the National Science Foundation grant DUE-0442420 and Microsoft Research under the Computer Gaming Curriculum in Computer Science RFP, Award Number 15871 and 16531. All opinions, findings, conclusions, and recommendations in this work are those of the authors and do not necessarily reflect the views of the National Science Foundation or Microsoft.

**References**

1. Adams, J.C. Chance-It: An object-oriented capstone project for CS-1. In *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 1998, 10–14.
2. Barnes, T., Richter, H., Powell, E., Chaffin, A., and Godwin, A. Game2learn: Building CS1 learning games for retention. In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ACM, NY, 2007, 121–125.
3. Bayliss, J.D. The effects of games in CS1-3. *Journal of Game Development 2*, 2 (2007).
4. Bierre, K., Ventura, P., Phelps, A., and Egert, C. Motivating OOP by blowing things up: An exercise in cooperation and competition in an introductory Java programming course. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2006, 354–358.
5. Chen, W-K and Cheng, Y.C. Teaching object-oriented programming laboratory with computer game programming. *IEEE Transactions on Education 50*, 3 (Aug. 2007), 197–203.
6. Clark, B., Rosenberg, J., Smith, T., Steiner, S., Wallace, S., and Orr, G. Game development courses in the computer science curriculum. *J. Comput. Small Coll. 23*, 2 (2007), 65–66.
7. Coleman, R., Krembs, M., Labouseur, A., and Weir, J. Game design and programming concentration within the computer science curriculum. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 545–550.
8. Cunningham, S., Brown, J.R. Burton, R.P., and Ohlson, M. Varieties of computer graphics courses in computer science. In *Proceedings of the 19th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 1988.
9. da Silva, F.S.C. Artificial intelligence for computer games. University of Sao Paulo, Microsoft Academic Alliance Repository Newsgroup, 2006; http://www.msdnaacr.net/curriculum/pfv.aspx?ID=6210.
10. da Silva, F.S.C. Software engineering for computer games. University of Sao Paulo, Microsoft Academic Alliance Repository Newsgroup, 2006; http://www.msdnaacr.net/curriculum/pfv.aspx?ID=6211.
11. Dann, W., Cooper, S., and Pausch, R. *Learning to Program with Alice.* Prentice Hall, Upper Saddle River, NJ, 2006.
12. Faltin, N. Designing courseware on algorithms for active learning with virtual board games. In *Proceedings of the 4th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*. ACM Press, NY, 1999, 135–138.
13. Freeman, P. Software engineering education: Needs and objectives. In *Proceedings of the ACM SIGCSE-SIGCUE Technical Symposium on Computer Science and Education*. ACM Press, NY, 1976, 266.
14. Frost, D. Ucigame, A Java library for games. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2008.
15. Gestwicki, P.V. Computer games as motivation for design patterns. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2007, 233–237.
16. Gestwicki, P.V. Computer games as motivation for design patterns. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2007, 233–237.
17. Giguette, R. Pre-games: Games designed to introduce CS1 and CS2 programming assignments. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2003, 288–292.
18. Guzdial, M. Contextulized computing education. Invited Presentation, Microsoft Research Faculty Summit; http://home.cc.gatech.edu/guzdial/169 (July 2008).
19. Haden, P. The incredible rainbow spitting chicken: Teaching traditional programming skills through games programming. In *Proceedings of the 8th Australian Conference on Computing Education*. Australian Computer Society, Darlinghurst, 2006, 81–89.
20. Haller, S., Ladd, B., Leutenegger, S., Nordlinger, J., Paul, J., Walker, H., and Zander, C. Games: Good/evil. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2008, 219–220.
21. Huang, T. Strategy game programming projects. In *Proceedings of the 6th Annual CCSC Northeastern Conference on the Journal of Computing in Small Colleges*. Consortium for Computing Sciences in Colleges, 2001, 205–213.
22. Külling, M. and Henriksen, P. Game programming in introductory courses with direct state manipulation. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ACM Press, NY, 2005.
23. Leutenegger, S. and Edgington, J. A games-first approach to teaching introductory programming. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2007, 115–118.
24. Lewis, M., Leutenegger, S., Panitz, M., Sung, K., and Wallace, S.A. Introductory programming courses and computer games. In *Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education*. Mar. 2009.
25. Lewis, M.C. and Massingill, B. Graphical game development in CS2: A exible infrastructure for a semester long project. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2006, 505–509.
26. Natale, M.J. The effect of a male-oriented computer gaming culture on careers in the computer industry. *SIGCAS Comput. Soc. 32*, 2 (2002), 24–31.
27. Nguyen, D.Z. and Wong, S.B. Design patterns for games. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2002, 126–130.
28. Parberry, I., Kazemzadeh, M.B., and Roden, T. The art and science of game programming. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2006, 510–514.
29. Parberry, I., Roden, T., and Kazemzadeh, M.B. Experience with an industry-driven capstone course on game programming: Extended abstract. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2005, 91–95.
30. Pulimood, S.M. and Wolz, U. Problem solving in community: a necessary shift in CS pedagogy. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2008, 210–214.
31. Repenning, A. and Loannidou, A. Broadening participation through scalable game design. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. New York, NY, USA, 2008.
32. Ross, J.M. Guiding students through programming puzzles: Value and examples of Java game assignments. *SIGCSE Bull. 34*, 4 (2002), 94-98.
33. Sung, K., Panitz, M., Wallace, S., Anderson, R., and Nordlinger, J. Game-themed programming assignments: The faculty perspective. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2008, 300–304.
34. Sung, K. and Shirley, P. Algorithm analysis for returning adult students. In *Proceedings of the Sixth Annual CCSC-NW Conference. J. Computing Sciences in Colleges 20*, 2 (Dec. 2004) 62–72.
35. Sung, K., Shirley, P. and Baer, S. *Essentials of Interactive Computer Graphics: Concepts and Implementation.* A.K. Peters, Wellesley, MA, 2008.
36. Sung, K., Shirley, P. and Reed-Rosenberg, R. Experiencing aspects of games programming in an introductory computer graphics class. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2007, 249–253.
37. Sweedyk, E., deLaet, M., Slattery, M.C., and Kuffner, J. Computer games and CS education: Why and how. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, NY, 2005, 256–257.
38. Valentine, D.W. Playing around in the CS curriculum: Reversi as a teaching tool. *J. Comput. Small Coll. 20*, 5 (2005), 214–222.
39. Wallace, S.A. and Nierman, A. Using the Java instructional game engine in the classroom. *J. Comput. Small Coll. 23*, 2 (2007), 47–48.
40. Zyda, M. Guest editor's introduction: Educating the next generation of game developers. *IEEE Computer 39*, 6 (June 2006), 30–34.

**Kelvin Sung** (ksung@u.washington.edu) is a professor in the department of Computing and Software Systems at the University of Washington, Bothell, WA.

DOI:10.1145/1610252.1610274

# Technical Perspective
# Design Tools for the Rest of Us

By James A. Landay

THERE ARE MANY who believe we are on the verge of the biggest change in the way products are made since the Industrial Revolution kicked into high gear over 150 years ago. This is not a revolution where factories will make products faster or more efficiently, as with the earlier manufacturing revolution; it is one where individuals will be in control of personal or customized production. This idea, often referred to as personal fabrication,[6] is that people will have the means at home and in their offices to "print" new physical objects to their exact specifications. This is no longer simply a far-fetched idea of science fiction. The ever-decreasing price of 3D printers as well as the demonstration of printable plastic electronics and other such fabrication machines leads one to see that personal fabrication is not far off.

Some might question whether individuals will want to design and produce new physical objects or even customize existing objects. Although it is doubtful we will use these technologies for all of the physical objects we own and use, the proliferation of customized t-shirts, shoes, posters, and other objects available via Web-based custom product shops gives credence to the idea that a large segment of the population will be interested in creating objects customized to their specifications. Whether personalized objects will come through these Web-based services or be created right at home on 3D fabrication devices is a question only time will answer.

One major problem often ignored by this view of the future is how the individual will specify the design of the objects they wish to create with these technologies. It is one thing to stipulate a phrase to go on a t-shirt in one of a large set of predefined layouts, colors, and font families, and it is another to define the exact shape, color, patterns, and working mechanisms of a 3D part or a complete product. Training most of the public in the skills of industrial design is simply not viable.

Design tools that can be used by everyday users to create customized products is the solution to the specification problem in personalized fabrication. In fact, it is probably the most important technical problem that must be solved before the vision of personal fabrication can come into being. This is why the work of Igarashi and Igarashi described in the following article is so important to the future—not only for computing, but for production. Many in the research community have worked to bring powerful, sketch-based design tools to non-computing experts, for example, architects,[7] students,[1,10] musicians,[3,5] interface designers,[9,11] multimedia authors,[2] and amateur animators.[4] Some of the most innovative work in this area has been accomplished over the last 10 years by the University of Tokyo group led by Takeo Igarashi. The work has led the field in the design of 3D objects by non-experts by creating design tools that allow a user to simply sketch a 2D representation and then have the system carry out the difficult underlying computation to build a reasonable 3D model.[8]

At first read, this paper on designing plush toys may seem to address a narrow portion of the family of 3D objects that users will want to fabricate, but these plush toys share many of the important characteristics of other complex design objects: different patterns/colors for different parts, a non-rigid filling that can change the shape of the object depending on how the outside fabric is sewn, the production of instructions to assemble the requisite pieces to create the finished product. The system described here is technically sophisticated. By making the modeling and simulation phases concurrent, the underlying model is always maintained as a realizable product. This sophisticated computing makes it much easier for the designer, as they will never get into the trap of creating something that simply cannot be built. This paper is the first of what I hope will be a series of design tools that will one day allow the public to realize their own personally designed products.


**References**

1. Alvarado, C. and Davis, R. Resolving ambiguities to create a natural computer-based sketching environment. *IJCAI* (2001), 1365–1374.
2. Bailey, B.P., Konstan, J.A., and Carlis, J.V. DEMAIS: Designing multimedia applications with interactive storyboards. In *Proceedings of ACM Multimedia* (Ottawa, Canada, 2001), 241–250.
3. Buxton, W., Sniderman, R., Reeves, V., Patel, S. and Baecker, R. The evolution of the SSSP score editing tools. *Computer Music J. 3*, 4 (1979), 14–25.
4. Davis, R.C., Colwell, B., and Landay, J.A. K-sketch: A 'kinetic' sketch pad for novice animators. In *Proceedings of the 26th Annual ACM SIGCHI Conference* (Florence, Italy, 2008).
5. Forsberg, A., Dieterich, M., and Zeleznik, R. The music notepad. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*. (San Francisco, CA, 1998).
6. Gershenfeld, N., *Fab: The Coming Revolution on Your Desktop—from Personal Computers to Personal Fabrication.* Basic Books, 2007.
7. Gross, M.D. and Do, E.Y.-L. Ambiguous intentions: A paper-like interface for creative design. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (1996), 183–192.
8. Igarashi, T., Matsuoka, S., and Tanaka, H. Teddy: A sketching interface for 3D freeform design. In *Proceedings of ACM SIGGRAPH* (1999), 409–416.
9. Landay, J.A. and Myers, B.A. Sketching interfaces: Toward more human interface design. *IEEE Computer 34*, 3 (2001), 56–64.
10. LaViola, J. Advances in mathematical sketching: Moving toward the paradigm's full potential. *IEEE Computer Graphics and Applications 27*, 1 (2007), 38–48.
11. Newman, M.W., Lin, J., Hong, J.I., and Landay, J.A. DENIM: An informal web site design tool inspired by observations of practice. *Human-Computer Interaction 18*, 3 (2003), 259–324.


**James A. Landay** is an associate professor in computer science and engineering at the University of Washington and a founder of its Design:Use:Build (DUB) Center.

DOI:10.1145/1610252.1610275

# Designing Plush Toys with a Computer

By Yuki Igarashi and Takeo Igarashi


## Abstract

**We introduce Plushie, an interactive system that allows nonprofessional users to design their own original plush toys. To design a plush toy, one needs to construct an appropriate two-dimensional (2D) pattern. However, it is difficult for nonprofessional users to appropriately design a 2D pattern. Some recent systems automatically generate a 2D pattern for a given three-dimensional (3D) model, but constructing a 3D model is itself a challenge. Furthermore, an arbitrary 3D model cannot necessarily be realized as a real plush toy, and the final sewn result can be very different from the original 3D model. We avoid this mismatch by constructing appropriate 2D patterns and applying simple physical simulation to it on the fly during 3D modeling. In this way, the model on the screen is always a good approximation of the final sewn result, which makes the design process much more efficient. We use a sketching interface for 3D modeling and also provide various editing operations tailored for plush-toy design. Internally, the system constructs a 2D cloth pattern in such a way that the simulation result matches the user's input stroke. We successfully demonstrated that nonprofessional users could design plush toys or balloon easily using Plushie.**


## 1. INTRODUCTION

A computer can be a powerful tool for designing real-world objects. One can build a virtual three-dimensional (3D) model on a computer using computer-aided design (CAD) and use the model to run various simulations with computer-aided engineering (CAE) without the need to build or damage costly real objects. The benefits are evident in many areas from architecture to automobile design. However, these tools are mainly designed for professional users and are not particularly accessible to the ordinary person. The construction of a 3D model using a standard CAD system is tedious, and running a physical simulation using a standard CAE system requires a certain level of expertise.

Our goal is to bring the benefits of CAD and CAE to the hands of nonprofessional users including children. This article introduces our plush-toy design system,[18] Plushie, as an example of our efforts to achieve this goal. Plush toys are familiar objects in our daily lives, but their design is difficult. One must design an appropriate two-dimensional (2D) pattern to obtain a particular 3D shape, but the relationship between the two is nontrivial, and intensive experience and knowledge are required to achieve satisfactory results. As a result, most people simply buy ready-made plush toys and do not enjoy the design and construction of their own.

Figure 1. Overview of Plushie system.

We have provided a way for people to design their own toys using a simple but powerful modeling tool that tightly integrates a sketching interface with physical simulation in the modeling process.

Plushie allows the user to design a plush toy from scratch by simple sketching operations.[18] The user first draws the desired silhouette, and the system automatically generates a 3D plush-toy model and corresponding 2D cloth pattern. The user can also edit the model, e.g., cut it or add a part, using a simple sketching interface, and the 3D model and 2D cloth pattern are automatically updated. The 3D model is the result of a physical simulation that mimics the inflation of the sewn 2D cloth patch. Therefore, the model on the screen is always a good estimate of the final sewn result (Figure 1). When we ran workshops in a museum to have novice users try our system, we observed that even children could design their own plush toys.

We first give an overview of sketching interfaces for 3D modeling and previous efforts to enable end users to design physical objects. We then describe the user interface and implementation of the Plushie system, followed by results and user experiences. Finally, we conclude the article with some discussion of future work.

## 2. SKETCHING INTERFACES FOR 3D MODELING

The sketching interface part of Plushie is an evolution of the Teddy system we presented in 1999.[11] That system allowed the user to create an interesting 3D model simply by sketching a silhouette of the target model (Figure 2, left). It was designed for the modeling of free-form rotund models, a task that is particularly difficult using standard modeling interfaces. Figure 3 shows an example of a modeling sequence using Teddy. The user's strokes are shown in red, and everything else is inferred and drawn by the system. The user first draws

A previous version of this paper appeared in *Proceedings of SIGGRAPH 2007/ACM Transactions on Graphics 26*, 3 (2007), 45.

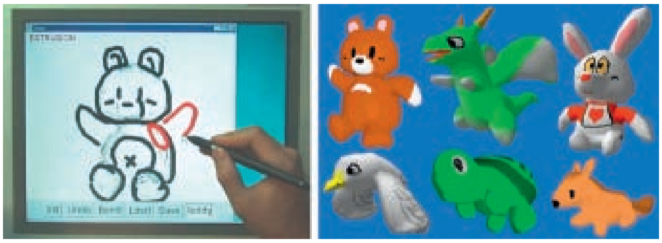


**Figure 2. Screenshot of Teddy and sample 3D models created using the system.**

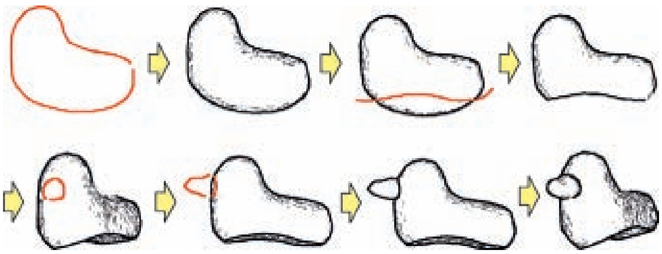


**Figure 3. Modeling session in Teddy. The user can create a 3D model using simple sketching operations.**

the silhouette of the base primitive, and the system generates the corresponding 3D geometry. The user then draws a stroke across the model and the system cuts the model at the line. The user can also add parts to the base model by drawing two strokes. Figure 2 right shows several 3D models created using the system.

Several sketching systems for free-form shapes were developed after Teddy. The original Teddy system used polygonal meshes, but some later systems experimented with other representations such as voxels[20] and implicit surfaces.[2] Some systems extended the interface to support subsequent editing by direct manipulation. ShapeShop[21] represents a model as a collection of blob primitives and allows the user to move or scale each primitive. Fibermesh[19] keeps the original stroke as a control curve on the model surface and allows the user to adjust the shape by deforming the curves. However, all these systems are designed for purely virtual 3D models without consideration of the physical properties of materials. Plushie is innovative in that it shows the feasibility of using a sketching interface for free-form shapes in the design of physical objects.

## 3. DESIGNING PHYSICAL MODELS WITH A COMPUTER

Another key aspect of our work is the tight integration of physical simulation into the 3D modeling process. In traditional applications, modeling and simulation are completely separate. A virtual model is created in 3D modeling software without considering any physical constraints, and it is then passed to a simulation environment. If the simulation result reveals a problem, the user returns to the model to fix the problem. We made this process more efficient by running the simulation concurrently with the modeling to create only models that are physically realizable. In this way, the user can more efficiently explore the design dimensions within realistic constraints. From the user's point of view, the model generated by the system may not correspond exactly to the shape that was input, but it will be a physically realizable shape reflecting the input shape.

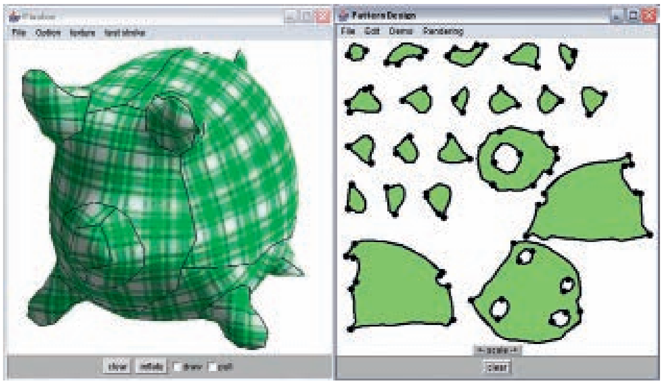
Some recent systems have tried to incorporate fast physical simulation into an interactive design process. Igarashi and Hughes developed a mark-based interface for putting clothing on a virtual character,[10] and Decaudin et al. proposed a system for designing an original garment via sketching.[6] Both used simple geometric simulations to represent the physical properties of cloth material. Masry and Lipson described a system in which the user can quickly build a CAD model by sketching and immediately apply a finite element analysis to the model.[15] However, the model construction is computed before the simulation in these systems, and no dynamic feedback loop exists between the simulation result and the original user input.

Several efforts have also been made to support the design of physical objects by end users in the computer graphics research community. Mitani and Suzuki[16] and Shatz et al.[22] presented automatic segmentation of a 3D model into surface patches that can be perfectly flattened onto a plane without distortion for constructing paper craft models. Similarly, Julius et al. proposed similar method for plush toys[14] allowing small distortion. Pillow system[17] facilitates the manual segmentation of a model by providing automatic flattening and by showing the result of physical simulation. These systems make plush-toy design more accessible, but the fundamental challenge of creating an original plush toy is still unresolved.

## 4. THE PLUSHIE SYSTEM

The system consists of two windows: one shows the 3D plush-toy model being constructed and the other shows the corresponding 2D pattern (Figure 4). The user works on the 3D view, interactively building the 3D model by using a sketching interface. The 2D view is mainly for reference but the user can also edit the 2D pattern directly when desired. The 3D model is produced from a physical simulation of the assembled 2D pattern. After each input from the user,

**Figure 4. A screen snapshot of the Plushie system.**



the system updates the 2D pattern so that the simulation result matches the user input. This guarantees that the model is always realizable as a real plush toy and that the 2D pattern is readily usable as a template for cutting and sewing real fabric.

The modeling operations are based on the Teddy system.[11] The user interactively draws free-form strokes on the canvas as gestures and the system automatically generates a 3D model and corresponding 2D cloth pattern. We also provide some special editing operations tailored for plush-toy design.

**Creating a New Model**: Starting with a blank canvas, the user creates a new plush-toy model by drawing its silhouette as a closed free-form stroke. The system automatically generates two cloth patches corresponding to the stroke and visualizes the shape of the resulting plush toy by applying a simple physical simulation (Figure 5).

**Cut**: The user can cut the model by drawing a stroke that starts outside of the model, crosses it, and ends outside of the model (Figure 6). The model is cut at the intersection and flat patch is generated at the cross-section. This operation is useful for creating relatively flat surfaces, such as those in a foot or belly.

**Creation of a Part**: The user can add protruding parts such as the ears and arms to the base model by drawing a single stroke that defines the silhouette of the part. The stroke should start and end on the base model (Figure 7a). The system generates two candidate shapes and presents them to the user as suggestions[9] (Figure 7b). One is for fat, rounded parts like the body, arm, and leg (Figure 7c). Their base is connected to the base model with an open hole. The other candidate shape is for thin parts like ears and the tail whose base is closed (Figure 7d). The user clicks the desired thumbnail and the system updates the main model accordingly. We found that the ability to create thin parts with a single stroke is particularly useful. They are frequently seen in real toys and are difficult to design using standard modeling software. Figure 20 shows a couple of example models with thin parts.

**Pull**: The user can grab a seam line and pull it to modify the shape. For example, the user can pull an ear to make it larger when it is smaller than the other (Figure 8). The pulling operation begins when the user starts dragging on the background region near a seam line. The system changes the mouse cursor when it approaches a seam line to indicate that the user can start pulling. We use the peeling interface introduced by Igarashi et al.[12] to adjust the size of the region to be deformed, that is, the larger area is deformed as the user pulls more. The system continuously updates the 2D cloth pattern during pulling and shows the simulation result in the 3D view.

**Insertion and Deletion of Seam Lines**: The modeling operations performed thus far automatically generate 2D patches according to predefined algorithms and seam lines (patch boundaries) appear on the 3D model surface without the user's explicit control. However, it is sometimes desirable for knowledgeable users to design seam lines manually, for more detailed control. This is especially important when using nonstretchy cloth as in balloon models because one needs to divide a rounded surface into many almost-developable small patches (Figure 19 bottom).

The user can add a new seam in the seam line drawing mode by drawing a free-form stroke on the model surface (Figure 9). The corresponding cloth patch is then automatically cut along the new seam line. If the stroke crosses the entire patch, the patch is divided into two separate

**Figure 5. Creating a new model.**

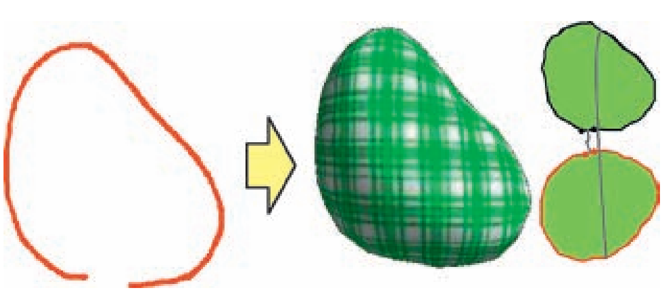


**Figure 6. Cut operation.**

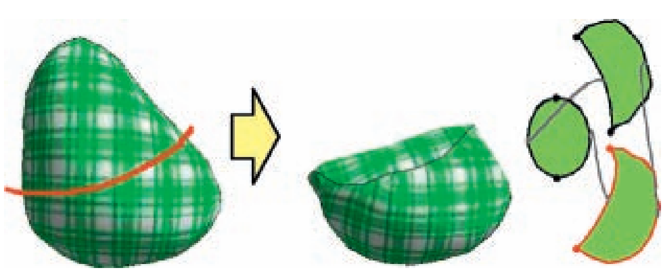


**Figure 7. User interface of part creation. (a) The user draws a stroke and (b) the system suggests two different possibilities. The user chooses one (c, d).**

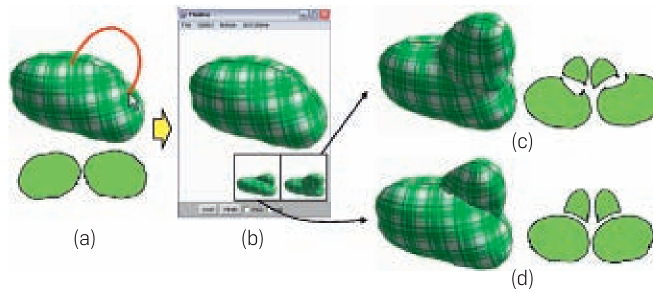



**Figure 8. User interface of the pull operation.**

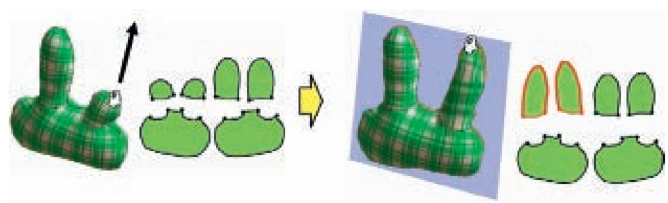


Figure 9. Insertion of a seam line. (a) Before drawing a line, (b) after drawing a line, (c) the seam line's two endpoints snap at other seam lines, and (d) After pulling the seam line.

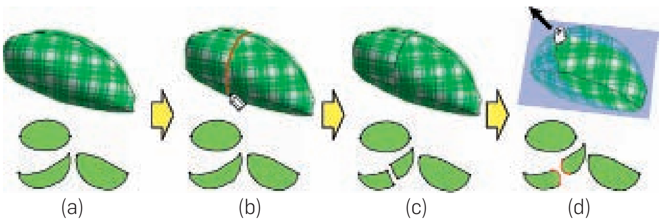



Figure 10. Deletion of a seam line.

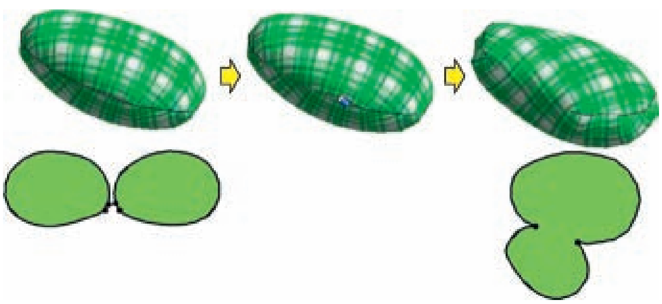
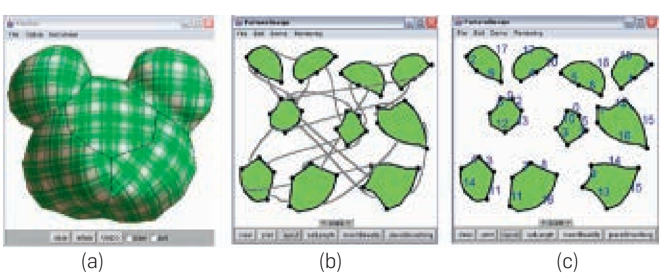


Figure 11. Patches connected to each other using connectors (b) and numbers (c).




patches. If the stroke starts or ends in the middle of a patch, it becomes a dart. The 3D geometry does not change immediately after the insertion of these seam lines, but the user can pull the seam line afterwards to modify the shape. This operation is very useful for creating a salient feature in the middle of a flat patch. Deletion is achieved by tracing the target seam line in the erasing mode. This merges the separated patches together and thus flattens the area (Figure 10).

**Operations on the 2D Pattern View**: The 2D pattern view is mainly used to preview the pattern to be printed for sewing, but it also works as an interface for advanced users to edit the pattern directly. The preview helps the user to understand the relationship between the 3D model and 2D patches and gives a sense of the labor required for assembling the patches. The system can display how patches are connected by showing connectors or paired numbers (Figure 11). Connectors are useful for understanding the relationship on the screen and numbers are useful as a printed reference on each

Figure 12. Pulling a 2D patch.

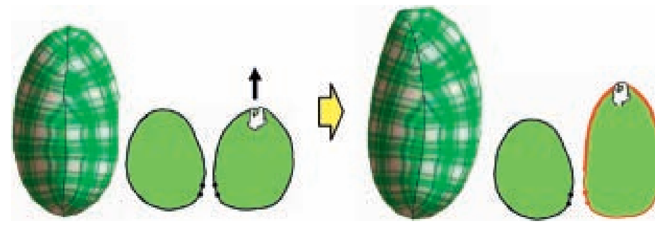
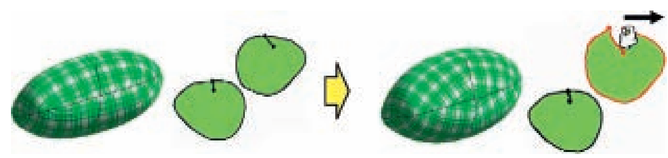


Figure 13. Opening a dart line.



patch. The system provides an automatic layout and manual arrangement interface for preparing the final pattern to be printed.

The system also allows the user to edit the patches directly by using the pulling interface. The user can grab the boundary of a patch and pull it to deform the shape.[12] We again use a peeling interface to adjust the size of area to be deformed. The effect of 2D deformation immediately appears in the 3D view because of the physical simulation. The ability to deform an individual patch is useful for designing asymmetric shapes such as a penguin belly (Figure 12). The pull operation is also useful for opening a dart line to make a flat patch swell more (Figure 13).

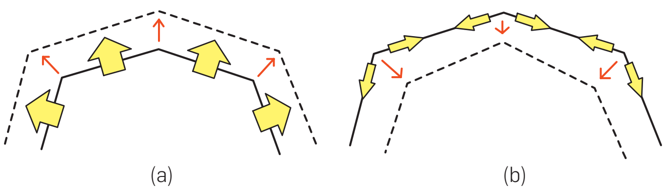
## 5. IMPLEMENTATION

This section briefly describes the implementation of Plushie. A more detailed description is found in our original paper.[18] We use a standard triangle mesh for the representation of the 3D model and 2D patches. We use a relatively coarse mesh (1000–2000 vertices) to achieve interactive performance. Each vertex, edge, and face of the 3D mesh is associated with corresponding entities in the 2D mesh. A 3D mesh is always given as a result of applying a physical simulation to the assembled 2D pattern. To be more precise, the physical simulation applied to the 3D mesh is governed by the rest length of each edge and the rest length is defined in the 2D mesh. For each modeling operation, the system constructs the initial 2D patches and the 3D geometry corresponding to the input stroke, and then runs a physical simulation to update the 3D geometry. The system then adjusts the patch shape so that the simulation results match the input strokes.

### 5.1. Physical simulation

We use a simple static method for the physical simulation. We examined other, more elaborate methods, such as finite element methods,[8] dynamic simulation,[5] and energy minimization,[4] but we found that the simple approach is

Figure 14. Our simple model to mimic stuffing effect. Internal pressure pushes vertices outwards (a) and edge springs pull them back (b).



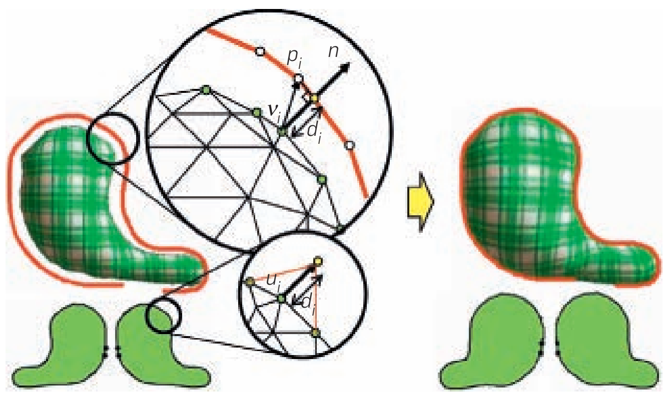

best suited for our purpose. It is easy to implement, fast enough for interactive modeling, and sufficiently robust for dealing with adverse user operations. More importantly, it produces a reasonable estimation of the resulting plush-toy shape. As shown in Figure 14, it successfully reproduces characteristic behaviors seen in the stuffed cloth. This algorithm is also used in a garment capture system.[3]

In each simulation cycle, the system first moves each face slightly in its normal direction to mimic the effect of internal pressure (Figure 14a). The system then adjusts the length of each edge to preserve the integrity of the cloth material[5] (Figure 14b). We decided to prevent stretching only and tolerate compression because plush toys' rotund shape is generated from compression (small wrinkles) along the seam lines. The second part (adjustment of edge length) runs 10 times in each cycle to prevent excessive stretch. It takes ~30 simulation cycles (2 s) to converge in our typical examples. Although it is possible to show the result only after the convergence, we decided to show the intermediate shape because test users preferred to see the inflation process.

### 5.2. 3D modeling operations

**Creating a New Model**: The input stroke is projected onto an invisible plane at the center of the world facing the screen, and the system generates an initial two-sided mesh inside of the closed region. Each side of the mesh is used directly as a 2D patch for the model. The system then applies the physical simulation to the mesh. It inflates the mesh to the direction perpendicular to the viewing direction, but its silhouette actually becomes smaller as it inflates (Figure 15). The system waits until the simulation converges and then starts to adjust the 2D pattern so that the simulation result matches the input stroke. Specifically, the system calculates the distance $d_i$ from a vertex $v_i$ of the 3D mesh along the seam line to the corresponding point $p_i$ in the projected input stroke along its normal direction, and moves the corresponding 2D vertex $u_i$ on the patch boundary to its normal direction by that amount $d_i$. After modifying the patch boundary, the system updates the 2D mesh so that vertices are uniformly distributed inside of the patch. The length of the edges in the updated 2D mesh is then used as the new rest length in the simulation. The system repeats this adjustment process and the physical simulation until convergence. It takes approximately 20 iterations (2 seconds) to converge in our typical examples.

Figure 15. Adjustment process after creation. The system enlarges the 2D pattern so that the simulation result matches the input stroke. The 2D boundary vertex ($v$) moves in its normal direction by the amount proportional to the distance between the corresponding 3D vertex and the input stroke.




This simple algorithm works well in practice for our application. Figure 16 shows some examples in which our algorithm successfully found appropriate 2D patches that yielded the desired 3D shapes. In some situations, the input shape is not realizable as a plush-toy model consisting of two patches. For example, a sharp concavity is not realizable without causing self-intersection in the 2D patch. In these cases, the system terminates the optimization process, leaving a gap between the input stroke and the 3D model. This indicates that the desired shape is not possible with two patches. The user must add additional seam lines to obtain more control.

**Cut**: The system constructs a curved surface by sweeping the cutting stroke on the screen to the viewing direction and dividing the mesh along the surface. The right-hand side of the surface is removed and a new mesh is created on the cross-section. The cross-section is always developable, so the system simply flattens it and uses it as a 2D patch. The system then applies the inflation process to the model. Note that the silhouette of the inflated 3D model does not exactly matches with the input cut stroke because we do not apply any adjustment as in the initial creation.

Figure 16. Physical simulation and shape adjustment. The red lines indicate the input strokes. The top row shows the result of converting the input into patterns directly, and the bottom row shows the outcome when the adjustment process is applied to the patterns. The green shapes in the middle show the simulation results and the brown ones on the right show the real fabric models, both resulting from the 2D pattern on the left.

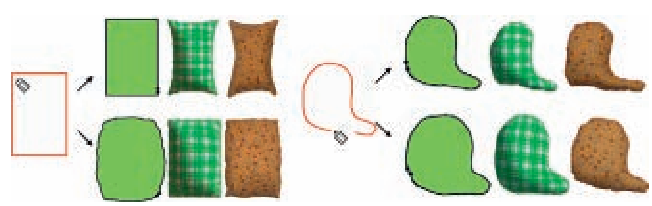


Figure 17. (a) Creation of a part. The system projects the input stroke to a working plane and cuts the base mesh with either an elliptic curve or a line (b). The 3D geometry is constructed by creating a mesh between the projected stroke and the base curves (c).

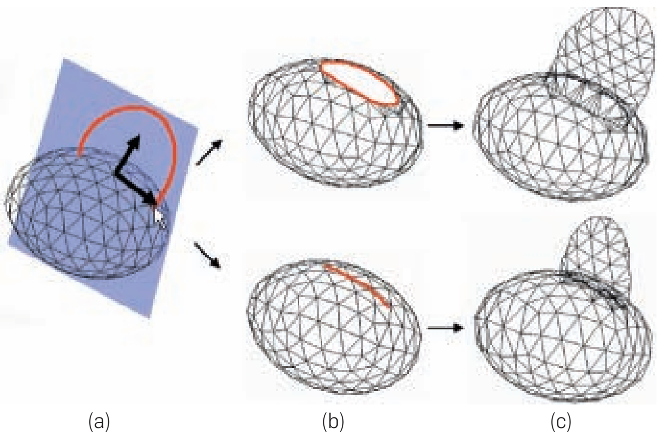
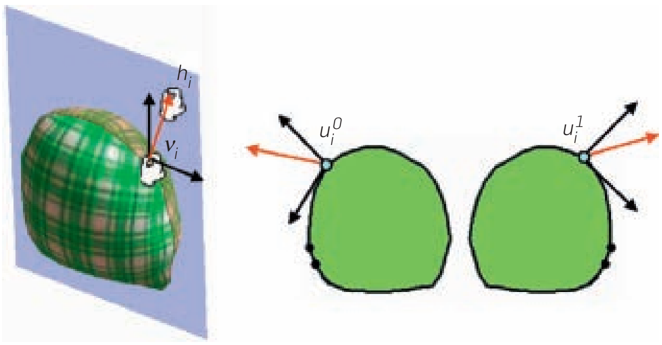



Figure 18. Pulling a vertex on a seam line.




**Creation of a Part**: The system first projects the two endpoints of the input stroke onto the base model surface. A plane that passes through these 3D points and is facing toward the screen is constructed and the input stroke is projected onto it. The system then draws an ellipse on the model surface for constructing a fat part and draws a line for creating a thin part (Figure 17). The ellipse or the line (what we call base curves) is also projected to the plane. The system generates a 2D mesh on the projection plane in the area enclosed by the projected input stroke and the projected base curve. The 2D mesh is duplicated and serves as 2D pattern and as the initial 3D geometry for the added part. As in the initial model creation case, the flat two-sided 3D mesh is inflated by physical simulation. The silhouette of the added part gradually shrinks and the system enlarges the 2D pattern so that the silhouette matches the input stroke as in initial creation.

In case of a part with an elliptic base curve, the system cuts open the base surface and stitches it with the newly created mesh. The result is a single connected mesh, and physical simulation is applied uniformly to the entire mesh. On the other hand, the system does not open the base mesh in case of the linear base curve. The new part is created as an independent closed mesh and the simulation is applied separately to the base mesh and the new part. The base mesh inflates independently of the part mesh, and the base curve is treated as a positional constraint in the simulation of the part mesh (we simply do not move these vertices in the simulation cycle).

**Pull**: The pull operation is a bit involved because the system cannot directly modify the 3D mesh and must do so indirectly by deforming the corresponding 2D pattern. As the user starts pulling a vertex on a seam line, the system first constructs a projection plane that passes through the seam line (Figure 18). The mouse cursor position on the screen is projected onto the projection plane, and it is used as a target position for the pulled vertex during subsequent dragging. The system computes the displacement $\delta_i$ in the local coordinate frame on the projected plane from the original position $v_i$ to the target position $h_i$, and moves the corresponding vertices $u_i^0$ and $u_i^1$ in the 2D mesh in their local coordinate frames by that amount $\delta_i$. These 3D and 2D coordinate frames are defined by the pulled vertex's normal vector and the direction of the seam line. The system iterates this displacement process with physical simulation until it converges. To achieve a smooth deformation, the system also moves the surrounding vertices in the 2D mesh using the curve manipulation method introduced in Igarashi et al.[12] It enlarges the region to be deformed proportional to the displacement of the pulled vertex.

**Insertion and Deletion of Seam Lines**: Insertion of a new seam line is straightforward. The system simply cuts the patch along the added seam line, and basically does not change the result of simulation. Deletion is more complicated because the merged patch is not necessarily developable. The system applies an approximate flattening operation[23] to the merged 3D surface to obtain the geometry of the new 2D patch.

## 6. RESULTS

Plushie is implemented as a Java™ program. Construction of 2D patterns and a physical simulation run in real-time on a 1.1 GHz Pentium M PC. We designed a couple of plush toys using our system and created a real toy based on the printed pattern. A modeling session typically takes 10–20 min and sewing takes 3–5 h. Figure 19 shows a plush toy and balloon model designed in our system. It shows that the physical simulation successfully captures the overall shape of the real objects. We interviewed with professional balloon designers and they supported our system, saying that it can significantly reduce the time necessary for designing original balloon.

The user can assign different textures to individual patches (Figure 20). Therefore the user can explore various design possibilities before actuary sewing the real fabric (such as Figure 20 right). These models also demonstrate the effectiveness of thin parts.

We ran four small workshops to test the usability of the system and found that novice users, mainly children, can successfully create original plush toys using our system. Here is an observation from one of these workshops. Nine

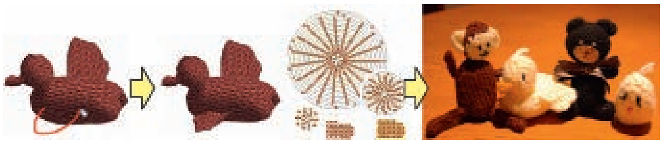
Figure 19. A plush toy and a balloon designed in our system.

Figure 20. Example of texture changed. These models have many thin parts.

Figure 21. Example of original plush toys designed and created by children in the workshop.

children, approximately 10–14 years old, joined the workshop accompanied by their parents. We gave a brief tutorial at the beginning and had them design their own plush toys using the system. It took about an hour for the design. They then printed the designed pattern and sewed a real toy in ~3 h. Figure 21 shows a couple of plush toys created in the workshop. Participants quickly learned how to use the system and successfully designed the 3D models they wanted, with some help from volunteers. Furthermore,

Figure 22. Knitty system allows the user to design an original knitted toy by simply drawing the desired silhouette.



they enjoyed the process. These toys were their own creations and one-of-a-kind designs. Participants also gave us valuable feedback for future improvement. They wanted to have some auxiliary functions such as the ability to design symmetric parts and remove existing parts, but no one complained about the quality of the visual simulation. A perfectly accurate simulation is not necessary because many small variations inevitably occur during the real sewing and stuffing process.

## 7. CONCLUSION

We introduced a plush-toy design system as an example of our efforts to make CAD and CAE accessible to end users. The system allows the user to design a plush toy quickly and simply by combining simple sketching operations. The user draws the desired silhouette on the canvas, and the system automatically generates a 3D plush-toy model and a 2D cloth pattern. The system runs a simple physical simulation in the background so that the resulting 3D model is always a good estimate of the final sewn result. The user can construct a real plush toy by printing the pattern and sewing the resulting pieces together.

To demonstrate the effectiveness of the approach even further, we also developed a system for the design of knitted toys.[13] A knitted toy is a toy made of knitted yarn instead of cloth patches. One can construct a knitted toy by knitting according to a specific knitting pattern, but it is difficult to produce an appropriate knitting pattern for a desired 3D shape. The Knitty system allows the user to design an original knitted toy by simply drawing the desired silhouette (Figure 22). The system then generates a 3D knitted animal model and corresponding knitting diagram. We ran a workshop using this system and observed that children could use it to design their own knitted animals.

Interactive 3D modeling assisted by concurrent physical simulation can be a powerful tool in many other application domains. For example, if one could run an aerodynamic simulation during the interactive design of a paper airplane model, the entire geometry could be adjusted in an intelligent manner in response to the user's simple deformation operations to produce a model that could actually fly. This kind of interaction would make it easier for designers to pursue aesthetic goals while satisfying engineering constraints. Real-time simulation does require high-performance computing resources, but some meaningful support should be possible by carefully limiting the target task and designing appropriate interfaces as shown in this paper. We hope that our work inspires more work in this direction.

**References**

1. Agrawala, M., Doantam, P., Heiser, J., Haymaker, J., Klingner, J., Hanrahan, P., Tversky, B. Designing effective step-by-step assembly instructions. *ACM Trans. Graph.* (In *Proceedings of SIGGRAPH 2003*), *22*, 3 (2003), 828–837.
2. Alexe, A., Barthe, L., Cani, M., Gaildrat, V. Shape modeling by sketching using convolution surfaces. In *Proceedings of Pacific Graphics 2005* (2005).
3. Bradley, D., Popa, T., Sheffer, A., Heidrich, W., Boubekeur, T. Markerless garment capture. *ACM Trans. Comp. Graph.* (In *Proceedings of SIGGRAPH 2008*), *27*, 3 (2007), 99.
4. Breen, D.E., House, D.H., Wozny, M.J. Predicting the drape of woven cloth using interacting particles. In *Proceedings of SIGGRAPH 1994* (1994), 365–372.
5. Choi, K.-J., Ko, H.-S. Stable but responsive cloth. *ACM Trans. Graph.* (In *Proceedings of SIGGRAPH 2002*), *21*, 3 (2002), 81–97.
6. Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A., Cani, M.-P. Virtual garments: A fully geometric approach for clothing design. *Comp. Graph. Forum*. (In *Eurographics 2006 Proceedings*), *25*, 3 (2006), 625–634.
7. Desbrun, M., Schroder, P., Barr, A. Interactive animation of structured deformable objects. In *Proceedings of Graphics Interface 1999* (1999), 1–8.
8. Grinspun, E., Krisl, P., Schroder, P. CHARMS: A simple framework for adaptive simulation. *ACM Trans. Graph. 21*, 3 (2002), 281–290.
9. Igarashi, T., Hughes, J.F. A suggestive interface for 3D drawing. *14th Annual Symposium on User Interface Software and Technology*. (In *ACM UIST 2001*) (2001), 173–181.
10. Igarashi, T., Hughes, J.F. Clothing manipulation. *15th Annual Symposium on User Interface Software and Technology*. (In *ACM UIST 2002*) (2002), 91–100.
11. Igarashi, T., Matsuoka, S., Tanaka, H. Teddy: A sketching interface for 3D freeform design. *ACM SIGGRAPH 1999* (1999), 409–416.
12. Igarashi, T., Moscovich, T., Hughes, J.F. As-rigid-as-possible shape manipulation. *ACM Trans. Comp. Graph.* (In *ACM SIGGRAPH 2005*), *24*, 3 (2005), 1134–1141.
13. Igarashi, Y., Igarashi, T., Suzuki, H. Knitty: 3D Modeling of Knitted Animals with a Production. *Eurographics 2008 Annex to the Conference Proceedings* (ISSN 1017–4656) (2008), 187–190.
14. Julius, D., Kraevoy, V., Sheffer, A. D-Charts: quasi developable mesh segmentation. *Comp. Graph. Forum*. (In *Proceedings of Eurographics 2005*), *24*, 3 (2005), 981–990.
15. Masry, M., Lipson, H. A sketch-based interface for iterative design and analysis of 3D objects. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces* (2005), 109–118.
16. Mitani, J., Suzuki, H. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph 23*, 3 (2004), 259–263.
17. Igarashi, Y., Igarashi, T. Pillow: Interactive flattening of a 3D model for plush toy design. Smart Graphics 2008. Lecture Notes in Computer Science (LNCS) 5166, Springer-Verlag, 1–7.
18. Mori, Y., Igarashi, T. Plushie: An interactive design system for plush toys. *ACM Trans. Graph.* (In *Proceedings of SIGGRAPH 2007*), *26*, 3 (2007), 45.
19. Nealen, A., Igarashi, T., Sorkine, O., Alexa, M. FiberMesh: Designing freeform surfaces with 3D curves, *ACM Trans. Comp. Graph.* (In *Proceedings of SIGGRAPH 2007*), *26*, 3 (2007), 41.
20. Owada, S., Nielsen, F., Nakazawa, K., Igarashi, T. A Sketching Interface for Modeling the Internal Structures of 3D Shapes, Smart Graphics 2003, *Lecture Notes in Computer Science (LNCS) 2733*, Springer-Verlag, 49–57.
21. Schmidt, R., Wyvill, B., Sousa, M.C., Jorge., J.A. ShapeShop: Sketch-based solid modeling with BlobTrees. *2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005), 53–62.
22. Shatz, I., Tal, A., Leifman, G. Paper craft models from meshes. The Visual Computer. *Int. J. Comp. Graph.* (In *Proceedings of Pacific Graphics 2006*), *22*, 9 (2006), 825–834.
23. Sheffer, A., Levy, B., Mogilnitsky, M., Bogomyakov, A. ABF++: Fast and robust angle based flattening. *ACM Trans. Graph. 24*, 2 (2005), 311–330.



**Yuki Igarashi**
The University of Tokyo

**Takeo Igarashi**
The University of Tokyo, JST/ERATO

DOI:10.1145/1610252.1610276

# Technical Perspective
# A Graphical Sense of Touch

By Pat Hanrahan

ONE OF THE major innovations in computing was the invention of the graphical user interface at MIT, SRI, and Xerox PARC. The combination of computer graphics hardware with a mouse and keyboard enabled a new class of highly interactive applications based on direct manipulation of on-screen displays.

It is interesting to reflect on the relative rate of advance of input and output technology since the very first systems. At the time, graphics hardware consisted of single-bit framebuffers outputting to black-and-white displays.

Moving forward, we now have flat-panel, high-definition, full-color displays driven by inexpensive, high-performance graphics chips capable of drawing three-dimensional virtual worlds in real time.

Graphics hardware draws tens of millions of polygons and tens of billions of pixels per second. In comparison, most personal computers ship with a mouse and keyboard similar to that used at Xerox PARC in the 1970s.

The lack of progress in input technology has caused computers to become sensory deprived. They can output incredible displays of information, but they receive almost no information from their surrounding environment.

Contrast this situation to a living organism. Most organisms have extraordinary abilities to sense their environment, but limited ability to display information (except by movement; a few animals like the chameleon and the cuttlefish can change their skin color). Perhaps this explains why we enjoy interacting with our pets more than with our computers.

Stuart Card, a Senior Research Fellow at Xerox PARC, has observed that one of the breakthrough ideas in the graphical user interface was to amplify input relative to output. One mechanism is to enable input to be on output. Examples include on-screen buttons and menus. By leveraging output technology, we augment limited input by providing context. Another strategy for enhancing input is to use pattern recognition to extract as much information as possible from the stream of sensed data.

Fortunately, this state of sensory deprivation is beginning to change.

The biggest recent development is the commercial emergence of multitouch displays. Traditional display input technology only returns a single X, Y position at a time. As a result, the user can only point to a single location at a time and, consequently, use only one finger or one hand at a time.

In a multitouch display, multiple points are sensed simultaneously. This allows the application to sense multiple fingers from both hands. This in turn makes it possible to recognize finger gestures or coordinated two-handed motion.

Successful commercial examples of multitouch displays include the Apple iPhone and the Microsoft Surface. The iPhone has a unique user interface that is enabled by an embedded multitouch display. To zoom into a map, the user simply moves their fingers apart. Beyond touch, the iPhone has several additional built-in sensory modalities, including a microphone, camera, accelerometer, and a GPS receiver and compass. Relative to a modern desktop computer, it is sensory rich and output poor.

**The following paper by a team from Microsoft Research introduces a very novel way to build a multitouch interface. They modify a flat-panel display to sense touch directly.**

The following paper by a team from Microsoft Research introduces a very novel way to build a multitouch interface—the ThinSight system. They modify a flat-panel display to sense touch directly. Previously, touch was sensed indirectly; for example, by mounting a camera to look at the surface of the display. The camera-based approaches require a fairly large space, have problems with occlusion, and are difficult to calibrate. In the ThinSight system, the LED backlight that drives the display is modified to include infrared LEDs and sensors interspersed amid the visible light emitters.

The display surface can both emit light and sense position. Distributing sensors throughout the display substrate yields a compact, efficient design.

This paper is important because it proposes an innovative design that addresses a long-standing problem. However, there is much more work to do in this area. The authors have added the sense of touch to the display, but like real touch sensors it has limited resolution and the sensed object must be in contact with the display. In Pierre Wellner's pioneering work on the digital desk, the computer sensed remote hand positions as well as the position, type, and content of objects on the desk. Unfortunately, Wellner's system involved bulky cameras and projectors.

Hopefully HCI researchers will expand on the innovative sensing strategy proposed in this paper. We want compact interactive devices that have rich sensory capabilities including touch, sight, and hearing. C

**Pat Hanrahan** is the CANON Professor of Computer Science and Electrical Engineering in the Computer Graphics Laboratory at Stanford University, Stanford, CA.

DOI:10.1145/1610252.1610277

# ThinSight: A Thin Form-Factor Interactive Surface Technology

By Shahram Izadi, Steve Hodges, Alex Butler, Darren West, Alban Rrustemi, Mike Molloy and William Buxton


## ABSTRACT

**ThinSight is a thin form-factor interactive surface technology based on optical sensors embedded inside a regular liquid crystal display (LCD). These augment the display with the ability to sense a variety of objects near the surface, including fingertips and hands, to enable multitouch interaction. Optical sensing also allows other physical items to be detected, allowing interactions using various tangible objects. A major advantage of ThinSight over existing camera and projector-based systems is its compact form-factor, making it easier to deploy in a variety of settings. We describe how the ThinSight hardware is embedded behind a regular LCD, allowing sensing without degradation of display capability, and illustrate the capabilities of our system through a number of proof-of-concept hardware prototypes and applications.**


## 1. INTRODUCTION

Touch input using a single point of contact with a display is a natural and established technique for human computer interaction. Research over the past decades,[3] and more recently products such as the iPhone and Microsoft Surface, have shown the novel and exciting interaction techniques and applications possible if multiple simultaneous touch points can be detected.

Various technologies have been proposed for multitouch sensing in this way, some of which extend to detection of physical objects in addition to fingertips. Systems based on optical sensing have proven to be particularly powerful in the richness of data captured and the flexibility they can provide. As yet, however, such optical systems have predominately been based on cameras and projectors and require a large optical path in front of or behind the display. This typically results in relatively bulky systems—something that can impact adoption in many real-world scenarios. While capacitive overlay technologies, such as those in the iPhone and the Dell XT Tablet PC, can support thin form-factor multitouch, they are limited to sensing only fingertips.

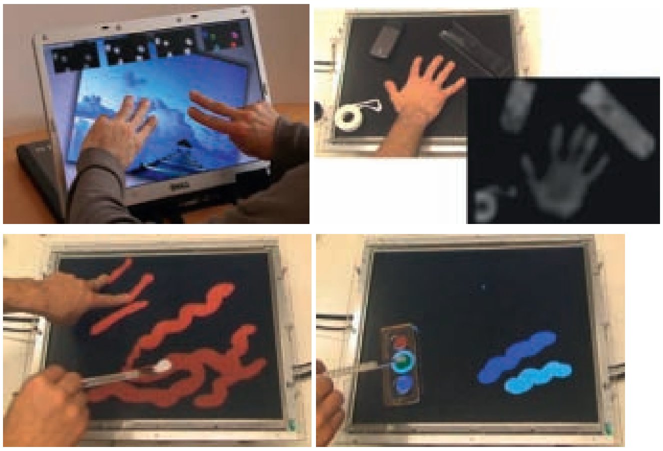
ThinSight is a novel interactive surface technology which is based on optical sensors integrated into a thin form-factor LCD. It is capable of imaging multiple fingertips, whole hands, and other objects near the display surface as shown in Figure 1. The system is based upon custom hardware embedded behind an LCD, and uses infrared (IR) light for sensing without degradation of display capability.

In this article we describe the ThinSight electronics and the modified LCD construction which results. We present two prototype systems we have developed: a multitouch laptop and a touch-and-tangible tabletop (both shown in Figure 1). These systems generate rich sensor data which can be processed using established computer vision techniques to prototype a wide range of interactive surface applications.

As shown in Figure 1, the shapes of many physical objects, including fingers, brushes, dials, and so forth, can be "seen" when they are near the display, allowing them to enhance multitouch interactions. Furthermore, ThinSight allows interactions close-up or at a distance using active IR pointing devices, such as styluses, and enables IR-based communication through the display with other electronic devices.

We believe that ThinSight provides a glimpse of a future where display technologies such as LCDs and organic light emitting diodes (OLEDs) will cheaply incorporate optical sensing pixels alongside red, green and blue (RGB) pixels in

**Figure 1. ThinSight brings the novel capabilities of surface computing to thin displays. Top left: photo manipulation using multiple fingers on a laptop prototype (note the screen has been reversed in the style of a Tablet PC). Top right: a hand, mobile phone, remote control and reel of tape placed on a tabletop ThinSight prototype, with corresponding sensor data far right. Note how all the objects are imaged through the display, potentially allowing not only multitouch but tangible input. Bottom left and right: an example of how such sensing can be used to support digital painting using multiple fingertips, a real brush and a tangible palette to change paint colors.**



Original versions of this paper appeared in *Proceedings of the 2007 ACM Symposium on User Interface Software and Technology* as "ThinSight: Versatile Multi-touch Sensing for Thin Form-factor Displays" and in *Proceedings of the 2008 IEEE Workshop on Horizontal Interactive Human Computer Systems* as "Experiences with Building a Thin Form-Factor Touch and Tangible Tabletop."

a similar manner, resulting in the widespread adoption of such surface technologies.

## 2. OVERVIEW OF OPERATION

### 2.1. Imaging through an LCD using IR light

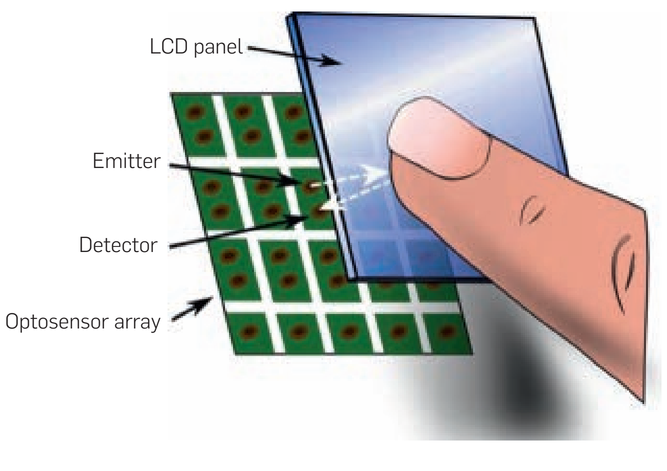
A key element in the construction of ThinSight is a device known as a retro-reflective optosensor. This is a sensing element which contains two components: a light emitter and an optically isolated light detector. It is therefore capable of both emitting light and, at the same time, detecting the intensity of incident light. If a reflective object is placed in front of the optosensor, some of the emitted light will be reflected back and will therefore be detected.

ThinSight is based around a 2D grid of retro-reflective optosensors which are placed behind an LCD panel. Each optosensor emits light that passes right through the entire panel. Any reflective object in front of the display (such as a fingertip) will reflect a fraction of the light back, and this can be detected. Figure 2 depicts this arrangement. By using a suitably spaced grid of retro-reflective optosensors distributed uniformly behind the display it is therefore possible to detect any number of fingertips on the display surface. The raw data generated is essentially a low resolution grayscale "image" of what can be seen through the display, which can be processed using computer vision techniques to support touch and other input.

A critical aspect of ThinSight is the use of retro-reflective sensors that operate in the infrared part of the spectrum, for three main reasons:

- Although IR light is attenuated by the layers in the LCD panel, some still passes through the display.[5] This is largely unaffected by the displayed image.
- A human fingertip typically reflects around 20% of incident IR light and is therefore a quite passable "reflective object."
- IR light is not visible to the user, and therefore does not detract from the image being displayed on the panel.

**Figure 2. The basic principle of ThinSight. An array of retro-reflective optosensors is placed behind an LCD. Each of these contains two elements: an emitter which shines IR light through the panel; and a detector which picks up any light reflected by objects such as fingertips in front of the screen.**




### 2.2. Further features of ThinSight

ThinSight is not limited to detecting fingertips in contact with the display; any suitably reflective object will cause IR light to reflect back and will therefore generate a "silhouette." Not only can this be used to determine the location of the object on the display, but also its orientation and shape, within the limits of sensing resolution. Furthermore, the underside of an object may be augmented with a visual mark—a barcode of sorts—to aid identification.

In addition to the detection of passive objects via their shape or some kind of barcode, it is also possible to embed a very small infrared transmitter into an object. In this way, the object can transmit a code representing its identity, its state, or some other information, and this data transmission can be picked up by the IR detectors built into ThinSight. Indeed, ThinSight naturally supports bidirectional IR-based data transfer with nearby electronic devices such as smartphones and PDAs. Data can be transmitted from the display to a device by modulating the IR light emitted. With a large display, it is possible to support several simultaneous bidirectional communication channels in a spatially multiplexed fashion.

Finally, a device which emits a collimated beam of IR light may be used as a pointing device, either close to the display surface like a stylus, or from some distance. Such a pointing device could be used to support gestures for new forms of interaction with a single display or with multiple displays. Multiple pointing devices could be differentiated by modulating the light generated by each.

## 3. THE THINSIGHT HARDWARE

### 3.1. The sensing electronics

The prototype ThinSight circuit board depicted in Figure 3 uses Avago HSDL-9100 retro-reflective infrared sensors. These devices are especially designed for proximity sensing —an IR LED emits infrared light and an IR photodiode generates a photocurrent which varies with the amount of incident light. Both emitter and detector have a center wavelength of 940 nm.

A $7 \times 5$ grid of these HSDL-9100 devices on a regular 10 mm pitch is mounted on custom-made $70 \times 50$ mm 4-layer printed circuit board (PCB). Multiple PCBs can be tiled together to support larger sensing areas. The IR detectors are interfaced directly with digital input/output lines on a PIC18LF4520 microcontroller.

The PIC firmware collects data from one row of detectors at a time to construct a "frame" of data which is then transmitted to the PC over USB via a virtual COM port. To connect multiple PCBs to the same PC, they must be synchronized to ensure that IR emitted by a row of devices on one PCB does not adversely affect scanning on a neighboring PCB. In our prototype we achieve this using frame and row synchronization signals which are generated by one of the PCBs (the designated "master") and detected by the others ("slaves").

Note that more information on the hardware can be found in the full research publications.[7, 10]

Figure 3. Top: the front side of the sensor PCB showing the 7×5 array of IR optosensors. The transistors that enable each detector are visible to the right of each optosensor. Bottom: the back of the sensor PCB has little more than a PIC microcontroller, a USB interface and FETs to drive the rows and columns of IR emitting LEDs. Three such PCBs are used in our ThinSight laptop while there are thirty in the tabletop prototype.

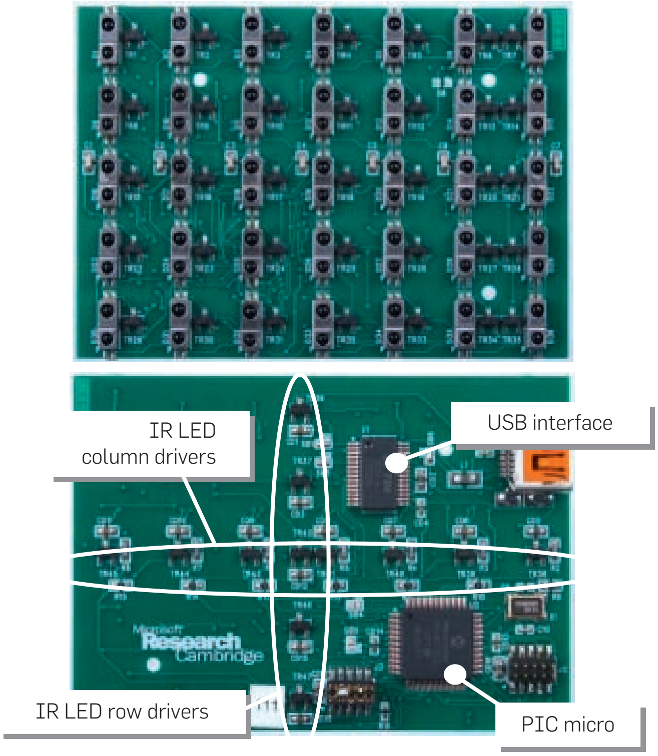
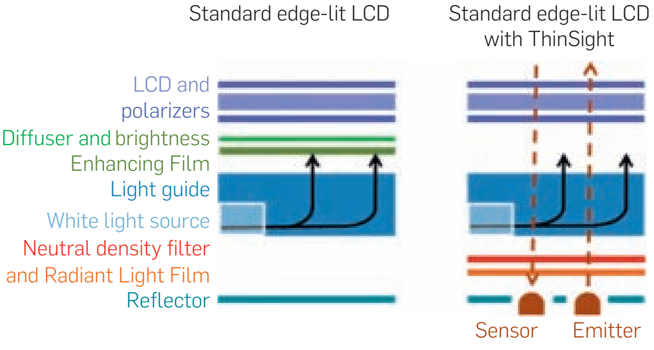



Figure 4. Typical LCD edge-lit architecture shown left. The LCD comprises a stack of optical elements. A white light source is typically located along one or two edges at the back of the panel. A white reflector and transparent light guide direct the light toward the front of the panel. The films help scatter this light uniformly and enhance brightness. However, they also cause excessive attenuation of IR light. In ThinSight, shown right, the films are substituted and placed behind the light guide to minimize attenuation and also reduce noise caused by LCD flexing upon touch. The sensors and emitters are placed at the bottom of the resulting stack, aligned with holes cut in the reflector.




### 3.2. LCD technology overview

To understand how the ThinSight hardware is integrated into a display panel, it is useful to understand the construction and operation of a typical LCD. An LCD panel is made up of a stack of optical components as shown in Figure 4. At the front of the panel is a thin layer of liquid crystal material which is sandwiched between two polarizers. The polarizers are orthogonal to each other, which means that any light which passes through the first will naturally be blocked by the second, resulting in dark pixels. However, if a voltage is applied across the liquid crystal material at a certain pixel location, the polarization of light incident on that pixel is twisted through 90° as it passes through the crystal structure. As a result it emerges from the crystal with the correct polarization to pass through the second polarizer. Typically, white light is shone through the panel from behind by a backlight and red, green, and blue filters are used to create a color display. In order to achieve a low profile construction while maintaining uniform lighting across the entire display and keeping cost down, the backlight is often a large "light guide" in the form of a clear acrylic sheet which sits behind the entire LCD and which is edge-lit from one or more sides. The light source is often a cold cathode fluorescent tube or an array of white LEDs. To maximize the efficiency and uniformity of the lighting, additional layers of material may be placed between the light guide and the LCD. Brightness enhancing film (BEF) "recycles" visible light at suboptimal angles and polarizations and a diffuser smoothes out any local nonuniformities in light intensity.

### 3.3. Integration with an LCD panel

We constructed our ThinSight prototypes using a variety of desktop and laptop LCD panels, ranging from 17" to 21". Two of these are shown in Figures 5 and 6. Up to 30 PCBs were tiled to support sensing across the entire surface. In instances where large numbers of PCBs were tiled, a custom hub circuit based on an FPGA was designed to collect and aggregate the raw data captured from a number of tiled sensors and transfer this to the PC using a single USB channel. These tiled PCBs are mounted directly behind the light guide. To ensure that the cold cathode does not cause any stray IR light to emanate from the acrylic light guide, we placed a narrow piece of IR-blocking film between it and the backlight. We cut small holes in the white reflector behind the light guide to coincide with the location of every IR emitting and detecting element.

During our experiments we found that the combination of the diffuser and BEF in an LCD panel typically caused excessive attenuation of the IR signal. However, removing these materials degrades the displayed image significantly: without BEF the brightness and contrast of the displayed image is reduced unacceptably; without a diffuser the image appears to "float" in front of the backlight and at the same time the position of the IR emitters and detectors can be seen in the form of an array of faint dots across the entire display.

To completely hide the IR emitters and detectors we required a material that lets IR pass through it but not visible light, so that the optosensors could not be seen but would operate normally. The traditional solution would be

Figure 5. Our laptop prototype. Top: Three PCBs are tiled together and mounted on an acrylic plate, to give a total of 105 sensing pixels. Holes are also cut in the white reflector shown on the far left. Bottom left: an aperture is cut in the laptop lid to allow the PCBs to be mounted behind the LCD. This provides sensing across the center of the laptop screen. Bottom right: side views of the prototype—note the display has been reversed on its hinges in the style of a Tablet PC.

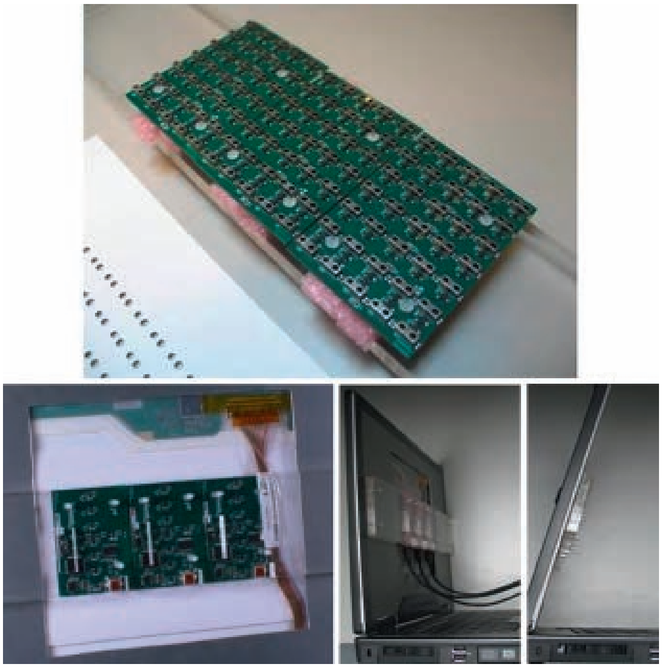
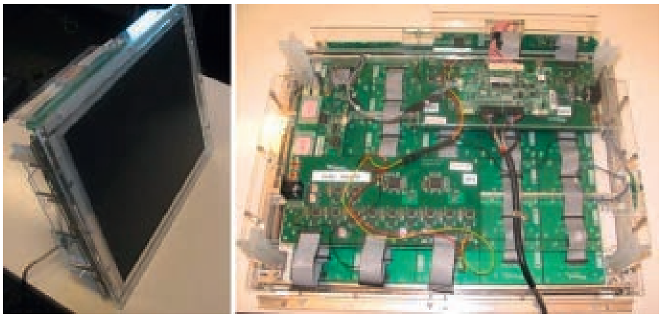


Figure 6. The ThinSight tabletop hardware as viewed from the side and behind. Thirty PCBs (in a 5×6 grid) are tiled with columns interconnected with ribbon cable and attached to a hub board for aggregating data and inter-tile communication. This provides a total of 1050 discrete sensing pixels across the entire surface.



to use what is referred to as a "cold mirror." Unfortunately these are made using a glass substrate which means they are expensive, rigid and fragile and we were unable to source a cold mirror large enough to cover the entire tabletop display. We experimented with many alternative materials including tracing paper, acetate sheets coated in emulsion paint, spray-on frosting, thin sheets of white polythene and mylar. Most of these are unsuitable either because of a lack of IR transparency or because the optosensors can be seen through them to some extent. The solution we settled on was the use of Radiant Light Film by 3M (part number CM500), which largely lets IR light pass through while reflecting visible light without the disadvantages of a true cold mirror. This was combined with the use of a grade "0" neutral density filter, a visually opaque but IR transparent diffuser, to even out the distribution rear illumination and at the same time prevent the "floating" effect. Applying the Radiant Light Film carefully is critical since minor imperfections (e.g. wrinkles or bubbles) are highly visible to the user—thus we laminated it onto a thin PET carrier. One final modification to the LCD construction was to deploy these films *behind* the light guide to further improve the optical properties. The resulting LCD layer stack-up is depicted in Figure 4 right.

Most LCD panels are not constructed to resist physical pressure, and any distortion which results from touch interactions typically causes internal IR reflection resulting in "flare." Placing the Radiant Light Film and neutral density filter behind the light guide improves this situation, and we also reinforced the ThinSight unit using several lengths of extruded aluminum section running directly behind the LCD.

## 4. THINSIGHT IN OPERATION

### 4.1. Processing the raw sensor data

Each value read from an individual IR detector is defined as an integer representing the intensity of incident light. These sensor values are streamed to the PC via USB where the raw data undergoes several simple processing and filtering steps in order to generate an IR image that can be used to detect objects near the surface. Once this image is generated, established image processing techniques can be applied in order to determine coordinates of fingers, recognize hand gestures, and identify object shapes.

Variations between optosensors due to manufacturing and assembly tolerances result in a range of different values across the display even without the presence of objects on the display surface. To make the sensor image uniform and the presence of additional incident light (reflected from nearby objects) more apparent, we subtract a "background" frame captured when no objects are present, and normalize relative to the image generated when the display is covered with a sheet of white reflective paper.

We use standard bicubic interpolation to scale up the sensor image by a predefined factor (10 in our current implementation). For the larger tabletop implementation this results in a $350 \times 300$ pixel image. Optionally, a Gaussian filter can be applied for further smoothing, resulting in a grayscale "depth" image as shown in Figure 7.

### 4.2. Seeing through the ThinSight display

The images we obtain from the prototype are quite rich, particularly given the density of the sensor array. Fingers and hands within proximity of the screen are clearly identifiable. Examples of images captured through the display are shown in Figures 1, 7 and 8.

Figure 7. The raw ThinSight sensor data shown left and after interpolation and smoothing right. Note that the raw image is a very low resolution, but contains enough data to generate the relatively rich image at right.

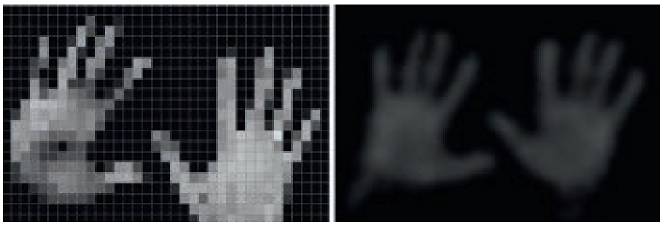
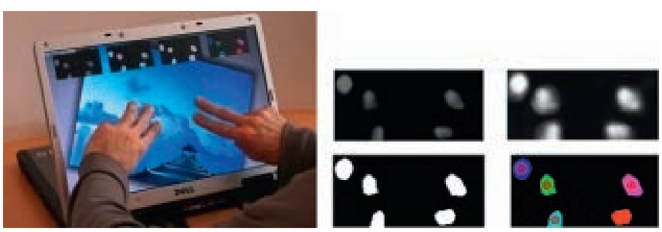


Figure 8. Fingertips can be sensed easily with ThinSight. Left: the user places five fingers on the display to manipulate a photo. Right: a close-up of the sensor data when fingers are positioned as shown at left. The raw sensor data is: (1) scaled-up with interpolation, (2) normalized, (3) thresholded to produce a binary image, and finally (4) processed using connected components analysis to reveal the fingertip locations.



Fingertips appear as small blobs in the image as they approach the surface, increasing in intensity as they get closer. This gives rise to the possibility of sensing both touch and hover. To date we have only implemented touch/no-touch differentiation, using thresholding. However, we can reliably and consistently detect touch to within a few millimeters for a variety of skin tones, so we believe that disambiguating hover from touch would be possible.

In addition to fingers and hands, optical sensing allows us to observe other IR reflective objects through the display. Figure 1 illustrates how the display can distinguish the shape of many reflective objects in front of the surface, including an entire hand, mobile phone, remote control, and a reel of white tape. We have found in practice that many objects reflect IR.

A logical next step is to attempt to uniquely identify objects by placement of visual codes underneath them. Such codes have been used effectively in tabletop systems such as the Microsoft Surface and various research prototypes[12, 28] to support tangible interaction. We have also started preliminary experiments with the use of such codes on ThinSight, see Figure 9.

Active electronic identification schemes are also feasible. For example, cheap and small dedicated electronic units containing an IR emitter can be stuck onto or embedded inside objects that need to be identified. These emitters will produce a signal directed to a small subset of the display sensors. By emitting modulated IR it is possible to transmit a unique identifier to the display.

### 4.3. Communicating through the ThinSight display

Beyond simple identification, an embedded IR transmitter also provides a basis for supporting richer bidirectional communication with the display. In theory any IR modulation scheme, such as the widely adopted IrDA standard, could be supported by ThinSight. We have implemented a DC-balanced modulation scheme which allows retro-reflective object sensing to occur *at the same time* as data transmission. This required no additions or alterations to the sensor PCB, only changes to the microcontroller firmware. To demonstrate our prototype implementation of this, we built a small embedded IR transceiver based on a low power MSP430 microcontroller, see Figure 10. We encode 3 bits of data in the IR transmitted from the ThinSight pixels to control an RGB LED fitted to the embedded receiver. When the user touches various soft buttons on the ThinSight display, this in turn transmits different 3 bit codes from ThinSight pixels to cause different colors on the embedded device to be activated.

It is theoretically possible to transmit and receive different data simultaneously using different columns on the

Figure 9. An example 2" diameter visual marker and the resulting ThinSight image after processing.

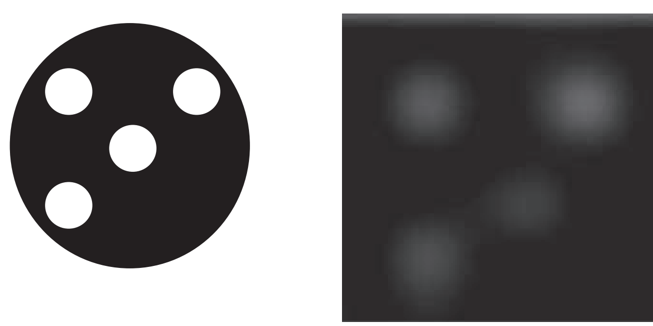
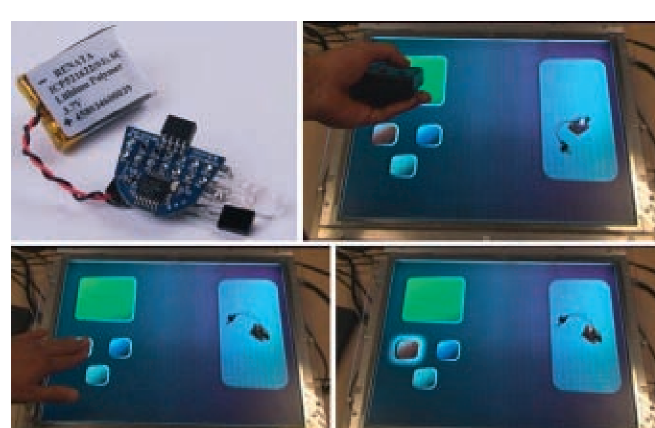


Figure 10. Using ThinSight to communicate with devices using IR. Top left: an embedded microcontroller/IR transceiver/RGB LED device. Bottom left: touching a soft button on the ThinSight display signals the RGB LED on the embedded device to turn red (bottom right). Top right: A remote control is used to signal from a distance the display whichin turn sends an IR command to the RGB device to turn the LED blue.



display surface, thereby supporting spatially multiplexed bidirectional communications with multiple local devices and reception of data from remote gesturing devices. Of course, it is also possible to time multiplex communications between different devices if a suitable addressing scheme is used. We have not yet prototyped either of these multiple-device communications schemes.

### 4.4. Interacting with ThinSight

As shown earlier in this section, it is straightforward to sense and locate multiple fingertips using ThinSight. In order to do this we threshold the processed data to produce a binary image. The connected components within this are isolated, and the center of mass of each component is calculated to generate representative $X$, $Y$ coordinates of each finger. A very simple homography can then be applied to map these fingertip positions (which are relative to the sensor image) to onscreen coordinates. Major and minor axis analysis or more detailed shape analysis can be performed to determine orientation information. Robust fingertip tracking algorithms or optical flow techniques[28] can be employed to add stronger heuristics for recognizing gestures.

Using these established techniques, fingertips are sensed to within a few millimeters, currently at 23 frames/s. Both hover and touch can be detected, and could be disambiguated by defining appropriate thresholds. A user therefore need not apply any force to interact with the display. However, it is also possible to estimate fingertip pressure by calculating the increase in the area and intensity of the fingertip "blob" once touch has been detected.

Figure 1 shows two simple applications developed using ThinSight. A simple photo application allows multiple images to be translated, rotated, and scaled using established multifinger manipulation gestures. We use distance and angle between touch points to compute scale factor and rotation deltas. To demonstrate some of the capabilities of ThinSight beyond just multitouch, we have built an example paint application that allows users to paint directly on the surface using both fingertips and real paint brushes. The latter works because ThinSight can detect the brushes' white bristles which reflect IR. The paint application also supports a more sophisticated scenario where an artist's palette is placed on the display surface. Although this is visibly transparent, it has an IR reflective marker on the underside which allows it to be detected by ThinSight, whereupon a range of paint colors are rendered underneath it. The user can change color by "dipping" either a fingertip or a brush into the appropriate well in the palette. We identify the presence of this object using a simple ellipse matching algorithm which distinguishes the larger palette from smaller touch point "blobs" in the sensor image. Despite the limited resolution of ThinSight, it is possible to differentiate a number of different objects using simple silhouette shape information.

## 5. DISCUSSION AND FUTURE WORK

We believe that the prototype presented in this article is an interesting proof-of-concept of a new approach to multitouch and tangible sensing for thin displays. We have already described some of its potential; here we discuss a number of additional observations and ideas which came to light during the work.

### 5.1. Fidelity of sensing

The original aim of this project was simply to detect fingertips to enable multi-touch-based direct manipulation. However, despite the low resolution of the raw sensor data, we still detect quite sophisticated object images. Very small objects do currently "disappear" on occasion when they are midway between optosensors. However, we have a number of ideas for improving the fidelity further, both to support smaller objects and to make object and visual marker identification more practical. An obvious solution is to increase the density of the optosensors, or at least the density of IR detectors. Another idea is to measure the amount of reflected light under different lighting conditions—for example, simultaneously emitting light from neighboring sensors is likely to cause enough reflection to detect smaller objects.

### 5.2. Frame rate

In informal trials of ThinSight for a direct manipulation task, we found that the current frame rate was reasonably acceptable to users. However, a higher frame rate would not only produce a more responsive UI which will be important for some applications, but would make temporal filtering more practical thereby reducing noise and improving subpixel accuracy. It would also be possible to sample each detector under a number of different illumination conditions as described above, which we believe would increase fidelity of operation.

### 5.3. Robustness to lighting conditions

The retro-reflective nature of operation of ThinSight combined with the use of background substitution seems to give reliable operation in a variety of lighting conditions, including an office environment with some ambient sunlight. One common approach to mitigating any negative effects of ambient light, which we could explore if necessary, is to emit modulated IR and to ignore any nonmodulated offset in the detected signal.

### 5.4. Power consumption

The biggest contributor to power consumption in ThinSight is emission of IR light; because the signal is attenuated in both directions as it passes through the layers of the LCD panel, a high intensity emission is required. For mobile devices, where power consumption is an issue, we have ideas for improvements. We believe it is possible to enhance the IR transmission properties of an LCD panel by optimizing the materials used in its construction for this purpose—something which is not currently done. In addition, it may be possible to keep track of object and fingertip positions, and limit the most frequent IR emissions to those areas. The rest of the display would be scanned less frequently (e.g. at 2–3 frames/s) to detect new touch points.

One of the main ways we feel we can improve on power consumption and fidelity of sensing is to use a more sophisticated IR illumination scheme. We have been experimenting with using an acrylic overlay *on top* of the LCD and using IR LEDs for edge illumination. This would allow us to sense multiple touch points using standard Frustrated Total Internal Reflection (FTIR),[5] but not objects. We have, however, also experimented with a material called Endlighten which allows this FTIR scheme to be extended to diffuse illumination, allowing both multitouch and object sensing with far fewer IR emitters than our current setup. The overlay can also serve the dual purpose of protecting the LCD from flexing under touch.

## 6. RELATED WORK

The area of interactive surfaces has gained particular attention recently following the advent of the iPhone and Microsoft Surface. However, it is a field with over two decades of history.[3] Despite this sustained interest there has been an evident lack of off-the-shelf solutions for detecting multiple fingers and/or objects on a display surface. Here, we summarize the relevant research in these areas and describe the few commercially available systems.

### 6.1. Camera-based systems

One approach to detecting multitouch and tangible input is to use a video camera placed in front of or above the surface, and apply computer vision algorithms for sensing. Early seminal work includes Krueger's VideoDesk[13] and the DigitalDesk,[26] which use dwell time and a microphone (respectively) to detect when a user is actually touching the surface. More recently, the Visual Touchpad[17] and C-Slate[9] use a stereo camera placed above the display to more accurately detect touch. The disparity between the image pairs determines the height of fingers above the surface. PlayAnywhere[28] introduces a number of additional image processing techniques for front-projected vision-based systems, including a shadow-based touch detection algorithm, a novel visual bar code scheme, paper tracking, and an optical flow algorithm for bimanual interaction.

Camera-based systems such as those described above obviously require direct line-of-sight to the objects being sensed which in some cases can restrict usage scenarios. Occlusion problems are mitigated in PlayAnywhere by mounting the camera off-axis. A natural progression is to mount the camera *behind* the display. HoloWall[18] uses IR illuminant and a camera equipped with an IR pass filter behind a diffusive projection panel to detect hands and other IR-reflective objects in front of it. The system can accurately determine the contact areas by simply thresholding the infrared image. TouchLight[27] uses rear-projection onto a holographic screen, which is also illuminated from behind with IR light. A number of multitouch application scenarios are enabled including high-resolution imaging capabilities. Han[5] describes a straightforward yet powerful technique for enabling high-resolution multitouch sensing on rear-projected surfaces based on FTIR. Compelling multitouch applications have been demonstrated using this technique. The Smart Table[22] uses this same FTIR technique in a tabletop form factor.

The Microsoft Surface and ReacTable[12] also use rear-projection, IR illuminant and a rear mounted IR camera to monitor fingertips, this time in a horizontal tabletop form-factor. These systems also detect and identify objects with IR-reflective markers on their surface.

The rich data generated by camera-based systems provides extreme flexibility. However, as Wilson discusses[28] this flexibility comes at a cost, including the computational demands of processing high resolution images, susceptibility to adverse lighting conditions and problems of motion blur. However, perhaps more importantly, these systems require the camera to be placed at some distance from the display to capture the entire scene, limiting their portability, practicality and introducing a setup and calibration cost.

### 6.2. Opaque embedded sensing

Despite the power of camera-based systems, the associated drawbacks outlined above have resulted in a number of parallel research efforts to develop a non-vision-based multitouch display. One approach is to embed a multitouch sensor of some kind behind a surface that can have an image projected onto it. A natural technology for this is capacitive sensing, where the capacitive coupling to ground introduced by a fingertip is detected, typically by monitoring the rate of leakage of charge away from conductive plates or wires mounted behind the display surface.

Some manufacturers such as Logitech and Apple have enhanced the standard laptop-style touch pad to detect certain gestures based on more than one point of touch. However, in these systems, using more than two or three fingers typically results in ambiguities in the sensed data. This constrains the gestures they support. Lee et al.[14] used capacitive sensing with a number of discrete metal electrodes arranged in a matrix configuration to support multitouch over a larger area. Westerman[25] describes a sophisticated capacitive multitouch system which generates x-ray-like images of a hand interacting with an opaque sensing surface, which could be projected onto. A derivative of this work was commercialized by Fingerworks.

DiamondTouch[4] is composed of a grid of row and column antennas which emit signals that capacitively couple with users when they touch the surface. Users are also capacitively coupled to receivers through pads on their chairs. In this way the system can identify which antennas behind the display surface are being touched and by which user, although a user touching the surface at two points can produce ambiguities. The SmartSkin[21] system consists of a grid of capacitively coupled transmitting and receiving antennas. As a finger approaches an intersection point, this causes a drop in coupling which is measured to determine finger proximity. The system is capable of supporting multiple points of contact by the same user and generating images of contact regions of the hand. SmartSkin and DiamondTouch also support physical objects, but can only identify an object when a user touches it. Tactex provide another interesting example of an opaque multitouch sensor, which uses transducers to measure surface pressure at multiple touch points.[23]

### 6.3. Transparent overlays

The systems above share one major disadvantage: they all rely on front-projection for display. The displayed image will therefore be broken up by the user's fingers, hands and arms, which can degrade the user experience. Also, a large throw distance is typically required for projection which limits portability. Furthermore, physical objects can only be detected in limited ways, if object detection is supported at all.

One alternative approach to address some of the issues of display and portability is to use a transparent sensing overlay in conjunction with a self-contained (i.e., not projected) display such as an LCD panel. DualTouch[19] uses an off-the-shelf transparent resistive touch overlay to detect the position of two fingers. Such overlays typically report the average position when two fingers are touching. Assuming that one finger makes contact first and does not subsequently move, the position of a second touch point can be calculated. An extension to this is provided by Loviscach.[16]

The Philips Entertaible[15] takes a different "overlay" approach to detect up to 30 touch points. IR emitters and detectors are placed on a bezel around the screen. Breaks in the IR beams detect fingers and objects. The SMART DViT[22] and HP TouchSmart[6] utilize cameras in the corners of a bezel overlay to support sensing of two fingers or styluses. With such line of sight systems, occlusion can be an issue for sensing.

The Lemur music controller from JazzMutant[11] uses a proprietary resistive overlay technology to track up to 20 touch points simultaneously. More recently, Balda AG and N-Trig[20] have both released capacitive multitouch overlays, which have been used in the iPhone and the Dell XT, respectively. These approaches provide a robust way for sensing multiple fingers touching the surface, but do not scale to whole hand sensing or tangible objects.

### 6.4. The need for intrinsically integrated sensing

The previous sections have presented a number of multitouch display technologies. Camera-based systems produce very rich data but have a number of drawbacks. Opaque sensing systems can more accurately detect fingers and objects, but by their nature rely on projection. Transparent overlays alleviate this projection requirement, but the fidelity of sensing is reduced. It is difficult, for example, to support sensing of fingertips, hands and objects.

A potential solution which addresses all of these requirements is a class of technologies that we refer to as "intrinsically integrated" sensing. The common approach behind these is to distribute sensing across the display surface, integrating the sensors with the display elements. Hudson[8] reports on a prototype 0.7" monochrome display where LED pixels double up as light sensors. By operating one pixel as a sensor while its neighbors are illuminated, it is possible to detect light reflected from a fingertip close to the display. The main drawbacks are the use of visible illuminant during sensing and practicalities of using LED-based displays. SensoLED uses a similar approach with visible light, but this time based on polymer LEDs and photodiodes. A 1" diagonal sensing polymer display has been demonstrated.[2]

Planar[1] and Toshiba[24] were among the first to develop LCD prototypes with integrated visible light photosensors, which can detect the shadows resulting from fingertips or styluses on the display. The photosensors and associated signal processing circuitry are integrated directly onto the LCD substrate. To illuminate fingers and other objects, either an external light source is required—impacting on the profile of the system—or the screen must uniformly emit bright visible light—which in turn will disrupt the displayed image.

The motivation for ThinSight was to build on the concept of intrinsically integrated sensing. We have extended the work above using invisible (IR) illuminant to allow simultaneous display and sensing, building on current LCD and IR technologies to make prototyping practical in the near term. Another important aspect is support for much larger thin touch-sensitive displays than is provided by intrinsically integrated solutions to date, thereby making it more practical to prototype multitouch applications.

## 7. CONCLUSION

In this article we have described a new technique for optically sensing multiple objects, including fingertips, through thin form-factor displays. Optical sensing allows rich "camera-like" data to be captured by the display and this is processed using computer vision techniques. This supports new types of human computer interfaces that exploit zero-force multi-touch and tangible interaction on thin form-factor displays such as those described in Buxton.[3] We have shown how this technique can be integrated with off-the-shelf LCD technology, making such interaction techniques more practical and deployable in real-world settings.

We have many ideas for potential refinements to the ThinSight hardware, firmware, and PC software. In addition to such incremental improvements, we also believe that it will be possible to transition to an integrated "sensing and display" solution which will be much more straightforward and cheaper to manufacture. An obvious approach is to incorporate optical sensors directly onto the LCD backplane, and as reported earlier early prototypes in this area are beginning to emerge.[24] Alternatively, polymer photodiodes may be combined on the same substrate as polymer OLEDs[2] for a similar result. The big advantage of this approach is that an array of sensing elements can be combined with a display at very little incremental cost by simply adding "pixels that sense" in between the visible RGB display pixels. This would essentially augment a display with optical multitouch input "for free," enabling truly widespread adoption of this exciting technology.

### Acknowledgments

We thank Stuart Taylor, Steve Bathiche, Andy Wilson, Turner Whitted and Otmar Hilliges for their invaluable input.

**References**

1. Abileah, A., Green, P. Optical sensors embedded within AMLCD panel: design and applications. In *Proceedings of EDT'07* (San Diego, California, August 04, 2007), ACM, New York, NY.
2. Bürgi, L. et al. Optical proximity and touch sensors based on monolithically integrated polymer photodiodes and polymer LEDs. *Org. Electron. 7* (2006).
3. Buxton, B. Multi-Touch Systems That I Have Known and Loved (2007), http://www.billbuxton.com/multitouchOverview.html.
4. Dietz, P., Leigh, D. DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, FL, Nov. 11–14, 2001), UIST '01. ACM, NY, 219–226.
5. Han, J.Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, Oct. 23–26, 2005), UIST '05. ACM, NY, 115–118.
6. HP TouchSmart, http://www.hp.com/united-states/campaigns/touchsmart/.
7. Hodges, S., Izadi, S., Butler, A., Rrustemi, A., Buxton, B. ThinSight: Versatile multi-touch sensing for thin form-factor displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, RI, Oct. 07–10, 2007), UIST '07. ACM, NY, 259–268.
8. Hudson, S.E. 2004. Using light emitting diode arrays as touch-sensitive input and output devices. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, Oct. 24–27, 2004), UIST '04. ACM, NY, 287–290.
9. Izadi, S. et al. C-Slate: A multi-touch and object recognition system for remote collaboration using horizontal surfaces. In *IEEE Workshop on Horizontal Interactive Human-Computer Systems* (Rhode Island, Oct. 2007), IEEE Tabletop 2007, IEEE, 3–10.
10. Izadi, S. et al. Experiences with building a thin form-factor touch and tangible tabletop. In *IEEE Workshop on Horizontal Interactive Human-Computer Systems* (Amsterdam, Holland, Oct. 2008), Tabletop 2008, IEEE, 181–184.
11. JazzMutant Lemur. http://www.jazzmutant.com/lemur_overview.php.
12. Jordà, S., Geiger, G., Alonso, M., Kaltenbrunner, M. The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, LA, Feb. 15–17, 2007). TEI '07. ACM, NY, 139–146.
13. Krueger, M. Artificial Reality 2, Addison-Wesley Professional (1991). ISBN 0-201-52260-8
14. Lee, S., Buxton, W., Smith, K.C. 5. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bull 16*, 4 (Apr. 1985), 21–25.
15. van Loenen, E. et al. Entertaible: a solution for social gaming experiences. In *Tangible Play Workshop, IUI Conference* (2007).
16. Loviscach, J. Two-finger input with a standard touch screen. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, RI, USA, October 07–10, 2007). UIST '07. ACM, New York, NY, 169–172.
17. Malik, S., Laszlo, J. Visual touchpad: A two-handed gestural input device. In *Proceedings of the 6th International Conference on Multimodal Interfaces* (State College, PA, Oct. 13–15, 2004), ICMI '04. ACM, NY, 289–296.
18. Matsushita, N., Rekimoto, J. 1997. HoloWall: Designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (Banff, Alberta, Canada, Oct. 14–17, 1997), UIST '97. ACM, NY, 209–210.
19. Matsushita, N., Ayatsuka, Y., Rekimoto, J. 2000. Dual touch: A two-handed interface for pen-based PDAs. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (San Diego, California, US, Nov. 06–08, 2000), UIST '00. ACM, New York, NY, 211–212.
20. N-trig, duo-touch sensor, http://www.n-trig.com/.
21. Rekimoto, J. SmartSkin: An infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, MN, Apr. 20–25, 2002), CHI '02. ACM, NY, 113–120.
22. Smart Technologies, http://smarttech.com (2009).
23. Tactex Controls Inc. Array Sensors. http://www.tactex.com/products_array.php.
24. Matsushita, T. LCD with Finger Shadow Sensing, http://www3.toshiba.co.jp/tm_dsp/press/2005/05-09-29.htm.
25. Westerman, W. Hand Tracking, Finger Identification and Chordic Manipulation on a Multi-Touch Surface. PhD thesis, University of Delaware (1999).
26. Wellner, P. Interacting with Paper on the Digital Desk. *CACM 36*, 7 (1993) 86–96.
27. Wilson, A.D. TouchLight: An imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th International Conference on Multimodal Interfaces* (State College, PA, Oct. 13–15, 2004), ICMI '04. ACM, NY, 69–76.
28. Wilson, A.D. PlayAnywhere: A compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, Oct. 23–26, 2005), UIST '05. ACM, NY, 83–92.


**Shahram Izadi, Steve Hodges, Alex Butler, Darren West, Alban Rrustemi, Mike Molloy, and William Buxton** ({shahrami, shodges, dab}@microsoft.com), Microsoft Research Cambridge, U.K.

### Boise State University
**Assistant Professor of Computer Science**

The Department of Computer Science at Boise State University is seeking a tenure-track faculty member at the assistant-professor level. Candidates specializing in the areas of databases, software engineering, and visualization are especially encouraged to apply, Additional details are found online at: http://coen.boisestate.edu/cs/Opportunities.asp

EEO/AA Institution, Veterans preference.

### Boston University
**Department of Electrical & Computer Engineering Faculty Positions**

The Department of Electrical & Computer Engineering (ECE) at Boston University (BU) is seeking candidates for several anticipated faculty positions at all ranks in signals and systems, communications, computer systems and networks, photonics, solid-state devices and nanostructures, and related areas. The Department is seeking to foster growth in the broad, interdisciplinary topics of energy, health, and communications; candidates with research interests that transcend the traditional boundaries of ECE are strongly encouraged to apply.

ECE at BU is a world-class, progressive department with excellent resources that is steadily gaining national and international prominence for its exceptional research and education record. ECE is part of BU's rapidly growing and innovative College of Engineering, and currently consists of approximately 45 faculty members, 200 graduate students, and 200 undergraduates. Outstanding collaboration opportunities are available with nationally recognized medical centers and universities/colleges, nearby research centers, and industry throughout the Boston area. Joint appointments with other BU departments and with the newly formed Divisions of Materials Science & Engineering and Systems Engineering are also possible.

Qualified candidates must possess a relevant, earned PhD, and have a demonstrable ability to teach effectively and develop funded programs in their area of expertise that will contribute to the tradition of excellence in research characteristic of the ECE Department. Self-motivated individuals who thrive on challenge and are eager to utilize their expertise to strengthen an ambitious program of departmental growth are desired. Women, minorities, and candidates from other underrepresented groups are especially encouraged to apply.

Beyond its research and academic activities, BU has a lively, urban campus situated along the banks of the Charles River in Boston's historic Fenway-Kenmore neighborhood. The campus and surrounding areas offer limitless opportunities for recreational activities, from world-class art and performances to sporting events and fine dining.

Applicants should send their curriculum vitae, statements of teaching and research plans, and three reference letters to: Professor Franco Cerrina, Chair, Department of Electrical & Computer Engineering, Boston University, 8 Saint Mary's Street, Room 324, Boston, MA 02215. Materials may also be sent in PDF format via email to rbell@bu.edu. Boston University is an Equal Opportunity/Affirmative Action Employer.

For additional information, please visit http://www.bu.edu/ece.

### Carleton College
**Computer Science Department**
***Tenure-Track Faculty Position***

Carleton College invites applications for a tenure-track position in computer science, in any area of specialization, to begin September 1, 2010. More details on this position can be found at
cs.carleton.edu/hiring.

Carleton is a highly selective liberal arts college with outstanding, enthusiastic students. We seek an equally enthusiastic computer scientist committed to excellence in teaching, curriculum design, ongoing research, and undergraduate research advising. To apply, send cover letter, CV, brief statements describing teaching philosophy and research agenda, graduate transcript, and three reference letters, at least one of which addresses your teaching, to cssearch@carleton.edu. Electronic submissions (pdf) are preferred, or write to: Tenure-Track Search Committee, Computer Science Department, Carleton College, One North College Street, Northfield, MN 55057. Applications completed by January 1, 2010 will receive full consideration.

Carleton College is an affirmative action/equal opportunity employer. We are committed to developing our faculty to better reflect the diversity of our student body and American society. Women and members of minority groups are strongly encouraged to apply.

### Coastal Carolina University
**Assistant/ Associate Professors of Computer Science**

The Department of Computer Science & Information Systems at Coastal Carolina University invites applications for two Assistant / Associate Professor positions beginning fall 2010. While special interest will be given to candidates specializing in data mining, computational intelligence, biomedical applications, and embedded systems, qualified applicants from all areas of Computer Science are encouraged to apply. Preference will be given to candidates who can work across multiple disciplines and collaborate with other faculty, developing successful inter-disciplinary research proposals.

The Department's 11 faculty support an ABET accredited B.S. in Computer Science, and a B.S. degree in Information Systems. Applicants must have a Ph.D. in Computer Science. Faculty are expected to commit to quality teaching, scholarship and service.

The University is a public mid-sized, comprehensive liberal arts-oriented institution. The University is located in Conway, South Carolina, just nine miles from the Atlantic coastal resort of Myrtle Beach, one of the fastest-growing metropolitan areas in the nation. It has an enrollment of 8,200 students and is expected to have continued growth for the next several years. The University is a part of the South Carolina system of public education and has close ties with its founders, the Horry County Higher Education Commission.

Interested candidates should apply online at: http://jobs.coastal/edu. Candidates are asked to submit: a letter of application, curriculum vitae, three letters of reference, and two conference proceedings or journal articles, along with a 4-5 paragraph statement of teaching philosophy. Review of applications will begin immediately and continue until the positions are filled.

Applicants must have a Ph.D. in Computer Science. Faculty are expected to commit to quality teaching, scholarship and service.

While special interest will be given to candidates specializing in data mining, computational intelligence, biomedical applications, and embedded systems, qualified applicants from all areas of Computer Science are encouraged to apply. Preference will be given to candidates who can work across multiple disciplines and collaborate with other faculty, developing successful inter-disciplinary research proposals.

Coastal Carolina University
is an EO/AA employer.

### College of William & Mary
**Faculty Position – Assistant Professor**

We invite applications for a tenure-track assistant professor position in Computer Science for Fall 2010. We are interested in individuals with research expertise in all areas of computer science. Applicants must hold a Ph.D. in computer science or a related field at the time of appointment, must have a strong research record, and should have an interest in teaching.

The College of William and Mary, consistently ranked in the elite group of the Best National Universities Doctoral by U.S. News and World Report, has committed to a multi-year effort to strengthen its computer science research program. The department currently consists of fourteen faculty members who support B.S., M.S., and Ph.D. programs. More information about the department and university can be obtained at:
http://www.cs.wm.edu.

Applicants should submit a current resume, research and teaching statements, the names of at least three references, and supporting documents they consider most relevant. We prefer that the application materials be submitted as PDF attachments in email to 09search@CS.WM.EDU. If necessary, hard copy may be submitted to:

Faculty Search Committee
Department of Computer Science
The College of William & Mary
P.O. Box 8795
Williamsburg, VA 23187-8795

Review of applications will begin December 15 and continue until the position is filled. The College is an EEO/AA employer.

---

**Duke University**
**Department of Computer Science**

The Department of Computer Science at Duke University invites applications and nominations for tenure-track faculty positions at an assistant professor level, to begin August 2010. We are interested in strong candidates in all active research areas of computer science, both core and interdisciplinary areas, including algorithms, artificial intelligence, computational economics, computer architecture, computer vision, database systems, distributed systems, machine learning, networking, security, and theory.

The department is committed to increasing the diversity of its faculty, and we strongly encourage applications from women and minority candidates.

A successful candidate must have a solid disciplinary foundation and demonstrate promise of outstanding scholarship in every respect, including research and teaching. Please refer to www.cs.duke.edu for information about the department and to www.provost.duke.edu/faculty/ for information about the advantages that Duke offers to faculty.

Applications should be submitted online at www.cs.duke.edu/facsearch. A Ph.D. in computer science or related area is required. To guarantee full consideration, applications and letters of reference should be received by January 3, 2010.

Durham, Chapel Hill, and the Research Triangle of North Carolina are vibrant, diverse, and thriving communities, frequently ranked among the best places in the country to live and work. Duke and the many other universities in the area offer a wealth of education and employment opportunities for spouses and families.

Duke University is an affirmative action, equal opportunity employer.

---

**Duke University**
**Tenure-track Faculty**

The Department of Electrical and Computer Engineering at Duke University invites applications for tenure-track faculty positions at all levels. We are interested in strong candidates in all areas of computer engineering.

Applications should be submitted online at www.ee.duke.edu/employment.

Applications and letters of reference should be received by December 31, 2009.

Duke University is an affirmative action, equal opportunity employer.

---

**Earlham College**
**Visiting Assistant Professor of Computer Science**

One-year, full-time, non-tenure-track beginning Fall 2010. Review of applications begins January 15, 2010. CS at Earlham is a small, dynamic, innovative, student-centered department. See http://cs.earlham.edu/ad.html

---

**The George Washington University**
**Department of Computer Science**
***Three Faculty Positions***

The Department of Computer Science at The George Washington University is seeking applicants for three faculty positions. The first is (A) a tenured senior position in security, at the rank of Associate or Full Professor. The second position is (B) a tenure-track position in the area systems security, and the third is (C) a tenure-track position in the area of Artificial Intelligence with a focus on Robotics. Position B and C will be at the rank of Assistant or Associate Professor. The rank for all positions will depend on experience. Successful candidates may start as early as Fall 2010.

Basic Qualifications: All applicants must have a Ph.D degree in Computer Science or a closely

---

# The University of New Mexico

***Great people doing great things***

## Computer Science Assistant Professor Position

The Department of Computer Science invites applications for one probationary appointment leading to a tenure decision at the Assistant Professor level. We seek applicants from across computer science, including but not limited to large data systems, networks, software design and engineering, graphics and visualization, and AI and adaptive systems. We are a strongly interdisciplinary department and are particularly interested in applicants pushing the boundaries of computer science with other fields.

For more information refer to full job ad at **http://www.cs.unm.edu/jobs/**

Candidates must have completed a doctorate in CS or a relevant area by August 15, 2010. Applicants should demonstrate a strong commitment to undergraduate and graduate education, have a research profile in one of the general CS areas enumerated above, and demonstrate the ability to establish a nationally visible research program.

For best consideration, complete applications must be received by January 11, 2010. The position will remain open until filled. Each application must include a cover letter summarizing the applicant's experience, curriculum vita, research statement, teaching statement, and three letters of reference. It is the applicant's responsibility to ensure that the letters of reference are submitted before the application deadline.

Applications must be submitted online through **https://UNMJobs.unm.edu**, by referencing **posting #0803446**. Reference letters should be emailed directly to **faculty_search@cs.unm.edu**

Inquiries should be sent to: **faculty_search@cs.unm.edu**

**The University of New Mexico is an Equal Opportunity/Affirmative Action Employer and Educator.**
**For additional information about UNM see http://www.unm.edu**

---

## INFORMATION SCIENCE
### Cornell University (Ref: ISF#3)

Cornell's Information Science Program (**www.infosci.cornell.edu**) is seeking to fill one or more tenure-track faculty positions.

Information Science at Cornell brings together those interested in studying information systems in their social, cultural, economic, historical, legal, and political contexts. We are especially interested in the areas of:

- large information networks and communities with a focus on methods for their design, analysis, or visualization;
- analysis of socio-technical systems with a focus on their governance or the practices that emerge within them.

We will also consider candidates in other areas of Information Science. The ideal applicant will have a strong mix of both technical and social science research skills and accomplishments. Please see **www.infosci.cornell.edu/jobs** for a full description of the positions.

To ensure full consideration, applications should be received by January 15, 2010, but will be accepted until all positions are filled. Applicants should submit a curriculum vita, brief statements of research and teaching interests through the web at **http://www.infosci.cornell.edu/apply**, and arrange to have at least three letters of reference uploaded on the Web.

Cornell University
*Cornell University is an Affirmative Action/ Equal Opportunity Employer and Educator.*

related field. (A) Applicants to the tenured senior position in security must currently have well-funded research programs, must be a recognized scholar in the research community, and must be prepared to take on a leading research role within the department and in the field. (B and C) ABD candidates may apply for the Assistant Professor rank, but they must complete their Ph.D degree by August 15, 2010. Applicants for the Assistant Professor rank must demonstrate a potential for developing an advanced research program and for attracting significant research funding. Applicants for the Associate Professor rank must have well-established and well-funded research programs. All applicants must have demonstrated teaching excellence or potential at both undergraduate and graduate levels.

The George Washington University is the largest academic institution in the nation's capital with close access to many Federal funding agencies and research laboratories. The University offers comprehensive programs of undergraduate and graduate liberal arts studies as well as degrees in engineering, law, medicine, public health, education, business and international affairs. A private institution, GW prides itself on excellent research, quality education, and low student-teacher ratio. The exceptional location affords the GW community unique cultural and intellectual opportunities. In the high-tech sector, the Washington, DC Metropolitan area is one of the largest IT areas in the nation, putting us in the center of activities such as security and biotechnology.

The Department of Computer Science offers an accredited Bachelor of Science program, a Bachelor of Arts program, and Master's and Ph.D degrees. The Department has 18 faculty members, numerous affiliated and adjunct faculty members, and over 425 students. The Department has educational and research programs in security, networks, graphics, biomedical applications, search and data mining, human computer interaction, and machine intelligence, with funding from various agencies; a NSA-designated Center of Academic Excellence and Center of Academic Excellence-Research in security, with funding from NSF, DOD, and other agencies; and NIH-funded collaborations with the medical school in the biomedical areas. For further information please refer to http://www.cs.gwu.edu.

Review of applications will begin on December 1, 2009, and will continue through the Spring 2010 semester, until the position is filled. Application Procedure: To be considered, applicants must send a letter containing (i) a brief statement that clearly indicates the position and rank of interest, (ii) a curriculum vita, (iii) a statement of research and teaching interests, and (vi) complete contact information for at least three references. These and other relevant supporting materials should be sent to: Chair, Faculty Search Committee, Department of Computer Science, PHIL 703, The George Washington University, Washington D.C. 20052. Electronic submissions are preferred, and can be sent to cssearch@gwu.edu. Only complete applications will be considered. Inquiries about applying will be accorded the utmost discretion. For complete instructions on the application process, please visit the department website www.cs.gwu.edu.

The George Washington University is an equal opportunity/affirmative action employer.

## Indiana University Southeast
**Assistant Professor of Informatics**

Tenure-track Informatics position for Fall 2010. Applications accepted until January 15, 2010 or until position is filled. For further position description and how to apply go to www.ius.edu/hr/employment_opportunities.cfm

## IST Austria is looking for
**Professors and Assistant Professors**

IST Austria (Institute of Science and Technology Austria) is a new Institute located near Vienna, dedicated to basic research at the highest international level. The Institute invites applications and nominations for Professors and Assistant Professors in Life Sciences, Physical Sciences, Mathematics and Computer Science, as well as in any multidisciplinary field.

The Institute (www.ist.ac.at), established by the Austrian Government, opened its campus in 2009. Its funding is substantial, allowing for over 500 employees and graduate students by 2016. IST Austria is entitled to award Ph.D. degrees and includes an English-language Graduate School. It aims to achieve an international mix of scientists and recruit them solely on the basis of their individual excellence and potential contribution to research.

The Institute is recruiting leaders of independent research groups. Professors will have indefinite contracts and Assistant Professors will have fixed-term contracts for an initial period of five years, with a possible, but not automatic, renewal for two additional years. Before the end of this period, the scientist will be considered for an indefinite appointment as a Professor at IST Austria, the decision being based on merit only (as is the case for a "Tenure-Track Assistant Professor" at U.S. universities).

The selected candidates will receive a competitive salary and a substantial annual research budget, covering operating expenses and the cost of Ph.D. students, postdoctoral fellows, and technical staff. Additional costs of starting a new laboratory, including instruments and infrastructure, will be offered separately. Scientists are also expected to apply for external research grants.

Applications and nominations should be sent to professor@ist.ac.at or assistant.professor@ist.ac.at, depending on the relevant position. Applications must include a CV, a list of publications and a research plan. Nominations should include an appraisal of the achievements and scientific qualifications of the nominee.

IST Austria is committed to Equality and Diversity. In particular female applicants are encouraged to apply.

## The Johns Hopkins University
**Tenure-track Faculty Positions**

The Department of Computer Science at The Johns Hopkins University is seeking applications for tenure-track faculty positions at the Assistant Professor level, as well as applications for a focused search at the Associate or Full Professor level.

EPFL

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

## Faculty Positions in Computer Science
### Ecole polytechnique fédérale de Lausanne

The School of Computer and Communication Sciences at EPFL invites applications for faculty positions in computer science. We are primarily seeking candidates **for tenure-track assistant professor positions, but suitably qualified candidates for senior positions will also be considered.**

Successful candidates will develop an independent and creative research program, participate in both undergraduate and graduate teaching, and supervise PhD students.

Candidates from all areas of computer science will be considered, but preference will be given to candidates with interests in **algorithms, bio-informatics, machine learning and verification.**

EPFL offers internationally competitive salaries, significant start-up resources, and outstanding research infrastructure.

To apply, please follow the application procedure at http://icrecruiting.epfl.ch. The following documents are requested in PDF format: curriculum vitae, including publication list, brief statements of research and teaching interests, names and addresses (including e-mail) of 3 references for junior positions, and 6 for senior positions. Screening will start on **January 15, 2010**. Further questions can be addressed to :

**Professor Willy Zwaenepoel**
**Dean**
**School of Computer and**
**Communication Sciences EPFL**
**CH-1015 Lausanne, Switzerland**
**recruiting.ic@epfl.ch**

For additional information on EPFL, please consult: **http://www.epfl.ch** or **http://ic.epfl.ch**

EPFL is an equal opportunity employer.

At the Assistant Professor level all areas will be considered, but candidates with research agendas in computational genomics, computer systems, and health-related applications will receive special attention. At the Associate or Full Professor level we are seeking outstanding applicants in the security area. All applicants must have a Ph.D. in computer science or a related field and are expected to show evidence of an ability to establish a strong, independent, multidisciplinary, internationally recognized research program.

Commitment to quality teaching at the undergraduate and graduate levels will be required of all candidates considered. The department webpage at http://www.cs.jhu.edu provides information about the department, including links to research laboratories and centers.

Applicants should apply using the online application which can be accessed from https://academicjobsonline.org/ajo/JHU/CS/201 (for full consideration, by January 4 2010). Questions should be directed to fsearch@cs.jhu.edu. The Department is committed to building a diverse educational environment; women and minorities are strongly encouraged to apply. The Johns Hopkins University is an EEO/AA employer.

Faculty Search
Johns Hopkins University
Department of Computer Science
Room 224 New Engineering Building
Baltimore, MD 21218-2694
Fax: 410-516-6134
Phone: 410-516-8775
fsearch@cs.jhu.edu
http://www.cs.jhu.edu/apply

## Lawrence Technological University
**Assistant Professor of Computer Science**

For appointment in August 2010. The ideal candidate will have a Ph.D. degree in computer science, be primarily committed to the development of undergraduate and professional graduate computer science students through teaching, applied projects and scholarship, be able to work effectively in interdisciplinary teams, and believe strongly in the value of both theory and application. Applicants should email a cover letter, curriculum vitae, statement of teaching philosophy and research interests, and three letters of recommendation.

## Lehigh University
**Department of Computer Science and Engineering**
***Faculty Opening***

Applications are invited for a tenure-track Assistant Professor position in Bioinformatics in the Computer Science and Engineering Department (http://www.cse.lehigh.edu) of Lehigh University. The position starts in August 2010.

The potential to establish a successful research program and teach effectively at both the undergraduate and graduate level are essential prerequisites. The successful applicant will hold a Ph.D. in Computer Science, Computer Engineering, Bioinformatics, or a closely related field. Outstanding candidates in areas of computer science with a direct connection to bioinformatics will be considered. Applicants should have an interest in teaching core courses in computer science as well as courses in their research area. The successful applicant will be expected to contribute directly to Lehigh's bioengineering program.

The faculty of the Computer Science and Engineering (CSE) department includes IEEE and ACM fellows, and four NSF CAREER award winners. We offer B.A., B.S., M.S., and Ph.D. degrees in Computer Science and jointly oversee B.S., M.S., and Ph.D. degree programs in Computer Engineering with the department of Electrical and Computer Engineering. We also offer a B.S. in Computer Science and Business with the College of Business and Economics. Lehigh offers a degree in Bioengineering and has a variety of research programs related to bioinformatics and medical informatics, including biomedical image processing, biopharmaceuticals, algorithms for mining large genomic databases, and assistive robotics. Lehigh is developing a biotech cluster that includes a vibrant undergraduate program, a new graduate program in bioengineering, both parts of a university-level initiative in healthcare research. The bioengineering program draws associated faculty from nearly all engineering and several science departments and will provide opportunities for the successful applicant to develop collaborative research projects.

Lehigh is a private institution that is consistently ranked among the top 40 national research universities by U.S. News & World Report and is rated "most selective" by both Barron's and Peterson's guides. Located in Bethlehem, Pennsylvania, Lehigh is 80 miles west of New York City and 50 miles north of Philadelphia, providing an accessible and convenient location that offers an appealing mix of urban and rural lifestyles.

Applications can be submitted online at http://www.cse.lehigh.edu/faculty-search, and should include a cover letter, vita, and both teaching and research statements. In addition, please provide the names and email addresses of at least three references. Applications will be evaluated until the position is filled, but materials should be received by January 15, 2010 for full consideration. Lehigh University is an Equal Opportunity/Affirmative Action Employer. Lehigh University provides comprehensive benefits including domestic partner benefits (see http://www.lehigh.edu/~insloan). Questions concerning this search may be sent to faculty-search@cse.lehigh.edu.

## The University of New Mexico
***Great people doing great things***

## Dean, School of Engineering

The University of New Mexico is seeking a visionary leader for the position of Dean of the School of Engineering (SOE). Located in Albuquerque, UNM is a Carnegie Very High Research University and a federally-designated Hispanic-Serving Institution. Founded in 1889, UNM is the largest and most comprehensive of the state's institutions of higher education. The Dean, who reports directly to the Provost/Executive Vice President for Academic Affairs, provides academic and administrative leadership to the SOE. As the primary academic and administrative officer of the faculty of the School of Engineering, the Dean will assume a central leadership role in continuing development of the SOE's disciplines toward national eminence with a commitment to research and teaching. The Dean is responsible for improving and promoting the SOE in areas of instruction, research, fiscal management, development and personnel. The School of Engineering (SOE) has over 100 faculty in five academic departments: Chemical and Nuclear Engineering, Civil Engineering, Computer Science, Electrical and Computer Engineering, and Mechanical Engineering. Among the internationally recognized centers associated with the school are the Center for High Technology Materials, the Center for Microengineered Materials, the Center for Advanced Research Computing, the Mind Research Network, the Center for Biomedical Engineering, and the Center for Emerging Energy Technologies. UNM is also a member institution of the WERC-A Consortium for Environmental Education & Technology Development. The SOE is ranked nationally for both its undergraduate and graduate programs.

Minimum qualifications for the position include an earned Ph.D. in Engineering or Computer Science or doctoral degree in a related discipline, a record of accomplishment or credentials that would merit appointment at the rank of tenured full professor in an academic department of UNM's School of Engineering, and three years of management and administrative experience at the level of department head, chair, research director, center director, dean, or the equivalent.

For a complete listing of the position description including desired qualifications and application procedures, access the job posting on UNM's online application system, UNMJobs.unm.edu, by referencing posting number 0803339. For best consideration, applications should be submitted by January 11, 2010, and the position is open until filled. Request a copy of the position announcement from Bonnie Leigh Reifsteck, Search Coordinator, **bonniec@unm.edu (phone: 505.277.2611)**.

**The University of New Mexico is an Equal Opportunity/Affirmative Action Employer and Educator.**
**For additional information about UNM see http://www.unm.edu**

## Massachusetts Institute of Technology
**Faculty Positions**

The Department of Electrical Engineering and Computer Science (EECS) seeks candidates for faculty positions starting in September 2010. Appointment would be at the assistant or untenured associate professor level. In special cases, a senior faculty appointment may be possible. Faculty duties include teaching at the graduate and undergraduate levels, research, and supervision of student research. We will consider candidates with backgrounds and interests in any area of electrical engineering and computer science. Faculty appointments will commence after completion of a doctoral degree.

Candidates must register with the EECS search website at https://eecs-search.eecs.mit.edu, and must submit application materials electronically to this website. Candidate applications should include a description of professional interests and goals in both teaching and research. Each application should include a curriculum vita and the names and addresses of three or more individuals who will provide letters of recommendation. Letter writers should submit their letters directly to MIT, preferably on the website or by mailing to the address below. Please submit complete application by December 15, 2009.

Send all materials not submitted on the website to:

Professor W. Eric L. Grimson
Department Head, Electrical Engineering and Computer Science
Massachusetts Institute of Technology
77 Massachusetts Avenue, Room 38-401
Cambridge, MA 02139

M.I.T. is an equal opportunity/affirmative action employer.

---

### Max Planck Institute for Software Systems (MPI-SWS)
**Tenure-track openings**

Applications are invited for tenure-track and tenured faculty positions in all areas related to the design, analysis and engineering of software systems, including programming languages, formal methods, security, distributed, networked and embedded systems, databases and information systems, and human-computer interaction. A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups. Senior candidates must have demonstrated leadership abilities and recognized international stature.

MPI-SWS, founded in 2005, is part of a network of eighty Max Planck Institutes, Germany's premier basic research facilities. MPIs have an established record of world-class, foundational research in the fields of medicine, biology, chemistry, physics, technology and humanities. Since 1948, MPI researchers have won 17 Nobel prizes. MPI-SWS aspires to meet the highest standards of excellence and international recognition with its research in software systems.

To this end, the institute offers a unique environment that combines the best aspects of a university department and a research laboratory:

a) Faculty receive generous base funding to build and lead a team of graduate students and post-docs. They have full academic freedom and publish their research results freely.
b) Faculty have the opportunity to supervise doctoral theses, teach graduate and undergraduate courses, and have the flexibility to incorporate teaching into their research agenda.
c) Faculty are provided with outstanding technical and administrative support facilities as well as internationally competitive compensation packages.

Funds have been committed to grow the institute to a strength of 17 tenured and tenure-track faculty, and about 100 doctoral and post-doctoral positions. Additional growth through outside funding is expected. We maintain an open, international and diverse work environment and seek applications from outstanding researchers regardless of national origin or citizenship. The working language is English; knowledge of the German language is not required for a successful career at the institute.

The institute is located in Kaiserslautern and Saarbruecken, in the tri-border area of Germany, France and Luxembourg. The area offers a high standard of living, beautiful surroundings and easy access to major metropolitan areas in the center of Europe, as well as a stimulating, competitive and collaborative work environment. In immediate proximity are the MPI for Informatics, Saarland University, the Technical University of Kaiserslautern, the German Center for Artificial Intelligence (DFKI), and the Fraunhofer Institutes for Experimental Software Engineering and for Industrial Mathematics.

Qualified candidates should apply online at http://www.mpi-sws.org/application. The review of applications will begin on January 4, 2010, and applicants are strongly encouraged to apply by that date; however, applications will continue to be accepted through January 2010.

The institute is committed to increasing the representation of minorities, women and individuals with physical disabilities in Computer Science. We particularly encourage such individuals to apply.

---

### New Mexico State University
**Assistant Professor**

The **Computer Science Department** at New Mexico State University invites applications for a tenure-track position at the assistant professor level, with appointment starting in the Fall semester 2010. We are particularly interested in candidates with expertise in database management systems, data mining or related areas. Applications from women and members of traditionally under-represented groups are particularly encouraged. Salary and start-up package will be competitive and commensurate with qualifications and experience.

The **minimum qualifications** are a Ph.D. degree in Computer Science or in a closely-related discipline by the time of appointment, along with demonstrated evidence of excellence in teaching and research. We particularly solicit applications from candidates with attitude for collaborative research, whose research foci can complement the existing expertise in the department; we are also interested in individuals that can promote interdisciplinary research initiatives. The successful candidate will be expected to develop an independent research program and teach graduate and undergraduate courses in Computer Science.

---

### New York University/Courant Institute of Mathematical Sciences
**Department of Computer Science**

The department expects to have several regular faculty positions beginning in September 2010 and invites candidates at all levels. We will consider outstanding candidates in any area of computer science with systems being high-priority area.

Faculty members are expected to be outstanding scholars and to participate in teaching at all levels from undergraduate to doctoral. New appoin-

---

NUS National University of Singapore

## TEMASEK RESEARCH FELLOWSHIP

The Temasek Research Fellowship (TRF) is a prestigious scheme aimed at recruiting outstanding young researchers at the post-doctoral level to undertake research as Principal Investigators and lead teams to undertake defence-related research in the National University of Singapore (NUS).

The TRF is a 3-year Fellowship with an option to extend up to 3 years. The Temasek Research Fellow (TRF-RF) may be offered a faculty appointment with NUS at the end of their term.

In addition to an attractive remuneration package that will commensurate with qualification and experience, the TRF-RF will be provided a research grant to pursue his/her research at Temasek Laboratories at NUS or other research entities at NUS.

For more information and application procedure, please visit NUS at

**http://www.nus.edu.sg/dpr/funding/fellowship.htm**

**Closing date: 22 December 2009**

**Short-listed candidates will be invited for an interview and scientific presentation expected to be held in March 2010**

tees will be offered competitive salaries and startup packages, with affordable housing within a short walking distance of the department. New York University is located in Greenwich Village, one of the most attractive residential areas of Manhattan.

The department has 33 regular faculty members and several clinical, research, adjunct, and visiting faculty members. The department's current research interests include algorithms, cryptography and theory; computational biology; distributed computing and networking; graphics, vision and multimedia; machine learning; natural language processing; scientific computing; and verification and programming languages.

Collaborative research with industry is facilitated by geographic proximity to computer science activities at AT&T, Google, IBM, Bell Labs, NEC, and Siemens.

Please apply at
https://webern.cs.nyu.edu/faculty_applications/

To guarantee full consideration, applications should be submitted no later than January 4, 2010; however, this is not a hard deadline, as all candidates will be considered to the full extent feasible, until all positions are filled. Visiting positions may also be available.

New York University is an equal opportunity/affirmative action employer.

## Northwestern University
### Tenure-Track Position in Technology and Social Behavior

The Department of Communication Studies in the Northwestern University School of Communication seeks to hire for a tenure-track appointment beginning September 1, 2010. Rank will be at the level of assistant professor.

We are looking for a candidate who can contribute to a strong interdisciplinary program in technology and social behavior. Possible areas of expertise include but are not limited to: human-computer interaction, computer-mediated communication and information infrastructures, language and behavior in virtual communities, and media and social networks.

The Department of Communication Studies supports a popular undergraduate major and graduate programs in Media, Technology, and Society, Interaction and Social Influence, and Rhetoric and Public Culture, as well as the interdisciplinary Ph.D. program in Technology and Social Behavior. Scholarship includes leading work on new media, social networks, and the cultural determination of the public sphere. Through special resources for research support and scholarly event programming, the department is able to offer rich opportunities for scholarly development.

For full consideration, please send a letter of application with a curriculum vitae, evidence of teaching effectiveness, and three letters of reference to Professor Daniel O'Keefe at the address below by November 15.

Northwestern University is an equal opportunity, affirmative action educator and employer. Women and minorities are encouraged to apply. Hiring is contingent on eligibility to work in the United States.

Technology and Social Behavior Search Chair: Daniel O'Keefe
Search No. 00050140
Department of Communication Studies
2240 Campus Drive
Evanston, Il 60208
tsb-search@northwestern.edu

## Old Dominion University
### Richard T. Cheng Endowed Chair in Computer Science

Applications are invited for the Richard T. Cheng Chair at the tenured rank of full professor. The Computer Science Department offers BS, MS and PhD degrees and features research programs in the areas of bioinformatics, computational science, digital libraries, computer and wireless networking, security, and computer simulation. Old Dominion University has major collaborative opportunities with the Virginia Modeling and Simulations Center, NASA Langley, the Thomas Jefferson National Laboratory, the Eastern Virginia Medical School, and the DoD Joint Training Analysis and Simulation Center.

We are seeking an internationally recognized scholar with a significant ongoing externally funded research program to fill this endowed position. The Cheng Chair will be expected to play a leadership role in departmental research and to mentor junior faculty, post doctoral fellows and graduate students. The department is particularly interested in attracting scholars who will enable interdisciplinary collaboration. Minimum requirements include a PhD in Computer Science or a closely related field.

Applicants should submit a letter of interest and curriculum vitae that includes contact information for four references. Applications, nominations or inquiries should be sent to: chengsearch@cs.odu.edu or to: Chair, Cheng Chair Committee, Computer Science Department, 3300 E & CS Bldg, Old Dominion University, Norfolk, VA 23529-0223.

Screening of applicants will begin immediately and continue until the position is closed. Old Dominion University is an affirmative action, equal opportunity institution and requires compliance with the Immigration Reform and Control Act of 1986.

## Old Dominion University
### Computer Science Faculty Positions

The College of Sciences at Old Dominion University is undergoing a major expansion to increase research activities of faculty and to support the recruiting of outstanding graduate and undergraduate students. The Department of Computer Science has recently hired a new chair and seeks candidates for multiple tenure track/tenured positions at senior and junior levels.

Candidates for full or associate professor must show evidence of strong, active research programs as evidenced by publications and peer-reviewed funding and demonstrate a strong teaching record. Candidates at the assistant professor level must demonstrate the ability to build strong research programs and perform quality teaching. The department offers competitive salaries and substantial start-up packages. The Department of Computer Science is well supported by peer-reviewed grants from NSF, NASA, DOE and other funding agencies. Department is particularly interested in attracting researchers who will enable interdisciplinary collaboration. Areas of interest are broad and include bioinformatics, computational biology, parallel computing, modeling and simulation, computational science, databases/data mining, but all strong applications will be considered.

Interested candidates should submit a curriculum vitae, a statement of research activities and future research plans, contact information for four references and a statement of teaching philosophy. Applicants should specify the position and level for which they are applying. Electronic applications are preferred and should be sent to searchcommittee@cs.odu.edu. Paper applications can be sent to:

Search Committee, Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162.

Applicants from dual-career couples are welcome. Review of applicants will begin immediately and continue until the positions are filled.

Old Dominion University is an affirmative action, equal opportunity institution and requires compliance with the Immigration Reform and Control Act of 1986.

## Rochester Institute of Technology
### Computing and information sciences
*Faculty Openings for Fall 2010*

Rochester Institute of Technology's B. Thomas Golisano College of Computing and Information Sciences (GCCIS) invites applications and nominations for the following positions:

A tenure-track faculty position in the Networking, Security, and Systems Administration department (www.nssa.rit.edu) primarily in the area of network/computer security and information assurance. Additionally, experience in computer networking and/or systems administration is preferred (IRC#34428)

A tenure-track faculty position in the Interactive Games and Media department (igm.rit.edu/jobs) in the areas of computer games and interactive entertainment (IRC#34427).

Successful candidates must demonstrate excellence in teaching and scholarship and have the ability to contribute in meaningful ways to RIT's commitment to cultural diversity and pluralism. A Ph.D. in computing or a closely related field or other terminal degree (such as an MFA for IRC#34427) with equivalent experience is required, as is a record of publicly disseminated scholarly and/or creative work and demonstrated potential for future scholarship.

The Golisano College of Computing and Information Sciences is home to the Computer Science, Software Engineering, Information Sciences and Technologies, Interactive Games & Media, and Networking, Security, and Systems Administration departments, as well as the Ph.D. program in Computer and Information Sciences. The college has 105 faculty and over 2400 undergraduate and 600 graduate students.

Candidates should visit https://mycareer.rit.edu and refer to the IRC number listed above for specific information about the positions and the application process. Refer to www.rit.edu for information about RIT and the B. Thomas Golisano College of Computing and Information Sciences.

## Rutgers University
**Tenure-Track Position**

The Department of Computer Science at Rutgers University invites applications for tenure-track faculty positions at the rank of Assistant, Associate or full Professor, with appointments starting in September 2010, subject to the availability of funds. All areas in experimental computer systems will be considered, but special emphasis will be given to pervasive computing and computer architecture.

Applicants for this research/teaching position must, at minimum, be in the process of completing a dissertation in Computer Science or a closely related field, and should show evidence of exceptional research promise, potential for developing an externally funded research program, and commitment to quality advising and teaching at the graduate and undergraduate levels. Hired candidates who have not defended their Ph.D. by September will be hired at the rank of Instructor, and must complete the Ph.D. by December 31, 2010 to be eligible for tenure-track title retroactive to start date.

Applicants should go to http://www.cs.rutgers.edu/employment/ and submit their curriculum vitae, a research statement addressing both past work and future plans and a teaching statement along with three letters of recommendation.

Applications should be received by January 2, 2010 for full consideration.

Rutgers subscribes to the value of academic diversity and encourages applications from individuals with varied experiences, perspectives, and backgrounds. Females, minorities, dual-career couples, and persons with disabilities are encouraged to apply.

Rutgers is an affirmative action/equal opportunity employer.

## Santa Clara University
**Assistant Professor in Scientific Computation/Statistics**

The Department of Mathematics and Computer Science at Santa Clara University invites applications for a tenure-track Assistant Professor position starting in September 2010 from candidates with expertise in scientific computation or statistics and interest in mathematical biology, computational physics or chemistry, or environmental science.

Application deadline is January 20, 2010. For more information regarding application submissions, see www.scu.edu/hr/careers/faculty.cfm

## Southern Polytechnic State University
**Tenure-track positions in Computing**

We are seeking to fill up to four tenure-track Assistant Professor positions (other ranks considered) in Information Technology, Computer Game Design and Development, and Software Engineering, and a Lecturer position in Computer Science. For the tenure-track positions, applicants must hold a relevant Ph.D., be committed to excellence in teaching, and have substantial research potential. SPSU is located in the metro Atlanta area, which offers excellent quality of life. For more details, including specializations of interest, go to http://www.spsu.edu/hr/faculty.html.

## Stanford University
**Department of Computer Science**
***Faculty Opening – Assistant or untenured Associate Professor***

The Department of Computer Science at Stanford University invites applications for a tenure-track faculty position at the junior level (Assistant or untenured Associate Professor). We give higher priority to the overall originality and promise of the candidate's work than to the candidate's sub-area of specialization within Computer Science.

We are seeking applicants from all areas of Computer Science, spanning theoretical foundations, systems, software, and applications. We are also interested in applicants doing research at the frontiers of Computer Science with other disciplines, especially those with potential connections to Stanford's main multidisciplinary initiatives: Energy, Human Health, Environment and Sustainability, the Arts and Creativity, and the International Initiative.

Applicants must have completed (or be completing) a Ph.D., must have demonstrated the ability to pursue a program of research, and must have a strong commitment to graduate and undergraduate teaching. A successful candidate will be expected to teach courses at the graduate and undergraduate levels, and to build and lead a team of graduate students in Ph.D. research. Further information about the Computer Science Department can be found at http://cs.stanford.edu. The School of Engineering website may be found at http://soe.stanford.edu.

Applications should include a curriculum vita, brief statements of research and teaching interests, and the names of at least four references. Candidates are requested to ask references to send their letters directly to the search committee. Applications and letters should be sent to: Search Committee Chair, c/o Laura Kenny-Carlson, via electronic mail to search@cs.stanford.edu.

The review of applications will begin on January 4, 2010, and applicants are strongly encouraged to submit applications by that date; however, applications will continue to be accepted until the position is filled, but no later than May 1, 2010.

**Stanford University is an equal opportunity employer and is committed to increasing the diversity of its faculty. It welcomes nominations of and applications from women and members of minority groups, as well as others who would bring additional dimensions to the university's research and teaching missions.**

## Stevens Institute of Technology
**Assistant Professor of Computer Science**

The Computer Science Department at Stevens Institute of Technology invites applications for a tenure-track position beginning in August 2010. Special consideration will be given to candidates in computer vision, computer graphics, and machine learning at the assistant professor level. However, outstanding applicants at other levels and/or in other areas of Computer Science may also be considered.

Applicants are expected to have a Ph.D. in Computer Science or a closely related field, a demonstrated record of excellence in research, and a strong commitment to teaching. A successful candidate will be expected to conduct a vigorous, funded research program and to teach at both the undergraduate and graduate levels.

Stevens Institute of Technology is a private university located in Hoboken, New Jersey. The 55-acre campus is on the Hudson River across from midtown Manhattan within a few minutes from NYC via public transportation. Hoboken is a small upscale city, the residence of New Jersey's governor, and the residence of choice for many professionals working in NYC. Faculty live in Hoboken, NYC, and in suburban communities in Northern New Jersey along commuter train lines to Hoboken and NYC. Stevens' location offers excellent opportunities for collaborations with nearby universities such as NYU, Princeton, Columbia, and Rutgers/DIMACS as well as industrial research laboratories such as Bell Labs, AT&T Labs, IBM Research, Google New York, Siemens, and the Sarnoff Corporation.

Applications should be submitted electronically at http://www.cs.stevens.edu/Search. Applications should include a curriculum vitae, teaching and research statements, and contact information for at least three references. Candidates should ask their references to send letters directly to the search committee. PDF is preferred for all application materials and reference letters. Further information is provided at the web site.

Review of applications will begin on December 1, 2009.

Stevens is an Affirmative Action/Equal Opportunity employer.

## Swarthmore College
**Computer Science Department**

Applications are invited for a two-year Visiting Assistant Professor position beginning August 2010.

Swarthmore College is a small, selective liberal arts college located in a suburb of Philadelphia. The Computer Science Department offers majors and minors in computer science at the undergraduate level. Applicants must have teaching experience and should be comfortable teaching a wide range of courses at the introductory and intermediate level. We are particularly interested in candidates who specialize in theory and algorithms or in systems areas, however, we will consider candidates from all areas of CS. A Ph.D. in CS by or near the time of appointment is preferred (ABD is required). We expect to begin interviewing in early February 2010.

See http://cs.swarthmore.edu/jobs for application submission information and more details about the position. Swarthmore College is an equal opportunity employer. Applications from women and members of minority groups are encouraged. Applications will be accepted until the position is filled.

## The Ohio State University
**Assistant Professor**

The Department of Computer Science and Engineering (CSE), **The Ohio State University**, invites applications for two tenure-track positions at the **Assistant Professor** level. The positions are open to all CSE areas (artificial intelligence, graphics and animation, networking, software engineering and programming languages, systems, and theory) with priority consideration given to candidates in database systems and software engineering & programming languages.

Applicants should hold or be completing a Ph.D. in CSE or a closely related field, and have a commitment to and demonstrated record of excellence in research as well as a commitment to excellence in teaching.

To apply, please submit your application via the online database. The link can be found at:

http://www.cse.ohio-state.edu/department/positions.shtml

Review of applications will begin in January and will continue until the positions are filled.

The Ohio State University is an Equal Opportunity/Affirmative Action Employer. Women, minorities, or individuals with disabilities are encouraged to apply.

## Toyota Technological Institute at Chicago (TTI-C)

**Faculty Positions at All Levels**

Toyota Technological Institute at Chicago (TTI-C) is a philanthropically endowed degree-granting institute for computer science located on the University of Chicago campus. The Institute is expected to soon reach a steady-state of 12 traditional faculty (tenure and tenure track), and 12 limited term faculty. Applications are being accepted in all areas, but we are particularly interested in:

Theoretical computer science
Speech processing
Machine learning
Computational linguistics
Computer vision
Scientific computing
Programming languages

Positions are available at all ranks, and we have a large number of limited term positions currently available.

For all positions we require a Ph.D. Degree or Ph.D. candidacy, with the degree conferred prior to date of hire. Submit your application electronically at:

http://ttic.uchicago.edu/facapp/

Toyota Technological Institute at Chicago is an Equal Opportunity Employer

## UMBC

**University of Maryland Baltimore County**
***An Honors University in Maryland***
***Information Systems Department***

The Information Systems Department at UMBC invites applications for a tenure-track faculty position at the Assistant Professor level in the area of human-centered computing (HCC) starting August 2010. Outstanding candidates in other areas will also be considered.

Candidates must have an earned PhD in Information Systems or a related field no later than August 2010. Applicants engaged in interdisciplinary research addressing accessibility (broadly defined to include issues associated with disabilities, age, culture, and other relevant topics), human-information interaction, or social computing are of primary interest. Ideal candidates will be engaged in research that spans two or more of these areas with preference given to those who can collaborate with current faculty. Candidates should have a strong potential for excellence in research, the ability to develop and sustain an externally funded research program, and the ability to contribute to our graduate and undergraduate teaching mission.

The Department offers undergraduate degrees in Information Systems and Business Technology Administration as well as both the MS and PhD in Information Systems. In addition, the Department offers an MS and PhD in Human-Centered Computing, and was ranked 8th in the nation in 2007 for scholarly productivity by Academic Analytics. Consistent with the UMBC vision, the Department has excellent technical support and teaching facilities as well as outstanding laboratory space and state of the art technology. UMBC's Technology Center, Research Park, and Center for Entrepreneurship are major indicators of active research and outreach. Further details on our research, academic programs, and faculty can be found at http://www.is.umbc.edu/. Underrepresented groups including women and minorities are especially encouraged to apply.

Applications will not be reviewed until the following materials are received: a cover letter, a one-page statement of teaching interests, a one-page statement of research interests, one or more sample research papers, and a curriculum vitae. Applicants should also arrange to have three letters of recommendation sent to the department as soon as possible. Electronic submission of materials as PDF documents is preferred. Electronic copies should be sent to bmorris@umbc.edu. Copies can also be sent to: Dr. Andrew Sears, Chair of Faculty Search Committee, Information Systems Department, UMBC, 1000 Hilltop Circle, Baltimore, MD 21250-5398. For inquiries, please contact Barbara Morris at (410) 455-3795 or bmorris@umbc.edu.

Review of applications will begin immediately and will continue until the position is filled. This position is subject to the availability of funds.

UMBC is an Affirmative Action/Equal Opportunity Employer and welcomes applications from minorities, women and individuals with disabilities.

## University of Calgary

**Assistant Professor, Information Security**

The **Department of Computer Science** at the **University of Calgary** seeks an outstanding candidate for a **tenure-track** position at the **Assistant Professor** level, in the **Information Security** area. Details for the position appear at: www.cpsc.ucalgary.ca/career. Applicants must possess a doctorate in Computer Science or a related discipline at the time of appointment, and have a strong potential to develop an excellent research record.

The Department of Computer Science is one of Canada's leaders as evidenced by our commitment to excellence in research and teaching. It has an expansive graduate program and extensive state-of-the-art computing facilities. Calgary is a multicultural city that is the fastest growing city in Canada. Calgary enjoys a moderate climate located beside the natural beauty of the Rocky Mountains. Further information about the Department is available at www.cpsc.ucalgary.ca/.

Interested applicants should send a CV, a concise description of their research area and program, a statement of teaching philosophy, and arrange to have at least three reference letters sent to:

**Dr. Ken Barker**
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada, T2N 1N4
Or via email to: search@cpsc.ucalgary.ca

The applications will be reviewed beginning November 2009 and will continue until the position is filled.

All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority. The University of Calgary respects, appreciates, and encourages diversity.

## University of Delaware

**Software Engineering Faculty Position**
**Department of Computer and Information Sciences Bioinformatics and Computational Biology**
***Faculty Position Center for Bioinformatics & Computational Biology***

Applications are invited for a tenure-track faculty position in Software Engineering to begin Fall 2010. More information is available at http://www.cis.udel.edu/SEposition.

Applications are invited for a tenure-track assistant professor position to begin Fall 2010. We seek applicants from any area of bioinformatics and computational biology, but encourage applicants with interests in computational genomics. More information is available at http://bioinformatics.udel.edu/facultysearch.

## University of Houston-Downtown

**Assistant Professor**

The University of Houston-Downtown invites applications for a tenure-track Assistant Professor position in Computer Science starting Fall 2010. Successful candidates must have a PhD in Computer Science or closely related field in hand by the time of appointment, a promising research profile, and a commitment to excellence in teaching. For more information about the position please visit http://jobs.uhd.edu. Only online applications will be forwarded to the hiring committing for review. AA/EOE

## The University of Michigan, Ann Arbor

**Department of Electrical Engineering and Computer Science**
***Computer Science and Engineering Division Faculty Position***

Applications and nominations are solicited for a faculty position in the Computer Science and Engineering (CSE) Division as part of an interdisciplinary cluster hire funded by the University President to strengthen expertise in the area of Data Mining, Learning, and Discovery with Massive Datasets across multiple departments. Expertise is particularly sought in visual analytics and information visualization.

Candidates with a focus in this area are encouraged to apply. However, all computer science and engineering applications will be considered. Applications must be received by January 11, 2010.

Qualifications include an outstanding academic record, a doctorate or equivalent in computer engineering or computer science, and a strong commitment to teaching and research.

To apply please complete the form at: http://www.eecs.umich.edu/eecs/jobs/csejobs.html

Electronic applications are strongly preferred, but you may alternatively send resume, teaching statement, research statement and names of three references to:

Professor Karem A. Sakallah, Chair, CSE Faculty Search
Department of Electrical Engineering and Computer Science
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109-2121

**The University of Michigan is a Non-Discriminatory/Affirmative Action Employer with an Active Dual-Career Assistance Program. The college is especially interested in candidates who can contribute, through their research, teaching, and/or service, to the diversity and excellence of the academic community.**

## University of North Texas
### Department of Computer Science and Engineering
*Department Chair*

Applications and nominations are invited for the Chair position in the Department of Computer Science and Engineering at the University of North Texas. Candidates must have an earned doctorate in Computer Science and Engineering or a closely related field with a record of significant and sustained research funding and scholarly output that qualifies them to the rank of full professor. Candidates must also demonstrate a record of teaching, research accomplishments, and professional leadership. Preferred: Administrative experience as a department chair or director of personnel working in computer science and engineering; experience in curriculum development; and demonstrated experience mentoring junior faculty. A record of strategic planning and organizational adaptation as well as knowledge of academic standards and procedures required of accrediting agencies is also preferred.

The committee will begin its review of the applications on November 1, 2009 and will continue until the search is closed. For additional information and to apply please visit: http://facultyjobs.unt.edu/applicants/Central?quickFind=50503.

Additional information about the department is available at www.cse.unt.edu.

UNT is an AA/ADA/EOE.

## University of Rochester
### Assistant to Full Professor of Computer Science

The UR Department of Computer Science seeks applicants for a tenure-track position for 2010. Candidates in computer vision, machine learning, networks, security, or algorithms are of particular interest, but strong applicants from all areas of computer science are welcome. Candidates must have a PhD in computer science or related discipline. Senior candidates should have an extraordinary record of scholarship, leadership, and funding.

The Department of Computer Science is one of the best small, research-oriented departments in the nation, with an unusually collaborative culture and strong ties to cognitive science, linguistics, and electrical and computer engineering. Over the past decade, a third of its PhD graduates have won tenure-track faculty positions, and its alumni include leaders at major research laboratories such as Google, Microsoft, and IBM.

The University of Rochester is a private, Tier I research institution located in western New York state. The University of Rochester consistently ranks among the top 30 institutions, both public and private, in federal funding for research and development. Half of its undergraduates go on to post-graduate or professional education. The university includes the Eastman School of Music, a premiere music conservatory, and the University of Rochester Medical Center, a major medical school, research center, and hospital system. The Rochester area features a wealth of cultural and recreational opportunities, excellent public and private schools, and a low cost of living.

Candidates should apply online at http://www.cs.rochester.edu/recruit. Review of applications will begin on Dec. 1, 2009, and continue until all interview openings are filled. The University of Rochester has a strong commitment to diversity and actively encourages applications from candidates from groups underrepresented in higher education. The University is an Equal Opportunity Employer.

## University of South Carolina
### Department Chair - Computer Science and Engineering
*The Department of Computer Science and Engineering*

Department Chair - Computer Science and Engineering The Department of Computer Science and Engineering (www.cse.sc.edu) in the College of Engineering and Computing, University of South Carolina, seeks nominations and applications for the position of Department Chair. The Department offers bachelor's degrees in Computer Engineering, Computer Information Systems, and Computer Science, M.S., M.E. and Ph.D. degrees in Computer Science and Engineering, a Master of Software Engineering, and a Certificate of Graduate Studies in Information Assurance and Security. This is an active and engaged Department with 21 faculty members, including 20 with current research funding and 8 NSF CAREER award winners. Enrollment is over 300 undergraduate and 90 graduate students, including more than 50 doctoral students.

Applicants must have outstanding leadership and administrative skills, and credentials (including a Ph.D. in computer science, computer engineering, or related field) commensurate with appointment as a full professor with tenure. Nomination letters should include statements regarding the nominee's relevant credentials. Applicants should submit a current resume, a statement of professional interests and vision, and the names, affiliations, and contact information of professional references. Applications will be accepted until the position is filled and should be sent by email to

cse-chair-search@cec.sc.edu.

The Department is particularly interested in receiving applications from minorities and women. The University of South Carolina is an affirmative action, equal opportunity employer.

## The University of Texas at San Antonio
### Faculty Positions in Computer Science

The Department of Computer Science at The University of Texas at San Antonio invites applications for tenure/tenure-track positions at the Assistant, Associate or Professor level, starting Fall 2010. **All areas of computer science will be considered.** We are particularly interested in candidates in systems software and algorithms.

Required qualifications: Applicants for an Assistant Professor position must have earned a Ph.D. prior to September 1, 2010, in Computer Science or in a related field and must demonstrate a strong potential for excellence in research and teaching.

Applicants for an Associate Professor position must have a Ph.D. in Computer Science or related field and should have an established research program in their area of specialization.

Applicants for a Full Professor position must have a Ph.D. in Computer Science or a related field and previous experience in graduate and undergraduate teaching and academic program development as well as a recognized and well-funded program of research.

Responsibilities include research, teaching at the graduate and undergraduate levels, and program development. Salary and start-up supporting packages for the positions are highly competitive.

The Department of Computer Science currently has 24 faculty members and offers B.S., M.S., and Ph.D. degrees supporting a dynamic and growing program with 450 undergraduate and 120 graduate students. The research activities and experimental facilities have been well-supported by various federal research and infrastructure grants. See http://www.cs.utsa.edu for additional information on the Department of Computer Science and its faculty.

With over 29,000 students UTSA is the largest university in South Texas. The city of San Antonio has a population of over one million and is known for its rich Hispanic culture, historic attractions, affordable housing, and excellent medical facilities. The Austin-San Antonio corridor is a high-tech center that serves as the home of many major computer companies. Nearby higher education and research institutions include the UT Health Science Center and the Southwest Research Institute.

Applicants must submit a letter of application that identifies the level(s) of the position for which they wish to be considered. Applications must also include a complete dated curriculum vitae (including employment, peer-reviewed publications and grants in chronological order), a statement of research interests, and the names, addresses (postal and e-mail), and telephone numbers of at least three references. Applicants who are selected for interviews must be able to show proof that they will be eligible and qualified to work in the US by the time of hire. These positions are pending budget approval. Screening of applications will begin on January 15, 2010 and will continue until all positions are filled or the search is closed. The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer. Women, minorities, veterans, and individuals with disabilities are encouraged to apply. Applications must be submitted by email in PDF format to fsearch@cs.utsa.edu.

Chair of Faculty Search Committee
Department of Computer Science
The University of Texas at San Antonio
One UTSA Circle
San Antonio, TX 78249-0667
Phone: 210-458-4436 Fax: 210-458-4437
fsearch@cs.utsa.edu

## University of Waterloo

**David R. Cheriton School of Computer Science**
***Cheriton Chairs in Software Systems***

Applications are invited for one or two David R. Cheriton Chairs in Software Systems. These are senior positions and include substantial research support and teaching reduction. Candidates with outstanding research records in software systems (very broadly defined) are encouraged to apply. Successful applicants who join the University of Waterloo are expected to be leaders in research, have an active graduate student program and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2010 calendar year. The Chairs are tenured positions.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: http://www.cs.uwaterloo.ca.

To submit an application, please register at the submission site: http://www.cs.uwaterloo.ca/faculty-recruiting. Once registered, instructions will be provided regarding how to submit your application. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

## University of Waterloo

**David R. Cheriton School of Computer Science**
***Faculty Position in Software Engineering***

The University of Waterloo invites applications for a tenure-track or tenured faculty position in the David R. Cheriton School of Computer Science, in the area of Software Engineering. Candidates at all levels of experience are encouraged to apply. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2010 calendar year.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: http://www.cs.uwaterloo.ca.

To submit an application, please register at the submission site: http://www.cs.uwaterloo.ca/faculty-recruiting. Once registered, instructions will be provided regarding how to submit your application. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

## University of Waterloo

**David R. Cheriton School of Computer Science**
***Faculty Positions in Health Informatics***

The University of Waterloo invites applications for one or two tenure-track or tenured faculty positions in the David R. Cheriton School of Computer Science, in the area of Health Informatics. We define health informatics broadly to include medical informatics and biomedical systems. The School plans to start a new graduate degree program in health informatics in September 2010.

Candidates at all levels of experience are encouraged to apply. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2010 calendar year.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: http://www.cs.uwaterloo.ca.

To submit an application, please register at the submission site: http://www.cs.uwaterloo.ca/faculty-recruiting. Once registered, instructions will be provided regarding how to submit your application. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

## University of Wisconsin - Platteville

**Assistant Professor**

The Department of Computer Science and Software Engineering at the University of Wisconsin-Platteville invites applications for two tenure-track faculty positions starting fall, 2010.

One position will be in Computer Science, the other in Software Engineering. Candidates applying for the Software Engineering position must have a Ph.D. in Software Engineering, Computer Science, or closely related field. The primary duty for this position will be teaching courses in an ABET-accredited software engineering program. Candidates applying for the Computer Science position must have a Ph.D. in Computer Science or closely related field such as Information Systems. The primary duty for this position will be teaching courses in Computer Science.

Candidates must have excellent verbal and written communication skills and a strong commitment to teaching. Commitment to scholarly and professional activities is also required. The candidate must have a demonstrated commitment to or experience with racially diverse populations. Salary will be competitive and commensurate with experience and qualifications. In addition, there are consulting opportunities with local companies.

Send a letter of application, undergraduate & graduate transcripts, a statement illustrating your commitment to fostering campus racial diversity, and a vita including references to Search and Screen Committee, Department of Computer Science and Software Engineering, 207 Ullrich Hall, 1 University Plaza, Platteville, WI 53818-3099 or to stutenbm@uwplatt.edu. Visit our web site at http://www.uwplatt.edu/~csse for more information. Review of applications will start January 4, 2010 and will continue until suitable candidates are found.

UW-Platteville, an equal opportunity, affirmative action employer, seeks to build a diverse faculty and staff and encourages applications from women and persons of color. The names of all nominees and applicants who have not requested in writing that their identities be kept confidential, and of all finalists, will be released upon request. Employment will require a criminal background check.

## University of Wisconsin – Milwaukee
**Health Informatics Professor**

9-month tenure-track Assistant, Associate or Full Professor. Program planning and management, teaching, advising students, developing and maintaining an active line of research and seeking extramural funding. Apply URL: http://www.jobs.uwm.edu/applicants/Central?quickFind=50785

## Ursinus College
**Assistant Professor of Computer Science**

Ursinus College, a liberal arts college located in the suburbs of Philadelphia, invites applications for a tenure-track position in Computer Science. Please see http://www.ursinus.edu/NetCommunity/Page.aspx?pid=390 for more information.

## Virginia Tech
**Department of Computer Science**
***Artificial Intelligence/Machine Learning Senior Position***

The Department of Computer Science at Virginia Tech (www.cs.vt.edu) invites applications for a full-time tenured position at the Professor or Associate Professor rank from candidates in artificial intelligence with particular interests in machine learning, knowledge representation, or data mining. Candidates should have an established record of scholarship, leadership, and collaboration in computing and interdisciplinary areas; demonstrated ability to contribute to teaching at the undergraduate and graduate levels in AI and related subjects; sensitivity to issues of diversity in the campus community; and the skills needed to establish and grow a multidisciplinary research group.

CS@VT has over 40 tenure-track research-oriented faculty. PhD production is among the top 30 in the US and annual research expenditures exceed $6 million. There are rich opportunities in a highly collaborative department with strengths in HCI, HPC, CS education, digital libraries, computational biology and bioinformatics. Active interdisciplinary research also explores CyberArts, digital government, problem-solving environments. Emphases on security and personal health informatics are underway in collaboration with the newly formed VT-Carilion Research Institute associated with the VT-Carilion School of Medicine, opening in Fall 2010.

CS@VT is part of the College of Engineering (www.eng.vt.edu) in a comprehensive research university with more than 26,000 students. The main campus is in Blacksburg, which is consistently ranked among the country's best places to live (http://www.liveinblacksburg.com/).

Salary for suitably qualified applicants is competitive and commensurate with experience. Virginia Tech is an Equal Opportunity/Affirmative Action Institution.

Applications must be submitted online to https://jobs.vt.edu for posting #090529. Applicant screening will begin January 15, 2010 and continue until the position is filled. Inquiries should be directed to Dennis Kafura, Hiring Committee Chair, kafura@cs.vt.edu.

## Washington University in Saint Louis
**Multiple tenure-track/tenured faculty positions**

The Department of Computer Science and Engineering (CSE) and the School of Medicine (WUSM) are jointly searching for multiple tenure-track faculty members with outstanding records of computing research and a serious interest in collaborative research on problems related to biology and/or medicine. Appointments may be made wholly within CSE or jointly with the Departments of Medicine or Pathology & Immunology.

A key initiative in the CSE Department's strategic plan is Integrating Computing and Science. As part of that initiative, we expect to make synergistic hires with a combined research portfolio spanning the range from fundamental computer science/engineering to applied research focused on science or medicine. Specific areas of interest include, but are not limited to:

- Analysis of complex genetic, genomic, proteomic, and metabolomic datasets;
- Theory/Algorithms with the potential for biomedical applications;
- Image analysis or visualization with the potential for biomedical applications;
- Databases, medical informatics, clinical or public-health informatics;
- Computer engineering with applications to medicine or the natural sciences;
- All areas of computational biology and biomedical informatics

These positions will continue a successful, ongoing strategy of collaborative research between CSE and the School of Medicine, which is consistently ranked among the top 3 medical schools in the United States. CSE currently consists of 24 tenured and tenure-track faculty members, 71 Ph.D. students, and a stellar group of undergraduates with a history of significant research contributions. The Department seeks to build on and complement its strengths in biological sequence analysis, biomedical image analysis, and biomedical applications of novel computing architectures. Exceptional candidates conducting research in other areas of Computer Science are also encouraged to apply.

Washington University is a private university with roughly 6,000 full-time undergraduates and 6,000 graduate students. It has one of the most attractive university campuses anywhere, and is located in a lovely residential neighborhood, adjacent to one of the nation's largest urban parks, in the heart of a vibrant metropolitan area. St. Louis is a wonderful place to live, providing access to a wealth of cultural and entertainment opportunities without the everyday hassles of the largest cities.

We anticipate appointments at the rank of Assistant Professor; however, in the case of exceptionally qualified candidates appointments at any rank may be considered. Applicants must have a Ph.D. in computer science, computer engineering, electrical engineering, biomedical engineering, or a closely related field and a record of excellence in teaching and research appropriate to the appointment level. The selected candidate is expected to build an externally-supported research program, teach and mentor students at the graduate and undergraduate levels, and foster interdisciplinary interactions with colleagues throughout the university. Candidates who would contribute to enhancing diversity at the departmental and university levels are strongly encouraged to apply. Applications from academic couples are welcomed and encouraged.

Qualified applicants should submit a complete application (cover letter, curriculum vita, research statement, teaching statement, and names of at least three references) electronically by following the directions provided at http://cse.wustl.edu/faculty-recruiting/. Other communications may be directed to Prof. Michael Brent, Department of Computer Science and Engineering, Campus Box 1045, Washington University, One Brookings Drive, St. Louis, MO 63130-4899.

Applications submitted before January 31, 2010 will receive full consideration. Washington University is an equal opportunity/affirmative action employer.

## Wellesley College
**Computer Science Department**
***Norma Wilentz Hess Visiting Assistant Professor***

Wellesley College invites applications for a two-year Norma Wilentz Hess Fellowship in the Department of Computer Science. Funded at the Visiting Assistant Professor level, this fellowship will help Wellesley maintain a flexible Computer Science curriculum that explores interdisciplinary learning and new directions of special promise. With a teaching load of one course per semester, the Hess Fellow will have ample opportunity for innovative course development and collaborative teaching and research projects with Wellesley faculty and students. The fellowship also includes support for travel, research, equipment, conference attendance, and other academic activities. For more information about the department and the college, please visit http://cs.wellesley.edu.

Applicants should have a Ph.D (or be close to completion) in Computer Science or a related discipline. Strong candidates in any area of specialty will be considered. We especially encourage applicants in interdisciplinary fields such as artificial intelligence, bioinformatics, human/computer interaction, and ethical, educational, social, and legal aspects of computing.

Interested applicants are requested to submit a letter of application, curriculum vitae, and names and email addresses of three references through our on-line application system at https://career.wellesley.edu. Include a statement discussing teaching philosophy and ideas for course development and research projects. Applications will be reviewed starting January 11, 2010. If circumstances make it impossible to submit any materials electronically, you may email working@wellesley.edu for assistance. Questions concerning the position should be directed by email to Randy Shull at rshull@wellesley.edu.

Wellesley College is an Affirmative Action/Equal Opportunity Employer, and we are committed to increasing the diversity of the college community and the curriculum. Candidates who believe they can contribute to that goal are encouraged to apply.

# Indiana University
## School of Informatics and Computing
### Faculty Positions

The School of Informatics and Computing at Indiana University, Bloomington, invites applications for faculty positions. We are particularly interested in tenure-track candidates in the areas of:

- Health Informatics
- Cyber-infrastructure
- Software and Systems
- Data Mining, Machine Learning and Search
- Artificial Intelligence
- Human-Computer Interaction
- Graphics and Visualization; and
- Algorithms

Most positions are expected to be filled at the junior level but outstanding senior candidates will be considered, particularly in the broad area of cyber-infrastructure, software and systems.

The IU Bloomington School of Informatics and Computing is the first of its kind and among the largest in the country, with a faculty of more than 60 full time members, more than 400 graduate students, and widely subscribed undergraduate programs. It offers undergraduate degrees in Computer Science and in Informatics, M.S. degrees in Computer Science, Bioinformatics, Human Computer Interaction Design, and Security Informatics, and Ph.D. degrees in Computer Science and in Informatics. The School has received public recognition as a "top-ten program to watch" (Computerworld) thanks to its excellence and leadership in academic programs, interdisciplinary research, placement, and outreach. The school offers excellent work conditions, including attractive salaries and research support, low teaching loads, and world-class computing and library facilities. The school continues strong student growth, over 30% in the last two years. The school hired 5 new faculty last year and plans to hire a similar number this year.

Located in the wooded, rolling hills of southern Indiana, Bloomington is a culturally thriving college town with a moderate cost of living. It is renowned for its top-ranked music school, performing and fine arts, historic campus, cycling traditions, active lifestyle, and natural beauty.

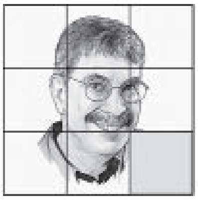
Applicants should submit a curriculum vitae, a statement of research and teaching, and the names of three references (six for associate and full professor candidates) using the recruit link at http://informatics.indiana.edu/hiring (preferred) or by mail to:

Chair, Faculty Search Committee
School of Informatics and Computing
919 E 10th Street
Bloomington, IN 47408

Questions may be sent to:
hiring@informatics.indiana.edu.

To receive full consideration completed applications must be received by December 31, 2009. However, applications received after this date will be considered until all positions are filled.

Indiana University is an Equal Opportunity/Affirmative Action employer. Applications from women and minorities are strongly encouraged.



DOI:10.1145/1610252.1610278

Peter Winkler

# Puzzled
# Solutions and Sources

*Last month (November 2009, p. 112) we posted a trio of brain teasers, including one as yet unsolved, concerning the covering of a plane.*

## 1. Identical Coins on Rectangular Table.

*Solution.* Let us observe that if we double the radius (say, from one to two inches) of each of the original coins, the result would be to cover the whole table. Why? Well, if a point $P$ isn't covered, it must be two inches or more from any coin center; thus a one-inch coin placed with its center at $P$ would fit in the original configuration.

If we could replace each big coin with four small coins covering the same area, we'd be done… but we can't cover a big coin with four small coins. Instead, let us note that rectangles have the property that they can be partitioned into four copies of themselves. So, we shrink the whole picture (of big coins covering the table) by a factor of two in each dimension and use four copies of the new picture to cover the original table.

Surprisingly (perhaps), this lovely but seemingly crude argument gives the best possible factor: Replace the factor 4 with anything smaller, say 3.99, and the statement of the puzzle (with 100 and 400 replaced by $n$ and $3.99n$) is no longer true.

The puzzle came to me by way of computer scientist Guy Kindler of the School of Computer Science and Engineering in the Hebrew University of Jerusalem during a marvelous visiting year (2003–2004) by each of us at the Institute for Advanced Study in Princeton, NJ.

## 2. Two Sets of Parallel Lines on Infinite Plane.

*Solution.* The puzzle asked whether we could cover each point of the plane exactly twice using a set of lines that contains lines in more than two different directions.

The solution will disappoint some of us. The answer is yes, assuming the Axiom of Choice, which allows us to make choices at every step of an infinite process. But the proof presented here requires transfinite induction(!), leaving us without any geometry we can wrap our mind around. The problem (and its solution) were provided to me by mathematical physicist Senya Shlosman of the Centre de Physique Theorique, Marseille, France, who is unaware of its origin.

Nonetheless, the solution appeals to me as an example of an easy application of a powerful tool. The idea is that we start off with three lines that cross one another (so we already have three directions). Roughly speaking, at each moment in our algorithm we have constructed a finite or countable number of lines, with no point covered twice; we then pick a point on the plane that is not doubly covered and add a line through that point. How do we know we can do this without triple-covering some other point? Because the number of *pairs* of lines so far constructed is countable, only countably many points are double-covered, and we have an uncountable number of angles at which the new line can be placed.

Does this seem like cheating? It should. First, those of us who have used transfinite induction know that details have been omitted (but easily supplied). More significant, there is no way we can do this construction in a useful manner. Does this mean there isn't some clever, effective, or at least imaginable way to double-cover the plane with lines, other than by two parallel classes? I haven't found one. Neither has Shlosman. Perhaps you can.

## 3. Colored Discs on Infinite Plane.

*Conjecture.* This open puzzle supposed that we were given a collection of open-unit discs that is a thousand-fold cover of the plane; that is, every point of the plane is covered by at least 1,000 discs. Our job was to prove (or disprove) that we can color each disc red or blue in such a way that the red discs cover the plane and the blue discs also cover the plane.

No solutions have been offered so far. Maybe this is just fundamentally very difficult to prove. Maybe it isn't even true. But it seems reasonable, don't you think?

**All readers are encouraged to submit prospective puzzles for future columns to puzzled@cacm.acm.org.**

**Peter Winkler** (puzzled@cacm.acm.org) is Professor of Mathematics and of Computer Science and Albert Bradley Third Century Professor in the Sciences at Dartmouth College, Hanover, NH.

Future Tense, one of the revolving features on this page, presents stories and essays from the intersection of computational science and technological speculation, their boundaries limited only by our ability to imagine what will and could be.

DOI:10.1145/1610252.1610279

Joe Haldeman

# Future Tense
# Mightier Than the Pen

*The computer reconfigures me.*

I'VE BEEN A writer for 40 years, and my attitude toward writing machines has never been simple. Right now I'm sitting in a quiet one-room cabin in the Maine woods, writing in natural light with a fountain pen into a bound blank book. I could have done that in the 19th century—or 5,000 years ago, if you trade the fountain pen for a quill or stylus.

But when the day's writing is done, I pull out my featherweight MacBook Air and type the words from the book into a text file. Then I port it over to NeoOffice to get a word count for the day.

I have various reasons to make it this complicated, but the main one is simple: that hand-written draft is a true first draft. You can name a computer file "first draft," but there's no way to prove it was first. In fact, you would be a very strange writer if you didn't go back and forth, changing things, long before you finished the draft.

Why should that make any difference? Well, as a practical matter, sometimes you do get something right the first time. After a book's gone through a half-dozen rewrites, it doesn't hurt to look back and see whether the changes were all for the better.

There's also the archival and financial value of the physical document itself. Even for a moderately well-known writer like myself, the holographic (handwritten) manuscript of a novel is worth thousands or tens of thousands of dollars. I've twice put handwritten drafts of short stories up on eBay, and collectors paid more than I got from the magazine that printed them. In addition, if you're egotistical enough to want your work to last after you die, it's not a bad strategy to leave behind lots of material for scholars—so they'll write papers on you and force their students to buy your books.

I get a lot of esthetic, or at least craft, pleasure out of writing an actual book. In front of my window in this little cabin is a cup holding a dozen different fountain pens and four different bottles of ink. Each combination has its own personality, and it's fun to start out the day making that small decision.

I've written about 20 books with this collaboration between pen and computer, and although I love my pens and blank books with hobbyist zeal, if I had to choose between them and the computer there would be no contest. The pens would have to go, even though they're so familiar they're like part of my hand. The computer is part of my brain. It has reconfigured me.

When I started writing, there weren't any out-of-the-box word processors within the budget of a freelance writer. (Stephen King had an industrial-strength Wang with a screen on a robot arm coming down from the ceiling.) If you were good with hobbyist electronics, you could cobble something together from Ohio Scientific or Radio Shack. But like most writers, I had to wait until around 1980, when the Apple II came out. I bought one of the first, along with a printer about the size and weight of a VW Beetle. It used Doctor Memory as a word processor; the letters were all screaming capitals, with actual caps showing in eye-straining white-on-black.

**Sometime during the Age of Primates the Internet got into the evolutionary stroll.**

By today's standards it was crude and slow, and it took a half-dozen big fragile floppy disks to store a novel. But what a feeling of power.

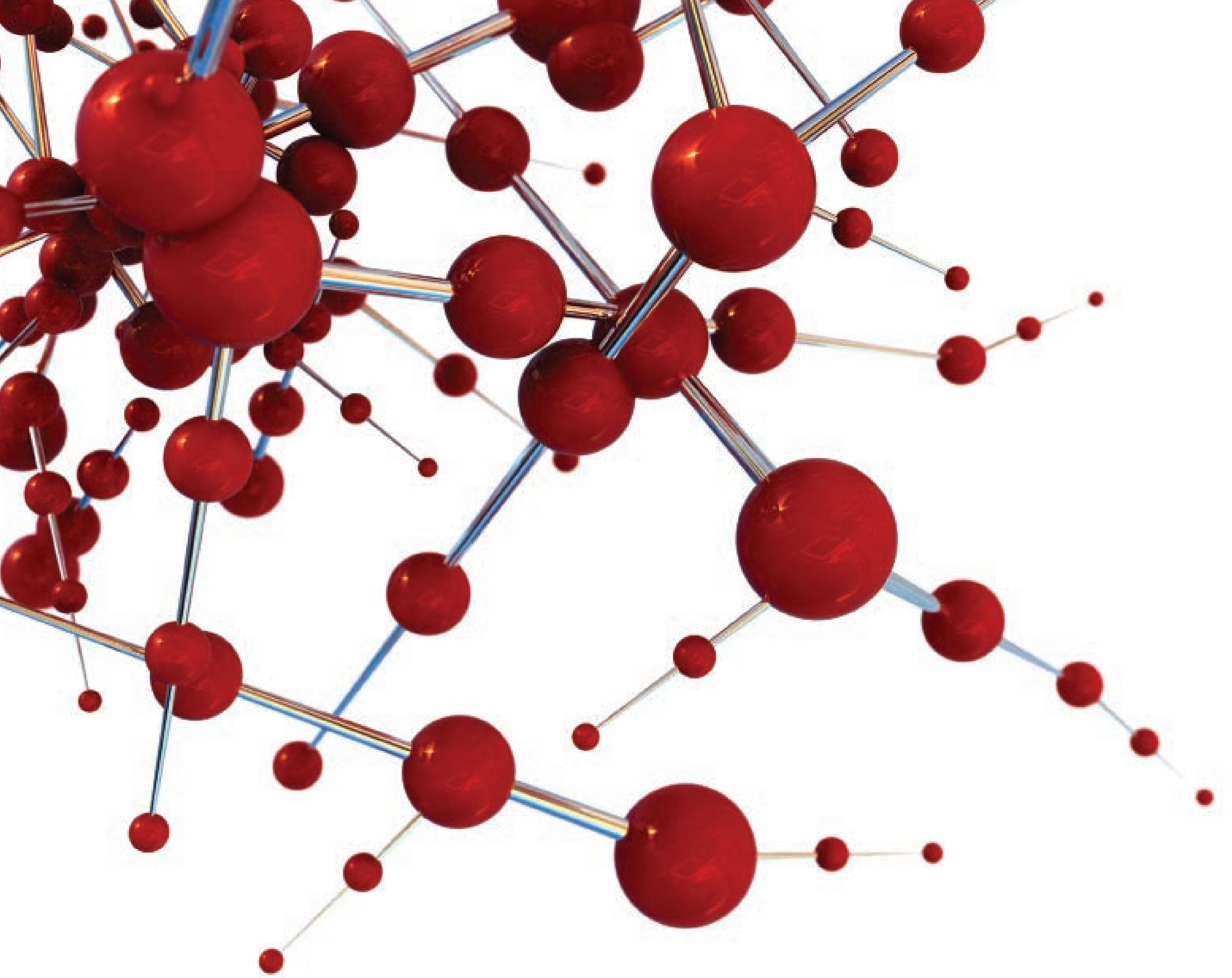
At least when it worked. You heard horror stories about a person touching the wrong key and losing an entire novel. I once lost a chapter by hitting "Backspace" at the wrong time. I told my friends that this new machine was really great; no mere electric typewriter could leap off your desk into a filing cabinet and eat a chapter of your novel.

One triumphant day I forced the top off the Apple II, I think voiding the warranty, and swapped out a couple of chips and soldered in a third-party component—and suddenly I had lower-case letters, black on white, just like a typewriter but infinitely malleable. From then on I was crawling up the beach, the fish gulping air becoming the lizard becoming the mammal and then the ape. Sometime during the Age of Primates the Internet and Web got into the evolutionary stroll, and the computer was no more a fancy typewriter than a human is a fancy chimp.

About 15 discarded machines litter that evolutionary beach, and it's a long way from the Sinclair that whistled into a tape recorder to the superlight Frisbee of a Mac Air that talks to satellites and brings a huge and complex universe into my rustic little shack.

Evolution doesn't end. The ape became the human, and now the human is drawing pictures in the sand. **C**

**Joe Haldeman** (haldeman@earthlink.net) is an adjunct professor in the Program in Writing and Humanistic Studies at the Massachusetts Institute of Technology, Cambridge, MA. His most recent novel is *Marsbound*. His novel *Camouflage* won the Nebula Award for Best Novel in 2005.



CALL FOR PAPERS

http://fdg2010.org

FDG

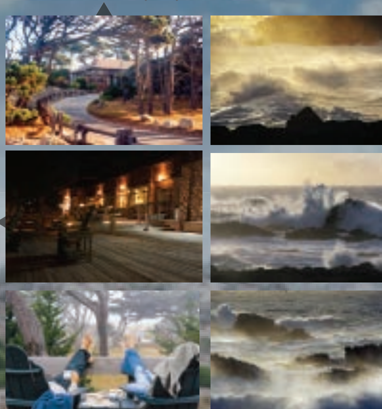
the International Conference on the

# Foundations of Digital Games

June 19 - 21 2010

(cc) (by =) Veronica Vale
www.flickr.com/people/vbv/

(cc) (by $ =) Andrew Fitzhugh
www.flickr.com/people/fitzhugh



(cc) (by $ =) Steve Schwartz
www.flickr.com/people/vasculata

*Submission Deadlines*

***Papers & Posters**: 5th Feb*

***Doctoral Consortium**: 12th Feb*

***Demos**: 2nd April*

*Asilomar Conference Grounds*

*Monterey, California, USA.*

*Learning in Games*

***Infrastructure** (Databases, Networks, Security)*

*Graphics & Interfaces*

*Game Design*

*Artificial Intelligence*

*Computer Science & Games Education*

*Game Studies*

| | |
|---|---|
| Conference Chair: | **Ian Horswill**, Northwestern University |
| Program Chair: | **Yusuf Pisan**, University of Technology, Sydney |
| Doctoral Consortium Chair: | **Zoran Popovic**, University of Washington |
| Workshops Chair: | **Michael Mateas**, University of California, Santa Cruz |
| Panels Chair: | **Ian Bogost**, Georgia Institute of Technology |
| Tutorials Chair: | **Robin Hunicke**, That Game Company |
| Industrial Relations Chair: | **Hiroko Osaka**, Northwestern University |
| Local Arrangements Chair: | **Marilyn Walker**, University of California, Santa Cruz |
| Webmaster: | **Karl Cheng-Heng Fua**, Northwestern University |

An Official Conference of the
**Society for the Advancement of the Science of Digital Games**

SASDG

FDG 2010 Supported by:

Microsoft Research