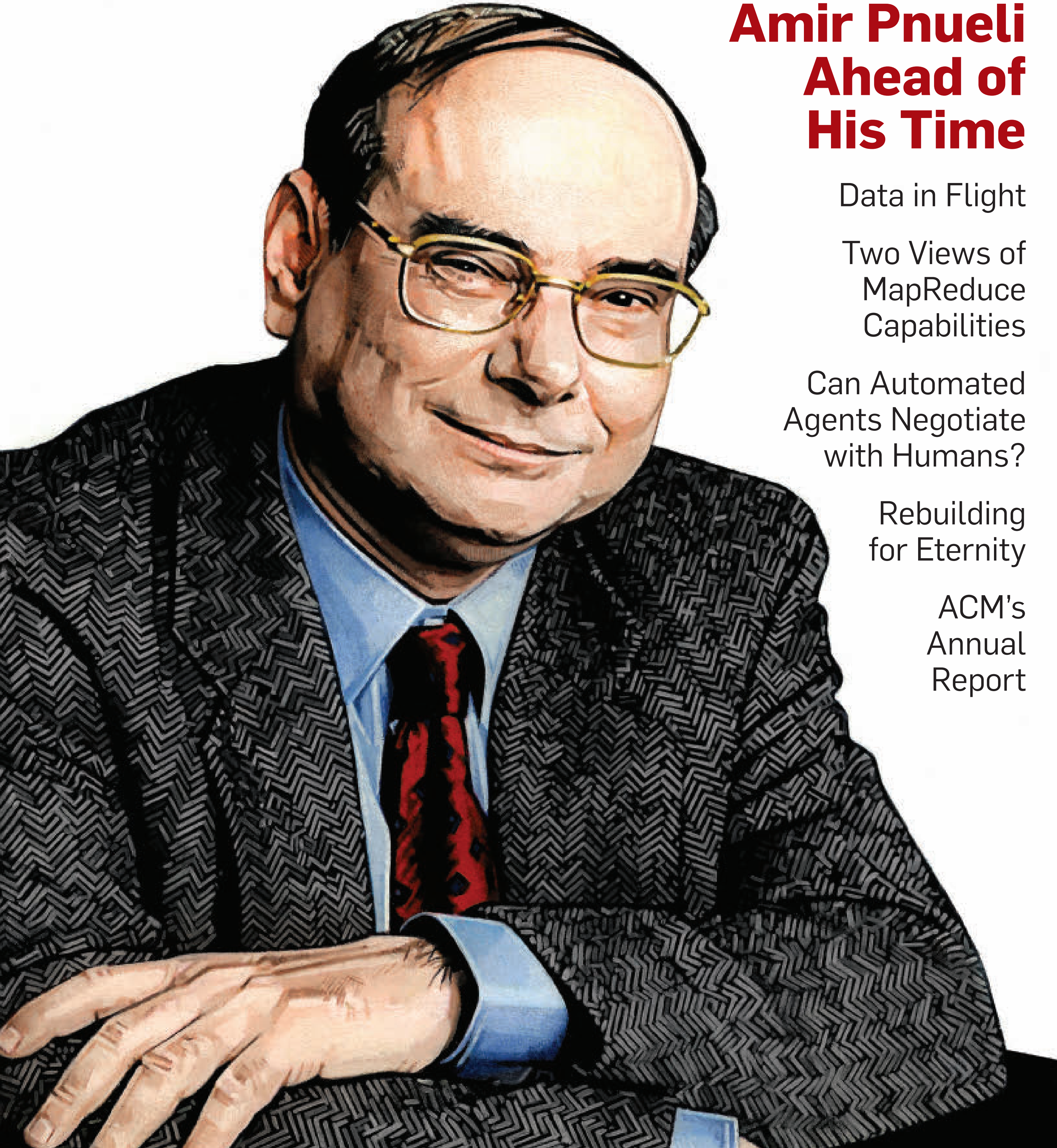
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

01/2010 VOL.53 NO.01



Amir Pnueli Ahead of His Time

Data in Flight

Two Views of MapReduce Capabilities

Can Automated Agents Negotiate with Humans?

Rebuilding for Eternity

ACM's Annual Report

Group Term Life Insurance**

10- or 20-Year Group Term Life Insurance*

Group Disability Income Insurance*

Group Accidental Death & Dismemberment Insurance*

Group Catastrophic Major Medical Insurance*

Group Dental Plan*

Long-Term Care Plan

Major Medical Insurance

Short-Term Medical Plan***

Who has time to think about insurance?

Today, it's likely you're busier than ever. So, the last thing you probably have on your mind is whether or not you are properly insured.

But in about the same time it takes to enjoy a cup of coffee, you can learn more about your ACM-sponsored group insurance program — a special member benefit that can help provide you financial security at economical group rates.

Take just a few minutes today to make sure you're properly insured.

Call Marsh Affinity Group Services at **1-800-503-9230** or visit **www.personal-plans.com/acm.**

3132851 35648 (7/07) © Seabury & Smith, Inc. 2007

The plans are subject to the terms, conditions, exclusions and limitations of the group policy. For costs and complete details of coverage, contact the plan administrator. Coverage may vary and may not be available in all states.

*Underwritten by The United States Life Insurance Company in the City of New York, a member company of American International Group, Inc.

**Underwritten by American General Assurance Company, a member company of American International Group, Inc.

***Coverage is available through Assurant Health and underwritten by Time Insurance Company.

AG5217

MARSH
Affinity Group Services
a service of Seabury & Smith

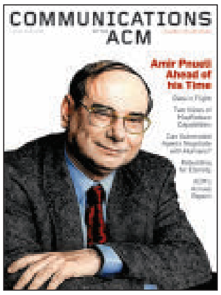
COMMUNICATIONS OF THE ACM

Departments

Last Byte

News

Viewpoints



About the Cover: Amir Pnueli, recipient of the 1996 ACM A.M. Turing Award, is remembered by friends and colleagues as a pioneer in formal specification and verification. His cover portrait is the work of Chris Kasch, a U.K.-based illustrator. For more about Kasch's work, visit http://www.chriskasch.co.uk.

01/2010
VOL. 53 NO. 01

Practice

Article development led by acmqueue
queue.acm.org

Review Articles

ILLUSTRATION BY RYAN ALEXANDER

Contributed Articles

Research Highlights

Virtual Extension

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

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

Association for Computing Machinery
Advancing Computing as a Science & Profession

COMMUNICATIONS OF THE ACM

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.

Executive Director and CEO
John White
Deputy Executive Director and COO
Patricia Ryan
Director, Office of Information Systems
Wayne Graves
Director, Office of Financial Services
Russell Harris
Director, Office of Membership
Lillian Israel
Director, Office of SIG Services
Donna Cappo

ACM COUNCIL

President
Wendy Hall
Vice-President
Alain Chesnais
Secretary/Treasurer
Barbara Ryder
Past President
Stuart I. Feldman
Chair, SGB Board
Alexander Wolf
Co-Chairs, Publications Board
Ronald Boisvert, Holly Rushmeier
Members-at-Large
Carlo Ghezzi;
Anthony Joseph;
Mathai Joseph;
Kelly Lyons;
Bruce Maggs;
Mary Lou Soffa;
Fei-Yue Wang
SGB Council Representatives
Joseph A. Konstan;
Robert A. Walker;
Jack Davidson

PUBLICATIONS BOARD

Co-Chairs
Ronald F. Boisvert and Holly Rushmeier
Board Members
Jack Davidson; Nikil Dutt; Carol Hutchins; Ee-Peng Lim; Catherine McGeoch; M. Tamer Ozsu; Vincent Shen; Mary Lou Soffa; Ricardo Baeza-Yates

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

STAFF

GROUP PUBLISHER
Scott E. Delman
publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Jack Rosenberger
Web Editor
David Roman
Editorial Assistant
Zarina Strakhan
Rights and Permissions
Deborah Cotton

Art Director
Andrij Borys
Associate Art Director
Alicia Kubista
Assistant Art Director
Mia Angelica Balaquiot
Production Manager
Lynn D'Addesio
Director of Media Sales
Jennifer Ruzicka
Marketing & Communications Manager
Brian Hebert
Public Relations Coordinator
Virgina Gold
Publications Assistant
Emily Eng

Columnists
Alok Aggarwal; Phillip G. Armour; Martin Campbell-Kelly; Michael Cusumano; Peter J. Denning; Shane Greenstein; Mark Guzdial; Peter Harsha; Leah Hoffmann; Mari Sako; Pamela Samuelson; Gene Spafford; Cameron Wilson

CONTACT POINTS

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

WEB SITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/guidelines

ADVERTISING

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

Director of Media Sales
Jennifer Ruzicka
jen.ruzicka@hq.acm.org

Media Kit acmmediasales@acm.org

EDITORIAL BOARD

EDITOR-IN-CHIEF
Moshe Y. Vardi
eic@cacm.acm.org

NEWS
Co-chairs
Marc Najork and Prabhakar Raghavan
Board Members
Brian Bershad; Hsiao-Wuen Hon; Mei Kobayashi; Rajeev Rastogi; Jeannette Wing

VIEWPOINTS
Co-chairs
Susanne E. Hambrusch; John Leslie King; J Strother Moore
Board Members
P. Anandan; William Aspray; Stefan Bechtold; Judith Bishop; Stuart I. Feldman; Peter Freeman; Seymour Goodman; Shane Greenstein; Mark Guzdial; Richard Heeks; Rachelle Hollander; Richard Ladner; Susan Landau; Carlos Jose Pereira de Lucena; Beng Chin Ooi; Loren Terveen

PRACTICE
Chair
Stephen Bourne
Board Members
Eric Allman; Charles Beeler; David J. Brown; Bryan Cantrill; Terry Coatta; Mark Compton; Benjamin Fried; Pat Hanrahan; Marshall Kirk McKusick; George Neville-Neil

The Practice section of the CACM Editorial Board also serves as the Editorial Board of acmqueue.

CONTRIBUTED ARTICLES
Co-chairs
Al Aho and Georg Gottlob
Board Members
Yannis Bakos; Gilles Brassard; Alan Bundy; Peter Buneman; Ghezzi Carlo; Andrew Chien; Anja Feldmann; Blake Ives; James Larus; Igor Markov; Gail C. Murphy; Shree Nayar; Lionel M. Ni; Sriram Rajamani; Jennifer Rexford; Marie-Christine Rousset; Avi Rubin; Abigail Sellen; Ron Shamir; Marc Snir; Larry Snyder; Veda Storey; Manuela Veloso; Michael Vitale; Wolfgang Wahlster; Andy Chi-Chih Yao; Willy Zwaenepoel

RESEARCH HIGHLIGHTS
Co-chairs
David A. Patterson and Stuart J. Russell
Board Members
Martin Abadi; Stuart K. Card; Deborah Estrin; Shafi Goldwasser; Monika Henzinger; Maurice Herlihy; Norm Jouppi; Andrew B. Kahng; Gregory Morrisett; Michael Reiter; Mendel Rosenblum; Ronitt Rubinfeld; David Salesin; Lawrence K. Saul; Guy Steele, Jr.; Gerhard Weikum; Alexander L. Wolf

WEB
Co-chairs
Marti Hearst and James Landay
Board Members
Jason I. Hong; Jeff Johnson; Greg Linden; Wendy E. MacKay

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

Printed in the U.S.A.

DOI:10.1145/1629175.1629176

Moshe Y. Vardi

More Debate, Please!

In the May 1979 issue of *Communications*, a powerfully written article by Richard A. De Millo, Richard J. Lipton, and Alan J. Perlis entitled "Social Processes and Proofs of Theorems and Programs," argued that formal verification of programs is "difficult to justify and manage." The article created the perception, in the minds of many computer scientists, that formal verification is a futile area of computing research.

That article did not cite a 1977 paper by Amir Pnueli entitled "The Temporal Logic of Programs." His paper had attracted little attention by 1979, but by 1997 it would be described as a "landmark paper" in the citation that accompanied Pnueli's 1996 ACM A.M. Turing Award. In his paper, Pnueli, whose sudden and unexpected death on Nov. 2, 2009 shocked the computer science community, laid the foundation for formal verification of concurrent and reactive programs. (An article describing Pnueli's scientific legacy appears on page 22.) The paper also laid the foundation for the development of model checking, an automated formal-verification technique for which Edmund A. Clarke, E. Allen Emerson, and Joseph Sifakis received the 2007 ACM Turing Award.

With hindsight of 30 years, it seems that De Millo, Lipton, and Perlis' article has proven to be rather misguided. In fact, it is interesting to read it now and see how arguments that seemed so compelling in 1979 seem so off the mark today. Should we infer that *Communications* erred in publishing that article? My answer is a resounding "no!"

My basic education included exposure to Talmudic scholarship. Jewish scholars in the first half of the first millennium believed that truth will emerge from vigorous debate. The Talmud, a monumental work of Jewish scholarship concluded circa 500 CE, is in essence a compendium of legal debates. Vigorous debate, I believe, exposes all sides of an issue—their strengths and weaknesses. It helps us to reach more knowledgable conclusions. To quote Benjamin Franklin: "When Truth and Error have fair Play, the former is always an overmatch for the latter." In my opinion, however, the editors of *Communications* in 1979 did err in publishing an article that can fairly be described as tendentious without publishing a counterpoint article in the same issue. Indeed, the article instigated so many reader responses, the editors published 10 pages of letters in the November 1979 Forum section of *Communications*, calling the work everything from "marvelous" to "humorous."

In 2007, when I met with various focus groups to discuss the relaunching of *Communications*, I was encouraged to keep this publication engaged in controversial topics. "Let blood spill over the pages of *Communications*," said one discussant jokingly. At the same time, however, participants believed that the magazine should represent all points of view fairly. This sentiment led to the establishment of the Point-Counterpoint feature, in which both sides of an issue are represented by opposing articles. Quoting Franklin again: "when Men differ in Opinion, both Sides ought equally to have the Advantage of being heard by the Publick."

Since the relaunch in July 2008, we have published several Point-Counterpoint pairs: on computing curricula, e-voting, Net neutrality, and the direction of CS education in the U.S. At this point, however, the pipeline for such articles is dry. I had assumed that both members of the editorial board and readers would propose topics for Point-Counterpoint articles, but that does not seem to be the case. It is almost as if people believe there is something improper about engaging in direct debate. In fact, several authors whom I invited to participate in Point-Counterpoint debates have declined in order to avoid head-on confrontation. The truth is, however, that there are many issues in computing that inspire differing opinions. We would be better off highlighting the differences rather than pretending they do not exist.

In this issue of *Communications* we have a debate that is quite a rarity in computing research: a *technical* debate. MapReduce (MR) is a software framework to support distributed computing on large data sets on computer clusters. It was introduced by J. Dean and S. Ghemawat of Google in a highly influential 2004 article, and featured as a Research Highlight paper in the January 2008 issue of *Communications*. The success of MapReduce led some to claim that the extreme scalability of MR will "relegate relational database management systems (RDBMS) to the status of legacy technology." A pair of Contributed Articles in this issue—Dean and Ghemwat on one side and Stonebraker et al. on the other—debate the relative merits of MR and RDBMS beginning on page 64. As parallel computation is one of the hottest topics in computing today, I have no doubt that our readers will find this technical debate highly instructive.

If you have topics that you think should be debated on the pages of *Communications*, please contact me. More debate, please!

Moshe Y. Vardi, EDITOR-IN-CHIEF

DOI:10.1145/1629175.1629178

Software Still As Much an Art As Science

C.A.R. HOARE'S VIEWPOINT "Retrospective: An Axiomatic Basis for Computer Programming" (Oct. 2009) reminded me of a saying attributed to both Jan L.A. van de Snepscheut and Yogi Berra: "In theory, there is no difference between theory and practice. But, in practice, there is." I recall as an undergraduate the elegance of using induction to prove that a recursive program was correct. Induction and recursion were two sides of the same coin, one theory, the other practice.

I've been a software engineer for almost 25 years. Though I've used axiomatic techniques designing, implementing, and debugging my code, no project (as a whole) could possibly rely on it. Any form of axiomatic verification requires a rock-solid foundation on which to argue the correctness of an implementation. Programming with well-defined functionality (such as data-structure manipulation) can be verified axiomatically, but as a project's size and complexity grow, its behaviors become less rigorous. Testing doesn't verify that the functionality of a large project is correct, only what the interpretation of that functionality should be.

Customers are rarely able to define what they want but tend to know it when they see (or don't see) it. This makes it difficult for programmers to create complete, consistent, unambiguous requirements. Early in my career, I used highlighter pens to extract requirements from whitepapers. Fortunately, requirements today are enumerated, version-controlled, and placed within context via use cases. Still, there are too many details and too few systems-engineering resources to document every behavior, condition, and exception.

Since an implementation must be able to handle every case, developers must make assumptions that cannot always be confirmed. Axiomatic confirmation and unit testing by developers verify code only as long as the assumptions hold true. Verification is needed by independent testers who can ignore the implementation but must be sure the product matches their interpretation of requirements. For this reason alone, software is still as much an art as it is a science.

Jim Humelsine, Neptune, NJ

DIY Biological (Nervous) Systems

Congratulations to Corrado Priami for sharing his insight into biological simulations in his article "Algorithmic Systems Biology" (May 2009). My own interest in emulating biological systems has yielded similar conclusions. In addition to computerized algorithmic representations in software, I've designed analog component circuits and linear coprocessors using operational amplifiers, including integrate-and-fire artificial neurons based on Hodgkin's and Huxley's research, synthetic emotion-processing neurons using sum-and-difference operational amplifiers, and artificial neural networks for machine vision. These components add instantaneous analog parallelism to the digital computer's software concurrency, as Priami said.

For the past 10 years I've been developing a fairly elaborate nervous-system emulator that embodies many of Priami's concepts. Designed originally as a control system for robotics written in a multitasking version of Forth, I've extended the project into a modular, extensible, open-systems design embodied in a multiprocessor network that emulates the major functions of the human nervous system. Included are interchangeable hardware/software components, a socketed software bus with plug-and-play capability, and self-diagnostics. The computer hardware is based on IEEE P996.1 bus cards; the operating system uses IEEE 1275-1994 standard software. The overall system features object-oriented design techniques and programming. I've also created a machine-independent high-level byte-coded script command language to manage it all.

Emulated neural-anatomical structures include cortex, brain stem, cerebellum, spinal cord, and autonomic and peripheral nervous systems, along with motor, sensory, auto-regulatory, and higher-cognitive AI behavior and synthetic emotions. Emulated body functions range from hormones and drugs acting on cell membranes to high-level responses.

As part of the IEEE 1275 standard, Forth helped me create a source-code library of individually compilable nervous-system components, per Priami. The library includes human childhood development milestones, epinephrine and oxytocin hormone functions, a pain mechanism and narcotic effects, the fear mechanism, and retrograde neuronal signaling via the endocannabinoid system. Recent enhancements include a form of autism based on a defective oxytocin receptor, the fibromyalgia syndrome (with simulated viral activity, immune-system responses, and antiviral antibiotic effects), and Bayesian probabilistic functions.

The system reflects intentional software and hardware flexibility. Using tiny six-pin eight-bit PIC10Fxx series microcontrollers, I've designed 35 different digital McCulloch-Pitts and analog Hebb artificial neurons. I also added eight-core 32-bit Parallax processors for coordinating brain stem sensorimotor, cerebellar, and low-level cortical activities. Moreover, the system can extend its original Forth-based, byte-coded AI scripting language via genetic algorithms to provide a form of machine learning and execution. It is also capable of examining its own internal variables, short- and long-term memory, knowledge base, and preferences profile to provide a limited form of self-awareness and personality expression.

I look forward to more such intelligent machines created through the kind of algorithmic systems biology explored by Priami.

Paul Frenger MD, Houston, TX

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

DOI:10.1145/1629175.1629179

In the Virtual Extension

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

Think Big for Reuse

Paul D. Witman and Terry Ryan

How can organizations successfully reuse not just objects and components, but rather very large-grained software elements, that is, entire systems and subsystems? This article examines an entire Internet banking system (applications and infrastructure) reused in business units all over the world. The authors explore the case of the BigFinancial Technology Center, and its parent company, which has created a number of software systems that have been reused in multiple businesses and in multiple countries. The article focuses on technology, process, and organizational elements of the development process, rather than on specific product features and functions.

Using the Thread-Fabric Perspective to Analyze Industry Dynamics

DongBack Seo and King-Tim Nak

To strive for competitive advantage, firms in the wireless industry are forming and dissolving partnerships and value chains at a rapid pace. More broadly, the nature of modern business competition appears to be undergoing a fundamental change. To explore the new industrial dynamics, the authors use the intuitive ideas of *threads*, *fabric,* and *weaving* to develop a framework that promises to facilitate the description and analysis of highly competitive and dynamic industries such as the wireless industry. The article aims to enhance our understanding of the changing value chain dynamics in modern industries by examining the wireless industry as a prototype.

Security Constructs for Regulatory Compliant Storage

Randal Burns and Zachary Peterson

Legislators and the courts have begun to recognize the importance of securing and maintaining electronic records. Sweeping pieces of electronic record management legislation, including Sarbanes-Oxley and HIPAA, now require storage systems to ensure the integrity and authenticity of financial records, protect consumer privacy, and guard against the unauthorized disclosure of a patient's medical information. Many storage vendors now provide "compliant" versions of their storage products, but often these platforms do not provide cryptographically strong evidence of compliance. The authors review three security constructs pursuant to meeting the requirements set forth by electronic records legislation.

The Future of Digital Imaging

Wonchang Hur and Dongsoo Kim

The authors envision digital imaging services in radiology, with emphasis on the recent advancements in digital imaging technology and its future direction. They focus on the four major issues prevailing in current imaging business practices: specialization, flexibility, reliability, and usability. In addition, they investigate the kinds of technologies pertaining to each issue, as well as the ways in which such technologies have enabled the invention of innovative services in diagnostic imaging practice.

Mobile Web 2.0 with Multidisplay Buttons

Seongwoon Kim, Inseong Lee, Kiho Lee, Seungki Jung, Joonah Park, Yuen Bae Kim, Sang Ryong Kim, and Jinwoo Kim

User-generated content (UGC) has become popular among Internet users for creating and sharing new media content. Mobile UGC services, with the technological advantages they possess and the convenience they offer in capturing new media content and adding tags, are likely to become the main driver of the UGC paradigm. Conventional mobile phone interfaces, however, do not support the exploratory browsing behavior typical of mobile UGC. In this study, the authors designed a new user interface specifically for exploratory browsing with a tag-based structure and a multidisplay button interface, and empirically investigated user perceptions of the new interface.

Designing Data Governance

Vijay Khatri and Carol V. Brown

As data is increasingly acknowledged as an organizational asset, organization leaders are realizing that data governance is critical for deriving business value. Building on an earlier IT governance model, the authors present a set of five data decision domains: data principles, data quality, metadata, data access, and data life cycle. They also discuss why they are important, and offer guidelines for what governance is needed for each decision domain. By operationalizing the locus of accountability of decision making (the "who") for each decision domain, the authors create a data governance matrix that can be used by practitioners to design a governance model for their data assets.

Domotic Technologies Incompatibility Becomes User Transparent

Vittorio Miori, Dario Russo, and Massimo Aliberti

The authors propose a solution to help overcome the obstacles currently hindering the spread of domotics (or, home automation). Their idea will enable consumers to freely choose home automation devices and systems based solely on considerations of cost, function, aesthetics, and brand preference, without any technical constraints. Moreover, users will be spared the frustrations of incompatible manufacturing products based on proprietary technologies. In short, the open standards proposed promise to bring about a spectacular surge in domotic services and applications.

Technical Opinion: Random Selection from a Stream of Events

Zvi Drezner

Consider a stream of events received during a limited period of time, such as applications on the Internet. One event needs to be randomly selected as a "winner." Due to physical limitations, the list of all events cannot be stored. Thus, the winner cannot be selected at the end of the period. It is not known how many events will materialize. Each event should be selected with the same probability. The author ponders a simple way for such a selection on the fly.

DOI:10.1145/1629175.1629180 Wendy Hall

ACM's Annual Report

It's been an exhilarating first year as President of ACM. As a longtime member of this Association, I've come to count on an amazing and far-reaching assembly of committed volunteers and staff to ensure the Association's goals are met and pledges kept. But to have a front-row seat to the whirlwind of activities and initiatives that make up a year in the life of ACM is a remarkable and, at times, daunting experience.

At the close of FY09, ACM stood as the largest educational and scientific computer society in the world. After seven consecutive years of steady growth, ACM ended the fiscal year with membership at an all-time high. It is a monumental testament to the value of an ACM membership that the Association continues to thrive in size, in scope, and in global reach at a time when the world struggles with widespread economic uncertainty.

> "It is a monumental testament to the value of an ACM membership that the Association continues to thrive in size, in scope, and in global reach at a time when the world struggles with widespread economic uncertainty."

Initiatives to broaden and share ACM's rich array of professional resources and services with a far greater global audience were a top priority this year, and the results of these efforts have been most rewarding. It was my honor to preside over the launch of ACM Europe in October, thus establishing a strong ACM presence throughout Europe and a base to support ACM's European members and activities. As you read this, we are preparing to launch ACM India in January to offer similar support and strengthen ACM's visibility in a country rich in high-tech opportunities. Both these initiatives were the result of exhaustive efforts of devoted ACM volunteers and executive staff working with industry and academic leaders in Europe and India to determine how best to serve current members and to attract new ones to the organization. Next up: China.

ACM is also committed to addressing the multifaceted issues related to the image of computing and the health of the discipline and profession. It is a challenge that has been embraced across the ACM spectrum—from its Education Board, Public Policy committee, ACM-W Council, and Computer Science Teachers Association, to NSF-funded initiatives involving ACM partnerships with other scientific organizations, and its many Special Interest Groups working to raise awareness and promote the possibilities offered by the computing field.

The following report gives you but a glimpse of some of the major highlights of ACM's FY09, none of which would have been possible without the influential efforts of so many dedicated and generous volunteers worldwide. The Association is also grateful for the ongoing endorsements from major corporations who value ACM's ability to recognize technical excellence by sponsoring or supporting a number of ACM's prestigious awards and student competitions.

As we look forward to another fruitful year, the challenges may appear great, but the opportunities are even more so. As always, I will count on the ACM to triumph.

Wendy Hall, ACM PRESIDENT

ACM's Annual Report for FY09

ACM, the Association for Computing Machinery, is an international scientific and educational organization dedicated to advancing the arts, sciences, and applications of information technology.

Publications

This year marked the 10th anniversary of ACM's Digital Library—the centerpiece of the ACM Publications portfolio. By year-end, the DL offered over 240,000 full-text articles and there were over 1.25 million citations covered by the *Guide to Computing Literature*. Indeed, some 22,000 articles were added to the DL this year alone, and more than 128,000 works were added to *Guide*.

ACM's pledge to continually upgrade the offerings and features available in the DL resulted in several significant enhancements in FY09. Working closely with the search applications company Endeca, ACM introduced a new DL platform this year that added more powerful search capability, allowing users to not only explore existing data but to discover information that goes beyond simple query results. This search technology employs a new class of database designed for exploring information, not just managing search transactions. Among the new search enhancements are guided navigation; discovered terms drawn from ACM's subject classifications and keywords; refinements by author, publication, conference, and other criteria; and the ability to view related material in ACM journals, magazines, SIGs, and conferences.

The ACM Publications Board initiated two programs this year to further its strategic goal of making ACM the preferred publisher in computing. The Board formulated a plan to relaunch ACM's *International Conference Proceedings Series* as a high-quality alternative to *Lecture Notes in Computer Science* and it has begun development of a set of statistical quality measures to assist in ongoing assessment of its expanding journals program.

ACM currently publishes 78 periodicals, including six journals, 31 *Transactions*, eight magazines, and 23 newsletters. Two new periodicals appeared this year: *Transactions on Computational Theory* and *Journal of Information and Data Quality*. In addition, the Publications Board approved proposals for two new publications: *ACM Transactions on Management Information Systems* and *ACM Transactions on Intelligent Systems and Technology*.

Education

ACM continues to work with multiple organizations on important issues related to the image of computing and the health of the discipline and profession. One of the Association's most ambitious undertakings this year was its partnership with the WGBH Educational Foundation (a Boston-based PBS station) on a project entitled "New Image of Computing." With funding from the National Science Foundation, the goal of this joint effort is to reshape the image of computing among high school students, with special efforts to reach Latina females and African-American males. The project will produce a wide-ranging national outreach and communications plan to spread the word about the rewards and benefits of a life in computing. A pilot project will also be launched to develop a comprehensive evaluation of the project findings, and create a plan for implementation on a national level.

ACM also worked with the Computing Research Association (CRA) and National Center for Women and Information Technology (NCWIT) to improve the profile of CS education efforts as part of the federal government's Networking and Information

ACM Council

PRESIDENT
Wendy Hall

VICE PRESIDENT
Alain Chesnais

SECRETARY/TREASURER
Barbara Ryder

PAST PRESIDENT
Stuart I. Feldman

SIG GOVERNING BOARD CHAIR
Alexander Wolf

PUBLICATIONS BOARD CO-CHAIRS
Ronald Boisvert, Holly Rushmeier

MEMBERS-AT-LARGE
Carlo Ghezzi, Anthony Joseph,
Mathai Joseph, Kelly Lyons,
Bruce Maggs, Mary Lou Soffa,
Fei-Yue Wang

SGB COUNCIL REPRESENTATIVES
Joseph A. Konstan, Robert A. Walker,
Jack Davidson

ACM Headquarters

EXECUTIVE DIRECTOR/CEO
John R. White

DEPUTY EXECUTIVE DIRECTOR/COO
Patricia M. Ryan

2008 ACM Award Recipients

A.M. TURING AWARD
Barbara Liskov

ACM-INFOSYS FOUNDATION AWARD IN THE COMPUTING SCIENCES
Jon Kleinberg

ACM/AAAI ALLEN NEWELL AWARD
Barbara Grosz and Joseph Y. Halpern

THE 2009–2010 ACM-W ATHENA LECTURER AWARD
Susan Eggers

GRACE MURRAY HOPPER AWARD
Dawson Engler

ACM-IEEE CS ECKERT-MAUCHLY AWARD
Joel Emer

KARL V. KARLSTROM OUTSTANDING EDUCATOR AWARD
John Hopcroft

OUTSTANDING CONTRIBUTION TO ACM AWARD
Wayne Graves and Bernard Rous

DISTINGUISHED SERVICE AWARD
Telle Whitney

PARIS KANELLAKIS THEORY AND PRACTICE AWARD
Corinna Cortes and Vladimir Vapnik

SOFTWARE SYSTEM AWARD
Gamma Parallel Database System,
David J. DeWitt, Robert Gerber,
Shahram Ghandeharizadeh,
Goetz Graefe, Michael L. Heytens,
Hui-I Hsiao, Murali Krishna,
Jeffrey F. Naughton, Donovan A. Schneider,
Anoop Sharma

SIAM-ACM AWARD IN COMPUTATIONAL SCIENCE AND ENGINEERING
Cleve Moler

ACM GORDON BELL PRIZES PEAK PERFORMANCE CATEGORY:
Multi-teraflops Simulations of Disorder Effects on the Transition Temperature of the High Tc Superconducting Cuprates.
From the Oak Ridge National Laboratory (ORNL):
Thomas C. Schulthess,
Gonzalo Alvarez, Michael S. Summers,
Don E. Maxwell, Markus Eisenbach,
Jeremy S. Meredith, Thomas A. Maier,
Paul Kent, Eduardo D'Azevedo,
From Cray, Inc.:
John M. Levesque, Jeffrey M. Larkin

SPECIAL CATEGORY IN ALGORITHM INNOVATION:
Linearly Scaling 3D Fragment Method for Large-Scale Electronic Structure Calculations:
From Berkeley Labs:
Lin-Wang Wang, Byounghak Lee,
Hongzhang Shan, Zhengji Zhao,
Juan Meza, Erich Strohmaier,
David Bailey

DOCTORAL DISSERTATION AWARD
Constantinos Daskalakis

HONORABLE MENTION
Derek Hoiem, Sachin Katti

Technology Research and Development (NITRD) program. NITRD spans several government agencies to coordinate investments in IT R&D. ACM, CRA, and NCWIT sent a joint letter to Congress earlier this year, making specific recommendations on how the NITRD Act of 2009 can be improved and how to expand and better utilize existing education efforts within the NITRD program.

The ACM Education Board finished a prolific year filled with projects and initiatives designed to reverse declining enrollments in computing disciplines and increase ACM's visibility within the worldwide educational community. With support from the National Science Foundation, the Education Board brought together several U.S.-based professional societies and organizations concerned with the current challenges in computing education. The goal of "The Future of Computing Education Summit" was to come to a shared vision of the problems facing those in computing education and how those problems might be addressed.

ACM's Computer Science Teachers Association (CSTA) continues to support and promote the teaching of computer science at the K–12 level as well as provides opportunities and resources for teachers and students to improve their understanding of computing disciplines. CSTA recently published *A Model Curriculum for K–12 Computer Science* to prepare young people to excel in computer science.

SIGITE completed a revised draft of the four-year IT model curriculum with the guidance and support of ACM's Education Council. The model is now being used as the basis for a two-year model curriculum.

Professional Development

A new *Queue* Web site (http://queue.acm.org/) was launched this year reflecting a significant effort by both the *Queue* Board and the ACM Professions Board to design an appealing and effective space for ACM practitioners. The site hosts editorial content from *Queue* as well as case studies and CTO Roundtable discussions from the Professions Board. The Board created this site as an important virtual community for high-powered practitioners.

ACM recently launched a new online course program (http://pd.acm.org/) through Element K that includes more than 2,500 online courses on a wide range of computing and business topics in multiple languages, 1,000 unique vLab exercises, an e-Reference Library, as well as a downloadable player that allows members to access assessments and self-study courses offline. The ACM Online Course Program is open to ACM professional and student members.

The Distinguished Speakers Program (DSP), ACM's primary outreach effort for student and professional chapters, continues to add new speakers to its roster. At year-end, 74 speakers from academia and industry were part of the program, speaking on a variety of topics from artificial intelligence and computer graphics, to emerging technologies and mobile computing. The speaker roster doubled in size last year and continues to flourish. Of the 40 speaking engagements that took place this year, 12 were hosted by international chapters.

Thousands of job seekers visited the Job Fair at SIGGRAPH Asia 2008, where 20 studios from around the globe, including Pixar, Lucasfilms, Animal Logic, and more, were recruiting talent.

Public Policy

The U.S. Public Policy Committee of ACM (USACM) made significant changes to its structure and its approach to developing policy positions. By the end of the fiscal year, USACM had six established subcommittees—voting; security and privacy; computing and the law; intellectual property; accessibility; and digital government—to provide specialized focus on particular issues to government leaders and policymakers. The committee works to educate legislators and the public about issues that will foster innovations in computing in ways that benefit society. Indeed, USACM members testified numerous times before congressional committees and helped develop principles on increasing the usability of government information online.

The ACM Education Policy Committee (ACM EPC), established to educate legislators about the role of computer science in K–12 education, made significant progress engaging policymakers and ensuring computer science at the K–12 level is explicitly considered in STEM (Science, Technology, Engineering, and Mathematics) Education Coalition discussions. Members of EPC and ACM staff also held several meetings with Congressional and Administration representatives to emphasize the critical role of CS education; introduced a bipartisan resolution to designate a National Computer Science Education Week; and convinced a group of governors and business interests (Achieve.org) to include Advanced Placement CS as a mathematics credit in its national framework.

Students

ACM's renowned International Collegiate Programming Contest, sponsored by IBM, drew 7,109 student teams representing 1,838 universities from 88 countries this year. The World Finals included 100 teams from around the world and was hosted by KTH, the Royal Institute of Technology in Stockholm. The 2009 ICPC World Finals Ceremony took place in the prestigious Stockholm Concert Hall, where Nobel Prizes are presented annually; students from St. Petersburg State University of IT, Mechanics and Optics took top honors.

The ACM Student Research Competition (SRC), sponsored by Microsoft Research, continues to provide a unique forum for undergraduate and graduate students to present their original research at well-known ACM-sponsored and co-sponsored conferences before a panel of judges and attendees. A select group of ACM conferences hosts two rounds of competition with winners from these meets advancing to the Grand Finals, where they are evaluated by a different panel of judges via the Web. Winners are invited to the annual ACM Awards Banquet where they receive formal recognition for their work.

ACM has developed partnerships with several leading technology companies, including Microsoft, Sun Microsystems, and Computer Associates, to offer valuable tools specifically for ACM student members. At no additional cost, student members can now access free software and courseware, offering a unique opportunity to access top resources, while also becoming part of the larger computing community.

SIGCOMM added its support to the scholarship program initiated by ACM-W Council offering financial aid to undergraduate and graduate

Balance Sheet: June 30, 2009 (in Thousands)

ASSETS	
Cash and cash equivalents	$22,502
Investments	47,696
Accounts receivable and other current assets	4,065
Deferred conference expenses and other assets	6,656
Fixed assets, net of accumulated depreciation and amortization	1,123
Total Assets	**$82,042**
LIABILITIES AND NET ASSETS	
Liabilities:	
Accounts payable, accrued expenses, and other liabilities	$7,461
Unearned conference, membership, and subscription revenue	21,354
Total liabilities	**$28,815**
Net assets:	
Unrestricted	47,412
Temporarily restricted	5,815
Total net assets	**53,227**
Total liabilities and net assets	**$82,042**
Optional contributions fund – program expense	($000)
Education board accreditation	$80
USACM Committee	20
Total expenses	**$100**

women students in computer science programs who are interested in attending research conferences. SIGCOMM will cover full costs of travel, lodging, and registration for any recipient of an ACM-W scholarship who chooses to attend a SIGCOMM-sponsored or in-cooperation conference or workshop.

Local Activities

There were 58 new chapters charted in FY09. Of the 11 new professional chapters, eight were internationally based; of the 47 new student chapters, 20 were international.

The Association introduced a new Special Interest Group in '09 to focus on the acquisition, management, and processing of spatially related information. SIGSPATIAL (http://www.sigspatial.org/) provides a forum for researchers, engineers, and practitioners designed to encourage research in handling spatial information, participation in standardization activities including terminology, evaluation, and methodology, and interdisciplinary education.

International

ACM's international initiatives resulted in the establishment of new Councils in India and Europe that promise to strengthen the Association's ties with these global technology hubs and better understand the key issues and initiatives within their academic, research, and professional computing communities.

ACM Europe is lead by the ACM Europe Council, comprised of 15 distinguished European computer scientists from both academia and industry who have pledged to help build an ACM presence that would focus on bringing high-quality technical activities, conferences, and services to ACM members and computing professionals throughout the continent.

In India, a similar effort to enhance ACM's efforts in the region came to fruition with the creation of **ACM India,** a non-profit learned society led by the ACM India Council comprised of 19 of the country's industry and academic leaders who plan to foster technical activities and better serve members, conferences, and chapters in the region. ACM India hopes to influence public discourse and political decision-making and to draw more of its burgeoning IT population into the ACM fold.

ACM's Education Board extended its international activities to include planning the Informatics Education Europe Conference in Venice; keeping a close eye on accreditation developments within Europe; and monitoring activities leading to the signing of the Seoul Accord, which helped develop criteria for the mutual recognition internationally of accreditation activity.

Electronic Community

ACM launched a new Web site for its flagship publication *Communications of the ACM* (http://cacm.acm.org/). The site features a wide range of high-quality and topical News, Opinion, Research, and Practitioner-oriented content from the magazine, as well as original and user-generated content exclusive to the site. Among the site's numerous features is the ability to access the complete *Communications*' archive spanning more than 50 years of in-depth coverage of the computing profession, as well as the ability to search content from across the entire ACM Digital Library and other sources around the Web. In addition, the site contains extensive blog content that presents a completely new forum for a growing community of the world's leading industry and academic experts on a range of topics within computing. The site is updated daily and is acces-

Statement of Activities: Year ended June 30, 2009 (in Thousands)

		Temporarily	
REVENUE	**Unrestricted**	**Restricted**	**Total**
Membership dues	$9,178		$9,178
Publications	15,873		15,873
Conferences and other meetings	23,253		23,253
Interests and dividends	1,844		1,844
Net (depreciation) of investments	(5.613)		(5.613)
Contributions and grants	3,120	$874	3,994
Other revenue	341		341
Net assets released from restrictions	678	(678)	0
Total Assets	**48,674**	**196**	**48,870**
EXPENSES			
Program:			
Membership processing and services	$958		$958
Publications	11,327		11,327
Conferences and other meetings	21,783		21,783
Program support and other	8,299		8,299
Total	**42,367**		**42,367**
Supporting services:			
General administration	8,764		8,764
Marketing	1,474		1,474
Total expenses	**52,605**		**52,605**
Increase (decrease) in net assets	(3,931)	196	(3,735)
Net assets at the beginning of the year	51,343	5,619	56,962
Net assets at the end of the year	**$47,412***	**$5,815**	**$53,227***

* Includes SIG Fund balance of $28,867K

sible by both the general public and *Communications* subscribers.

The Association joined the social media movement with the creation of an ACM page on Facebook (http://www.facebook.com/pages/ACM-Association-for-Computing-Machinery/17927643151?v=wall&viewas=1755757376)."Fans" are able to keep up with the latest ACM developments with Facebook's popular sections: Wall; Info; Events; Photos; Boxes; Discussions.

Two sites on ACM's Web site were created expressly for new Professional and Student members. Both sites are divided into four sections and each section describes in detail all the information needed to get started as a new member of ACM. The site will continue to evolve as more benefits or newsworthy items arise.

SIGGRAPH's social networking site—Digital Arts Community (arts.siggraph.org)—now features the work of over 800 artists. The site passed the 300-member mark in June and has proven a vital site for artists to converse with fellow artist members.

SIGSIM created a Modeling and Simulation Knowledge Repository to provide valuable content to its members, including hyperlinks to 17 different areas of modeling and simulation.

KDD-09 featured a novel conference social networking and scheduling platform that provided conference attendees with many useful abilities, including managing conference schedules, commenting on papers, and conversing with fellow attendees.

Conferences

SIGMOD, SIGKDD, SIGIR, and SIGWeb co-sponsored the first ACM International Conference on Web Search and Data Mining (WSDM)—a hugely successful event that spotlighted the interdisciplinary nature of Web search and data mining.

SIGGRAPH 2008 attracted almost 28,500 artists, researchers, gaming experts, filmmakers, and developers representing 87 countries to its annual conference. The Los Angeles-based meeting also drew 230 international companies to its exhibition hall, an increase over the previous year.

Both Supercomputing (SC08) and Multimedia 2008 conferences posted a record number of attendees. The Vancouver-based Multimedia meeting drew sponsors from a variety of companies and organizations, including Google, Yahoo!, Microsoft, FXPal, RICOH, Telefonica, and Nokia. And SC08, marking its 20th anniversary, set an all-time exhibitors record with 337 companies and organizations filling out every exhibit hall in Austin Convention Center.

Recognition

The ACM Fellows Program, established in 1993 to honor outstanding ACM members for their achievements in computer science and IT, inducted 44 new fellows in FY09, bringing the total count to 675.

ACM also named 37 Distinguished Members in recognition of their individual contributions to both the practical and theoretical aspects of computing andinformationtechnology.Inaddition, 605 Senior Members were recognized for their demonstrated performance that sets them apart from their peers.

The ACM-IEEE CS Ken Kennedy Award, established in FY09 to recognize contributions to programmability and productivity in computing as well as community service or mentoring contributions., honors the late Ken Kennedy, the founder of Rice University's CS program and one of the foremost experts on high-performance computing.

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

Dear Colleague,

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

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

I would also like to take this opportunity to mention ACM-W, the membership group within ACM. ACM-W's purpose is to elevate the issue of gender diversity within the association and the broader computing community. You can join the ACM-W email distribution list at http://women.acm.org/joinlist.

ACM MEMBER BENEFITS:

- A subscription to ACM's newly redesigned monthly magazine, ***Communications of the ACM***
- Access to ACM's **Career & Job Center** offering a host of exclusive career-enhancing benefits
- **Free e-mentoring services** provided by MentorNet®
- **Full access to over 2,500 online courses** in multiple languages, and 1,000 virtual labs
- **Full access to 600 online books** from Safari® Books Online, featuring leading publishers, including O'Reilly (Professional Members only)
- **Full access to 500 online books** from Books24x7®
- Full access to the new ***acmqueue*** website featuring blogs, online discussions and debates, plus multimedia content
- The option to subscribe to the complete **ACM Digital Library**
- The **Guide to Computing Literature**, with over one million searchable bibliographic citations
- The option to connect with the **best thinkers in computing** by joining **34 Special Interest Groups** or **hundreds of local chapters**
- **ACM's 40+ journals and magazines** at special member-only rates
- ***TechNews***, ACM's tri-weekly email digest delivering stories on the latest IT news
- ***CareerNews***, ACM's bi-monthly email digest providing career-related topics
- ***MemberNet***, ACM's e-newsletter, covering ACM people and activities
- **Email forwarding service & filtering service**, providing members with a free acm.org email address and **Postini** spam filtering
- And much, much more

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

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

Sincerely,

Wendy Hall

Wendy Hall

President
Association for Computing Machinery

Association for
Computing Machinery

Advancing Computing as a Science & Profession

Association for
Computing Machinery

Advancing Computing as a Science & Profession

membership application & *digital library* order form

Priority Code: ACACM10

You can join ACM in several easy ways:

Online
http://www.acm.org/join

Phone
+1-800-342-6626 (US & Canada)
+1-212-626-0500 (Global)

Fax
+1-212-944-1318

Or, complete this application and return with payment via postal mail

Special rates for residents of developing countries:
http://www.acm.org/membership/L2-3/

Special rates for members of sister societies:
http://www.acm.org/membership/dues.html

Please print clearly

Name

Address

City State/Province Postal code/Zip

Country E-mail address

Area code & Daytime phone Fax Member number, if applicable

Purposes of ACM

ACM is dedicated to:
1) advancing the art, science, engineering, and application of information technology
2) fostering the open interchange of information to serve both professionals and the public
3) promoting the highest professional and ethics standards

I agree with the Purposes of ACM:

Signature

ACM Code of Ethics:
http://www.acm.org/serving/ethics.html

choose one membership option:

PROFESSIONAL MEMBERSHIP:

□ ACM Professional Membership: $99 USD

□ ACM Professional Membership plus the ACM Digital Library: $198 USD ($99 dues + $99 DL)

□ ACM Digital Library: $99 USD (must be an ACM member)

STUDENT MEMBERSHIP:

□ ACM Student Membership: $19 USD

□ ACM Student Membership plus the ACM Digital Library: $42 USD

□ ACM Student Membership PLUS Print *CACM* Magazine: $42 USD

□ ACM Student Membership w/Digital Library PLUS Print *CACM* Magazine: $62 USD

All new ACM members will receive an ACM membership card.
For more information, please visit us at www.acm.org

Professional membership dues include $40 toward a subscription to *Communications of the ACM*. Member dues, subscriptions, and optional contributions are tax-deductible under certain circumstances. Please consult with your tax advisor.

RETURN COMPLETED APPLICATION TO:

Association for Computing Machinery, Inc.
General Post Office
P.O. Box 30777
New York, NY 10087-0777

Questions? E-mail us at acmhelp@acm.org
Or call +1-800-342-6626 to speak to a live representative

Satisfaction Guaranteed!

payment:

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc. in US dollars or foreign currency at current exchange rate.

□ Visa/MasterCard □ American Express □ Check/money order

□ Professional Member Dues ($99 or $198) $ ____________________

□ ACM Digital Library ($99) $ ____________________

□ Student Member Dues ($19, $42, or $62) $ ____________________

Total Amount Due $ ____________________

Card # Expiration date

Signature

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:10.1145/1629175.1629181 http://cacm.acm.org/blogs/blog-cacm

Software Engineering, Smartphones and Health Systems, and Security Warnings

Greg Linden writes about frequent software deployments, Ruben Ortega reports on smartphones and health systems research, and Jason Hong discusses designing effective security warnings.

From Greg Linden's "Frequent Releases Change Software Engineering"

Software release cycles are usually long, measured in months, sometimes in years. Each of the stages—requirements, design, development, and testing—takes time.

Recently, some of the constraints on software deployment have changed. In Web software, deployment is to your own servers, nearly immediate and highly reliable. On the desktop, many of our applications routinely check for updates on each use and patch themselves. It no longer is the case that getting out new software to people is slow and inconsistent. The likelihood of a reliable, fast Internet connection on most machines has made it possible to deploy software frequently.

But just because a thing is possible does not mean it is desirable. Why would we want to deploy software more frequently? Is it not better to be careful, slow, and deliberate about change?

The main reason to consider frequent deployments is not the direct impact of getting software out to customers more quickly, but the indirect impact internally. Frequent releases force changes in how an organization develops software. These changes ultimately reduce risk, speed development, and improve the product.

For example, consider what is required to deploy software multiple times per day. First, you need to build new deployment tools that are capable of rapidly pushing out new software, can handle thousands of potential versions and enforce consistency, and allow rapid rollbacks in case of problems.

Software development has to change. With multiple near-simultaneous rollouts, no guarantee of synchronous deployment, and no coordination possible with other changes, all software changes have to be independent and backward-compatible. The software must always evolve.

Requirements, design, and testing can be shortened and replaced with online experimentation. To learn more about customer requirements and design preferences, deploy to a small set of customers, test against a larger control group, and get real data on what people want. Bugs are expected and managed as a risk through small deployments, partial deployments, and rapid rollbacks.

Compare this to a more traditional development process. Requirements gathering and design are based on small user studies and little data. Software is developed without concern about backward compatibility and must be rolled out synchronously with many other changes. Testing has the goal of eliminating bugs—not merely managing risk—and is lengthy and expensive. When the software does roll out, we inevitably find errors in requirements, design, and testing, but the organization has no inherent capacity to respond by rapidly rolling back the problems or rolling out fixes.

Frequent releases are desirable because of the changes it forces in software engineering. It discourages risky, expensive, large projects. It encourages experimentation, innovation, and rapid iteration. It reduces the cost of failure while also minimizing the risk of failure. It is a better way to build software.

The constraints on software deployment have changed. Our old assumptions on the cost, consistency, and speed of software deployments no longer hold. It is time to rethink how we do software engineering.

From Ruben Ortega's "Smartphones and Health Systems Research at Intel Seattle"

Intel Labs in Seattle, WA, hosted an open house on September 28, 2009 to showcase its research projects (http://seattle.intel-research.net/projects.php). Intel's health systems research encompasses myriad projects that are focused on long-term health monitoring and care systems. Since most people dislike carrying an extra health-dedicated device, the research has focused on adding sensors to the technology people carry with them everywhere—smartphones. The two specific areas with the most potential for near-term change are: (a) using sensors already present in smartphones (accelerometers and GPS) to monitor the movements and mobility of the wearer and (b) building applications that encourage ad hoc team-building and tracking for people to help accomplish their health goals.

Sensor technologies on cell phones can be adapted to help do long-term tracking of family and loved ones. Accelerometers could be used to identify different kinds of motion and measurement of "gait" in people's movement. By tracking and measuring the "gait" of someone's walking over time, the technology could help identify when someone is moving normally or if something has changed and an individual's walking gait is impaired. The information that is captured on the device could either be stored and analyzed locally, or uploaded to caregivers and health-care providers. Given the ubiquity of cell phones, the extra cost of adding sensors and inputs would be minimized as the large volume production costs should drive the price down.

A nearer-term application for smartphones would be to use their abilities to connect people via data-sharing technologies to form social health-support groups. You could easily imagine using an application to create teams of individuals who are working to improve their own health. The first best uses would be to create teams that encourage weight loss through the creation of ad hoc competitions modeling TV shows like *The Biggest Loser*. Using peer-pressure, peer-support, and real-time feedback, individuals could track how their peers are doing in improving their weight management over time. Other potential applications would be creating a tool so that compliance is tracked among groups of people in taking medication or monitoring their insulin level, or providing a pregnancy application to contact other people, like themselves, who are working through the trials of a pregnancy to ask, "Is this normal?"

The research being done at the Intel lab is still in the formative stages. However, I am eager to see this technology made into a product and launched so that it moves from "good idea" to useful to its intended customers.

From Jason Hong's "Designing Effective Warnings"

In my last post, "Designing Effective Interfaces for Usable Privacy and Security," I gave an overview of some of the issues in designing usable interfaces for security. Here, I will look at more of the nuts and bolts of designing and evaluating effective user interfaces.

Now, entire Web sites, courses, and books are devoted to how to design, prototype, and evaluate user interfaces. The core ideas—including observing and understanding your users' needs, rapid prototyping, iterative design, fostering a clear mental model of how the system works, and getting feedback from users, through both formal and informal user studies—all still apply.

However, there are also several challenges that are unique to designing interfaces dealing with security and privacy. Let's look at one common design issue with security, namely security warnings.

Computer security warnings are something we see every day. Sometimes these warnings require active participation from the user; for example, dialog boxes that ask the user if they want to store a password. Other times they are passive notifications that require no specific action by the user; for example, letting users know that the Web browser is using a secure connection.

Now, if you are like most people I've observed, you are either hopelessly confused by these warnings and just take your best guess or you pretty much ignore most of these warnings. And sometimes (perhaps too often) both of these situations apply.

At least three different design issues are in play here. The first is whether the warning is active or passive. *Active warnings* interrupt a person's primary task, forcing them to take some kind of action before continuing. In contrast, *passive warnings* provide a notification that something has happened, but do not require any special actions from a user. So far, research has suggested that passive warnings are not effective for alerting people to potentially serious consequences, such as phishing attacks. However, bombarding people with active warnings is not a viable solution, since people will quickly become annoyed with being interrupted all of the time.

The second design issue is habituation. If people repeatedly see a warning, they will become used to it, and the warning will lose its power. Worse, people will expect the warning and simply swat it away even if that was not their intended action. I know I've accidentally deleted files after confirming the action, only to realize a few seconds later that I had made a mistake.

A related problem is that these warnings have an emergent effect. People have been trained over time to hit "OK" on most warnings just so that they can continue. In other words, while people might not be habituated to your warnings specifically, they have slowly become habituated to warnings.

The third design issue here is defaults. In many cases, you, as the system designer, will know more about what users *should* be doing, what the safer action is. As such, warning interfaces need to guide users toward making better decisions. One strategy is providing good defaults that make the likely case easy (e.g., no, you probably don't want to go to that phishing site) while making it possible, but not necessarily easy, to override. C

Greg Linden is the founder of Geeky Ventures. **Ruben Ortega** is a technologist and startup enthusiast. **Jason Hong** is an assistant professor at Carnegie Mellon University.

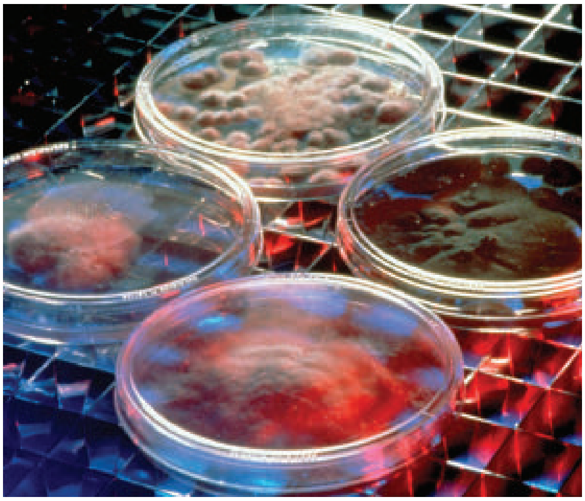
DOI:10.1145/1629175.1629182

David Roman

The Corollary of Empowerment

The unfilteredness of the Internet, while largely considered a plus, is taking some knocks. Abundant, easily accessible data sits side by side with "rumors, lies, and errors," and the victim is science, according to Michael Specter. "Anyone can seem impressive with a good Web site and some decent graphics," he writes in *Denialism: How Irrational Thinking Hinders Scientific Progress, Harms the Planet, and Threatens Our Lives*, (Penguin Press, 2009) (http://www.amazon.com/Denialism-Irrational-Thinking-Scientific-Threatens/dp/1594202303).

The Internet contributes to a "dysfunctional relationship with science" because its structure and evolution have created a place "where misinformation is likely to thrive and good information has a harder and harder time competing," says Chris Mooney, co-author of *Unscientific America: How Scientific Illiteracy Threatens Our Future* (Basic Books, 2009) (http://www.amazon.com/Unscientific-America-Scientific-lliteracy-Threatens/dp/0465013058), in an exchange with Specter on Slate (http://www.slate.com/id/2234719/entry/2234720/).



Science is difficult, and "too many scientists don't know how to explain it," Mooney writes. That gives quackery some footing. "For every accurate science blogger, there is an extremely popular anti-science blogger or Web site....As a consequence, real science is constantly abused, and the most credible experts can barely keep up with all the nonsense, much less refute it," Mooney says.

Social networks can compound the abuse by spreading information deemed "interesting" more quickly than information that is not so interesting, according to researchers at IBM and Carlos III University of Madrid (http://cacm.acm.org/news/50689). Indeed, Spanish researchers say their data "corroborates the predominant role of heterogeneity in social networks where the spread of information is concerned."

Misinformation is a corollary to Internet empowerment.

A silver lining may be the example of Wikipedia that questions the assumption that truth will prevail online. The online encyclopedia dropped its trademark egalitarianism and gave control of some of its content to editors (http://meta.wikimedia.org/wiki/Wikipedia_needs_editors). "Some more security, some more procedure can make things more organized," the site says, "[even] if it sounds anti-wiki."

ACM Member News

VISIONS OF COMPUTER SCIENCE CONFERENCE

ACM and the British Computer Society will jointly host the ACM-BCS 2010 Visions of Computer Science conference to be held at the Informatics Forum, Edinburgh University, Scotland, Apr. 13–16, 2010. This flagship event aims to energize the computing community by presenting some positive and inspiring visions of computer science.

The keynote speakers are Ross Anderson, University of Cambridge; Nicolo Cesa Bianchi, University of Milan; Jon Kleinberg, Cornell University; and Barbara Liskov, Massachusetts Institute of Technology.

The conference will cover a wide array of computing topics and issues, including computer architectures and digital systems; theoretical computer science; algorithms and complexity; logic and semantics; non-standard models of computation; quantitative evaluation of algorithms, systems, and networks; eScience; and bioinformatics and medical applications.

For more information, visit http://www.bcs.org/server.php?show=nav.11980.

DIGITAL LIBRARY CONTENT PRESERVATION

ACM is providing its institutional library customers with advanced electronic archiving services to preserve their electronic resources. These services, provided by Portico and CLOCKSS, address the scholarly community's critical need for long-term solutions that assure reliable, secure, and deliverable access to their burgeoning digital collection of scholarly works. This initiative is part of ACM's ongoing investment in content, features, performance, and the worldwide reach of its Digital Library, with the aim of making it easier for libraries to accelerate their transition away from print.

PHOTOGRAPH BY PACIFIC NORTHWEST NATIONAL LABORATORY

Society | DOI:10.1145/1629175.1629185

Tom Geller

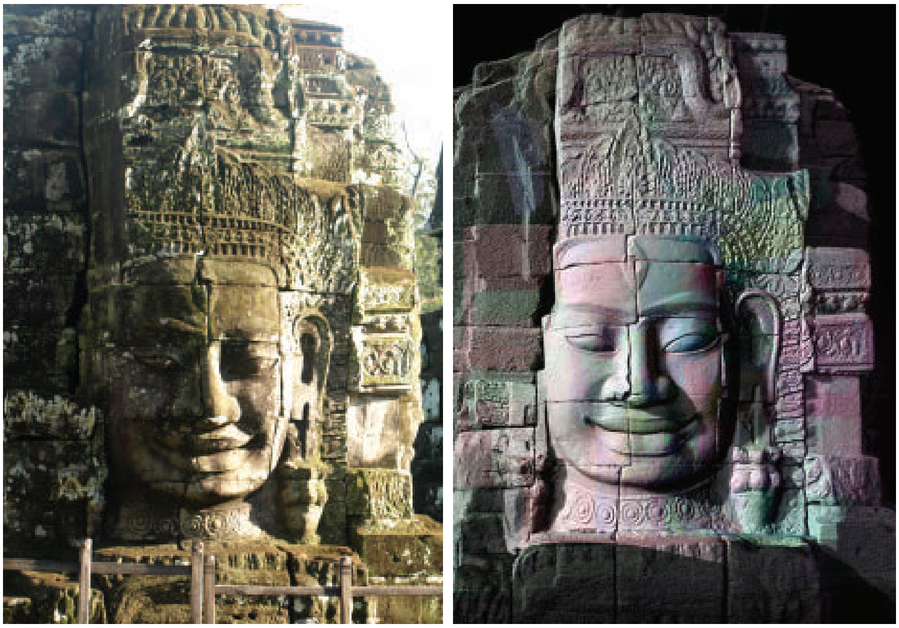
Rebuilding for Eternity

Researchers use computer vision techniques to preserve culturally significant sites as high-resolution 3D models.

BUILDINGS COLLAPSE. WIND and rain beat them, temperatures cycle from freezing to blistering, and random strikes of lightning threaten sudden obliteration. Those in wet climes face water rot; in the desert, ceaseless wear by dust and sand. Even more potent are the human challenges: war, fire, and deliberate destruction. No earthly structure is safe, from the Ancient Library of Alexandria to the Twin Towers of New York City.

But digital representations can survive such dangers, capturing structures forevermore. Digitization provides other benefits, such as the ability to "visit" a structure remotely, to examine its otherwise inaccessible details, or to observe how it's changed over time. Further, digitized structures invite researchers to apply computer-based analytic tools to draw out new discoveries in such fields as archaeology, history, and architecture.

Improvements in digital storage, network access, and processing power of the past 10 years have encouraged researchers to capture ever-larger sites. Now, two distinct methods enable high-resolution, 3D digitization of structures as large as a city street or a multi-acre historical site. One method uses high-end laser scanning equipment for accurate digitization to the sub-millimeter level; the other uses video or photo collections as the basis for analysis and reconstruction.



A tourist's photo, left, of the face of a statue at Bayon temple posted on the Flickr Web site. At right is an image of the same face in the library of the Bayon Digital Archive Project, led by Katsushi Ikeuchi of the University of Tokyo.

PHOTOGRAPH BY FLICKR USER DEEPWARREN

Mining Tourists' Photos

By now, many computer-savvy people have encountered site reconstruction in the form of Microsoft's Bing Maps, Google Earth, or Google Maps' Street View feature. All three are the result of mapping programs that were enhanced with real-world imagery captured through satellite photography, aerial photography, or at street level.

These efforts, while impressive, produce imagery that shows sites from only one or two aspects. Such views are sufficient for most purposes—to help people find, recognize, and "tour" remote locations—even while they're inadequate for 3D reconstructions on their own. Google Earth offers an additional way to add handcrafted 3D models to its terrains through the Google SketchUp program—a feature appropriate for both existing and historical buildings. One unusually extensive SketchUp initiative placed recreations of more than 6,000 ancient Roman buildings in their original positions in Rome.

Another consumer-level phenomenon that contributes to high-end 3D modeling is crowdsourced photos, available through such sources as the photo-sharing Web site Flickr. That was the basis for a reconstruction of present-day Rome, led by Sameer Agarwal, an acting assistant professor of computer science and engineering at the University of Washington, and described in the paper "Building Rome in a Day." Agarwal and colleagues selected 150,000 photos from more than two-and-a-half million returned from a Flickr.com search for "Rome" and "Roma," and applied existing Structure from Motion techniques to meld multiple views of the same structure into a unified 3D model. Central to this project—and to most others that agglomerate photos—is a feature detector that recognizes image elements. This project uses the popular Scale-Invariant Feature Transform algorithm; others include Speeded Up Robust Features and Maximally Stable Extremal Regions.

The Bayan temple project measured the entire site to a resolution of at least one centimeter.

The Rome project was unusual in its scale, and the University of Washington team used a variety of existing and novel approaches to match photos, place them in relation to each other, and ultimately meld them back together into a consistent geometry. The latter steps led to two innovations. First, the team computationally reduced the number of photographs to a "skeletal set" to simplify and confirm site geometry; second, newly developed software optimized the results over a distributed parallel network of about 500 cores. The result was a completion time, from photo matching to reconstruction, of less than 22 hours. Previous techniques would have taken more than 11 days to complete just the photo-matching process.

Much of the work on this project, and the Photo Tourism project by three of the same researchers, led to production of the consumer-level program Microsoft Photosynth and an open-source version, Bundler. According to Microsoft Partner Architect Blaise Aguera y Arcas, approximately 16% of recent, user-contributed Photosynth mosaics are of artwork or heritage-related sites.

The photographed environment for the Rome project was fairly static. Many of the sites were of long-standing structures, and nearly all photos were taken in the five years since the launch of Flickr. But 3D modeling techniques can also enhance historic photographs, as the paper "Inferring Temporal Order of Images from 3D Structure," by Grant Schindler, a Ph.D. candidate at the Georgia Institute of Technology, and colleagues, demonstrated with the use of a collection of 212 photos of downtown Atlanta taken over a period of 144 years. 3D analysis of the photos determined structure locations and then, based on the pattern of buildings appearing and disappearing, a constraint-satisfaction algorithm put them in the correct order. (An interactive applet at http://4d-cities.cc.gatech.edu/atlanta/ enables people to "time travel" through the images.)

Laser Scanning and Photometry

Photo-based modeling's big advantage is that many people can create its source material. But it has its failings, most notably when trying to coordinate photo collections that depict large blank spaces. Depth information is mostly interpolated from multiple views of the

Milestones

Feng Kang Prize and Other CS Awards

Tai Xue-Cheng, Manindra Agrawal, Raj Jain, and Anurag Kumar were recently honored for their contributions to computer science.

FENG KANG PRIZE

Tai Xue-Cheng, an associate professor at Nanyang Technological University's School of Physical and Mathematical Sciences, was awarded the eighth Feng Kang Prize in Scientific Computing. Xue-Cheng's research involves numerical analysis and computational mathematics, in particular image processing. His mathematical modeling has been used to restore degraded images to their original look. Xue-Cheng has also developed new models for MRI medical image processing and other medical and industrial applications.

G.D. BIRLA AWARD

Manindra Agrawal, a professor at the Indian Institute of Technology at Kanpur's department of computer science and engineering, was awarded the 2009 G.D. Birla Award for Scientific Research for his pioneering research in the theories of computation and algorithms. The award honors high-caliber scientific accomplishments of Indian scientists, preferably below the age of 50. A co-winner of the 2006 Gödel Prize, Agrawal's accomplishments include the first deterministic algorithm to test an *n*-digit number for primality in a time that has been proven to be polynomial in *n*.

CDAC-ACCS FOUNDATION AWARD

The Center for Development of Advanced Computing (CDAC) and the Advanced Computing and Communications Society (ACCS) jointly presented the 2009 CDAC-ACCS Foundation Award to Raj Jain, professor of computer and engineering at Washington University in St. Louis's School of Engineering and Applied Science, for his role in influencing the growth and impact of networking technology and to Anurag Kumar, professor and chairman of the department of electrical communication at the Indian Institute of Science at Bangalore, for his contributions to the analysis, optimization, and control techniques in communication networks.

same object. If those views don't exist, the image remains flat. Also, quality, while improved with more cameras, tends to be uneven. And photo-based modeling is only effective in places that are accessible to a camera.

Laser scanning combined with photometry comprises a much more reliable solution for applications that require it. That's the combination Katsushi Ikeuchi, a professor in the Institute of Industrial Science at the University of Tokyo, and colleagues used in capturing the Bayon temple, a complex of sacred structures in Angkor Thom, Cambodia that covers more than five acres. Ikeuchi's team used laser-measurement devices mounted on scaffolding, ground-level tripods, and a cherry picker to determine surface depth in places where those conventional approaches could reach. To scan down narrow corridors it added ladder-mounted climbing scanners, and for points high on the 40-meter-tall temple it scanned from a tethered balloon, in both cases developing software to compensate for the scanners' sometimes-unpredictable motion.

The Bayon Digital Archive Project gathered more than 10,000 range images totaling more than 250 gigabytes of data, measuring the entire site to a resolution of at least one centimeter. Because the data set was so big, several new algorithms were needed to match and align the points into a 3D mesh.

Instead of the "next-neighbor" alignment algorithm previously used, the Bayon temple team developed a two-step process that quickly identified matching pairs at the time of data capture, thereby converting the ultimate calculation from N^2 to N complexity. Later, the points were aligned simultaneously on a parallel processor cluster in a week of processor time—a 14-fold improvement over what the team claims was needed under existing algorithms.

The final model was important in two regards. First, it captured details of the 800-year-old temple, which is in danger of collapse. Second, it made comprehensive computer-aided scrutiny of the site possible—a benefit that bore fruit when the team was able to definitively categorize the temple's 173 carved stone faces in a new and significant way.

Similar results came from laser-scanned models at another historical site, the mausoleum of Henry VII, a 14th-century King of Germany. The monument comprises many parts that have been moved, lost, changed, and amended over the intervening 700 years, and the project's goals were both reconstructive and educational. As Clara Baracchini, officer at Superintendency for Environmental, Architectural, Artistic, and Historical Heritage of the Provinces of Pisa, Livorno, Lucca, and Massa Carrara, and colleagues noted, the project could be used "to teach medieval sculpture to students and to let them try to reconstruct the original monument from the disassembled components."

But while laser scanning can create models of unsurpassed detail, the main problem with it is that, as Georgia Tech Associate Professor Frank Dellaert put it, "You can't do it in the past." On the other hand, photo-based approaches lack some of laser scanning's advantages. The two approaches differ greatly in approach, cost, and purpose, but both have already proven themselves invaluable for historians and researchers. C

Further Reading

Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L.
Image-based façade modeling, *ACM Transactions on Graphics 27*, 5, December 2008.

Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.
Building Rome in a day. International Conference on Computer Vision, Kyoto, Japan, 2009.

Schindler, G., Dellaert, F., Kang, S.B.
Inferring temporal order of images from 3D structure. International Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007.

Ikeuchi, K. and Miyazaki, D.
Digitally Archiving Cultural Objects. Springer, New York, NY, 2008.

Ikeuchi, K.
UTokyo's e-Heritage Project: 3D Modeling of Heritage Sites. http://www.youtube.com/watch?v=DPiMJkZ0YKI.

Baracchini, C., Brogi, A., Callieri, M., Capitani, L., Cignoni, P., Fasano, A., Montani, C., C. Nenci, C., Novello, R.P., Pingi, P., Ponchio, F., Scopigno, R.
Digital reconstruction of the Arrigo VII funerary complex, VAST 2004.

Tom Geller is an Oberlin, OH-based science, technology, and business writer.

Education

Web Used for Final Exams in Denmark

The government of Denmark says the Internet is such an integral part of daily life, it should be included not only in the classroom but in final exams.

Currently 14 colleges in Denmark are piloting a new system of allowing students full access to the Internet during exams. All Danish schools are being invited by the government to join the new Web-based system by 2011, according to BBC News.

Denmark has been an innovative country in the adaptation and use of new technology, and for more than a decade Danish students have been allowed to use computers to write their exam answers.

"Our exams have to reflect daily life in the classroom and daily life in the classroom has to reflect life in society," said Bertel Haarder, the minister for education. "The Internet is indispensable, including in the exam situation. I'm sure that [it] would be a matter of very few years when most European countries will be on the same line."

BBC News recently reported about Greve High School, one of the pilot schools, located south of Copenhagen, and students' participation in a Danish language exam. IT experts help students set up with their laptops, and CD-ROMs and exam papers are given to the students. Standing in the front of the classroom, one of the teachers instructed the students to use any Internet site they wanted to answer any of four questions (which focus on a student's ability to locate and analyze information, not to regurgitate facts and figures), but they could not message each other or email anyone outside of the classroom.

"If we're going to be a modern school and teach [students] things that are relevant for them in modern life, we have to teach them how to use the Internet," says Sanne Yde Schmidt, who leads the project at Greve.

Obituary | DOI:10.1145/1629175.1629187

Leah Hoffmann

Amir Pnueli: Ahead of His Time

Remembering a legacy of practical and theoretical innovation.

It is with great sadness that we note the death of Amir Pnueli, a pioneer in the fields of formal specification, verification, and analysis. Pnueli suffered a brain hemorrhage on November 2, 2009; his sudden death shocked the international computing community.

A shy and unassuming man, Pnueli was born in Nahalal, Israel, in 1941, and spent the bulk of his career at Tel Aviv University (where he founded the department of computer science), the Weizmann Institute of Science, and New York University. His modest demeanor belied groundbreaking technical achievements. In 1977, Pnueli's paper, "The Temporal Logic of Programs," marked a crucial turning point in the verification of concurrent and reactive systems. Temporal logic was developed by philosophers in the late 1950s to reason about the use of time in natural language. By introducing it to the field of formal methods, Pnueli gave researchers a set of tools that enabled them to specify and reason about the ongoing behavior of programs.

"That paper opened up a new world, both for him and for the field," says Moshe Y. Vardi, professor of computational engineering at Rice University. At the time, program verification was widely considered to be a hopeless challenge. But Pnueli's paper quietly established a framework for advanced techniques and gave new life to the domain. Throughout the rest of his career, Pnueli continued to refine his ideas and contribute to the development of other verification methods. He also coined the term "reactive system" to describe systems that maintain an ongoing interaction with their environment, and together with David Harel, a colleague at the Weizmann Institute, argued for its significance; the term has since become deeply ingrained in the literature. Working with a variety of collaborators—including current and former students—Pnueli made additional contributions to a number of related topics, from model checking to the synthesis of reactive modules. "He had boundless curiosity," says Harel.

Pnueli was deeply interested in developing techniques that could be used in industrial applications and not only research settings. He cofounded several companies, designing and supervising systems that included message switching, operating systems, and compilers. Statecharts, a visual language for the specification and design of reactive systems that he created with Harel, has subsequently been applied to areas as diverse as avionics and electronic hardware systems.

Pnueli's graciousness and humility endeared him to colleagues and students alike. He listened attentively to all who sought him out, and had a knack for finding the best in what they said. "People loved working with him because he made them feel smart," says Lenore Zuck, a former graduate student who is currently an associate professor at the University of Illinois at Chicago. At work, Pnueli was laid-back and informal, alternating between research and conversation and indulging in long digressions about politics, food, and the latest title from Am Oved, an Israeli publishing house. "It never felt tiring to work with him," says Zuck.

His graciousness and humility endeared Pnueli to colleagues and students alike.

Beneath his casual comportment lay deep insight and intuition, and collaborations, says Harel, often happened inadvertently. "We once started something while standing in line for lunch at the cafeteria," says Harel. "Another time, it was at a conference coffee break." In fact, their work on reactive systems began on a plane flight. While discussing Statecharts, the visual language he had recently created, Harel suggested it was useful for particular kinds of systems—systems that interact frequently with one another and don't do heavy computation. "And Amir said, 'Yes, I thought the same thing. Maybe we should call them *reactive systems,*' " recalls Harel. "I said, 'Bingo.' And then we spent the next hour talking about what we mean."

Pnueli's accomplishments were widely celebrated. In 1996, he received ACM's A.M. Turing Award for his work in temporal logic and contributions to program and system verification. These accomplishments were also honored in 2000 by the Israel Prize for Exact Sciences, the state's highest honor. More recently, Pnueli received the 2007 ACM Software System Award, with Harel and others, for their work on Statemate, a software engineering tool that evolved from the Statecharts language and supports visual, graphical specifications that represent the intended functions and behavior of a system.

At Pnueli's funeral, Harel challenged himself to identify one of his colleague's vices. The verdict: procrastination. "He was often late," recalled Harel. In Pnueli's defense, he continued, the issue was always eventually taken care of, and "in great depth, in detail, and combined with the great grace of his personality and his deep wisdom."

Leah Hoffmann is a Brooklyn-based technology writer.

PHOTOGRAPH BY DAVID HAREL

Technology | DOI:10.1145/1629175.1629184

Gary Anthes

Automated Translation of Indian Languages

India faces a daunting task trying to manually translate among 22 official languages, but assistance, in the form of advanced technology enabled by a lot of hard work, is on the way.

THE COMPLEXITY AND diversity of human languages makes automated translation one of the hardest problems in computer science. Yet the job is becoming more important as writing and speech are increasingly digitized and as the traditional separations between societies dissolve.

Few parts of the globe have as much need to translate from one language to another as does India. According to India's 2001 census, the country has 122 languages, 22 of which are designated as official languages by the government. The top six—Hindi, Bengali, Telugu, Marathi, Tamil, and Urdu—are spoken by 850 million people worldwide.

Now a decades-long effort by researchers is about to bear fruit. A multipart machine translation architecture, Sampark, is nearing completion as the combined effort of 11 institutions led by the Language Technologies Research Center at the International Institute of Information Technology in Hyderabad (IIIT-H).

Sampark combines both traditional rules- and dictionary-based algorithms with statistical machine learning, and will be rolled out to the public at http://sampark.iiit.ac.in/. By this month, systems for 12 out of 18 language pairs (nine languages) will be online and available for experimentation, with six more to follow soon after.

Many Indian languages are derived from Sanskrit, which is based on rules set down by Panini, the 4th century B.C. grammarian. Even those Indian languages that are not derived from Sanskrit are structurally similar to others in India. This common underpinning makes the translation from one Indian language to another easier than from, say, German to Chinese. Nevertheless, there are 462 pair-wise translations (counting each direction for a pair) possible among the 22 official Indian languages, so clearly the researchers had to find a generalized approach that could be easily adapted from one language to another.

India is the home of 122 languages, 22 of which are designated as official languages.

The chosen method, a transfer-based approach, consists of three major parts: analyze, transfer, and generate. First, the source sentence is analyzed, then the results are transferred in a standard format to a set of modules that turn it into the target language. Each step consists of multiple translation "modules."

An advantage of the three-step approach, says Rajeev Sangal, director of the Language Technologies Research Center, is that a particular language analyzer, one for Telugu, for example, can be developed once, independent of other languages, and then paired with generators in various other languages, such as Hindi.

The 13 major translation modules together form a hybrid system that combines rules-based approaches—where grammar and usage conventions are codified—with statistical-based methods in which the software in essence discovers its own rules through "training" on text tagged in various ways by human language experts.

A Transfer-Based Approach

Translation systems for major languages today—from companies like Google and Microsoft, for example—often use statistical approaches based on parallel corpora, huge databases of corresponding sentences in two languages. These systems use probability and statistics to learn by example which translation of a word or phrase is most likely correct. And they move directly from

PHOTOGRAPH BY ARTI SANDHU

source language to target language with no intermediate transfer step.

"The statistical direct translation approach is, in a sense, the lazy man's approach, because all it requires is that you go and hunt for parallel corpora and you turn the crank and you get what you get," says Srinivas Bangalore, a speech and language processing specialist at AT&T Research in Florham Park, NJ. "But the transfer-based approach is much more linguistically motivated, because you are trying to analyze the sentence and trying to arrive at something that is close to a representation of its meaning."

Parallel corpora are specialized databases consisting of sentences very carefully translated and then mapped one-for-one to their translations. Moreover, to do a good job of training translation systems, the parallel corpora must be very large—in the billions of sentences. "People are coming to grips with the fact that parallel data are not easy to come by," Bangalore says. "This is a very specialized kind of data."

Indeed, parallel corpora for many Indian language pairs do not exist and cannot easily be built, in part because not much Indian language text has been digitized. Nevertheless, developers at the Language Technologies Research Center were able to apply statistical machine learning in a limited way by annotating small monolingual corpora and analyzing the tagged text with statistical techniques, Sangal says.

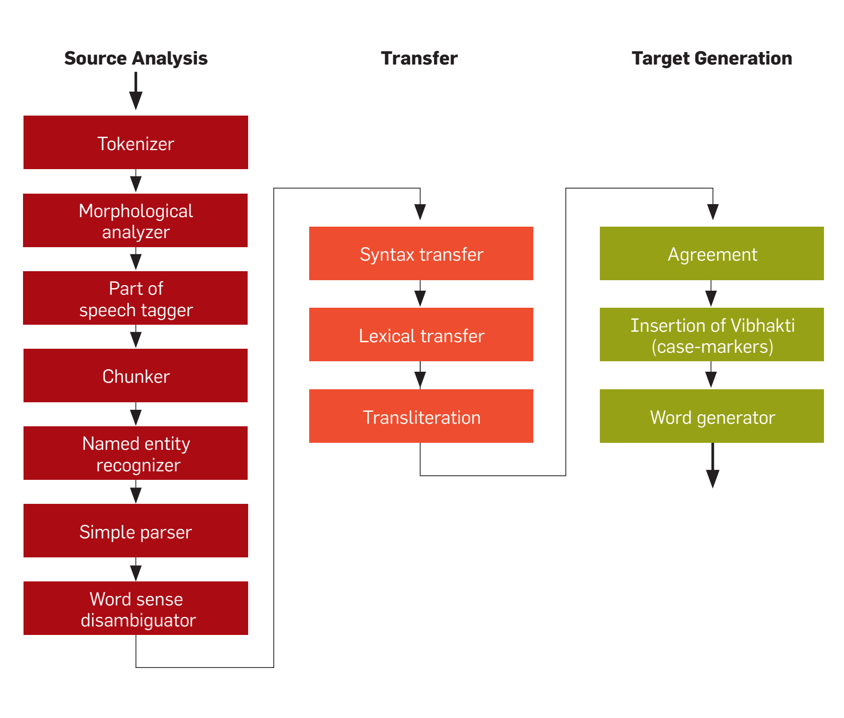
So although machine learning techniques were employed in some of the modules, developers painstakingly developed multilanguage dictionaries and codified rules in the Computational Paninian Grammar framework. They also held workshops of experts of all these languages to develop a standard tag set, and then used those tags to annotate the monolingual corpora.

"Most machine translation is not inspiration, it's perspiration," Bangalore says. "The hard part is building all the resources required, like dictionaries, morphological analyzers, parsers, and generators. It's a lot of grunt work."

Sangal says the effort that Sampark developers put into language analysis could have a broad impact beyond translating Indian languages. He says that even the best purely statistical systems can be made more accurate

How Sampark Works

An automated system for translating one Indian language to another, Sampark is a hybrid system consisting of traditional rules-based algorithms and dictionaries and newer statistical machine-learning techniques. It consists of three major parts and 13 modules arranged in a pipeline.



SOURCE ANALYSIS

Tokenizer: **Converts text into a sequence of tokens (words, punctuation marks, etc.) in Shakti Standard Format.**

Morphological analyzer: **Uses rules to identify the root and grammatical features of a word. Splits the word into its root and grammatical suffixes.**

Part of speech tagger: **Based on statistical techniques, assigns a part of speech, such as noun, verb or adjective, to each word.**

Chunker: **Uses statistical methods to identify and tag parts of a sentence, such as noun phrases, verb groups, and adjectival phrases, and a rule base to give it a suitable chunk tag.**

Named entity recognizer: **Identifies and tags entities such as names of persons and organizations.**

Simple parser: **Identifies and names relations between a verb and its participants in the sentence, based on the Computational Paninian Grammar framework.**

Word sense disambiguation: **Identifies the correct sense of a word, such as whether "bank" refers to a financial institution or a part of a river.**

TRANSFER

Syntax transfer: **Converts the parse structure in the source language to the structure in the target language that gives the correct word order, as well as a change in structure, if any.**

Lexical transfer: **Root words identified by the morphological analyzer are looked up in a bilingual dictionary for the target language equivalent.**

Transliteration: **Allows a source word to be rendered in the script of the target language. Useful in cases where translation fails for a word or a chunk.**

TARGET GENERATION

Agreement: **Performs gender-number-person agreement between related words in the target sentence.**

Insertion of Vibhakti: **Adds post position and other markers that indicate the meanings of words in the sentence.**

Word generator: **Takes root words and their associated grammatical features, generates the appropriate suffixes and concatenates them. Combines the generated words into a sentence.**

by first doing the types of detailed language analysis employed in Sampark. “What one can do in the future is to first do monolingual analysis of one or both sides in paralleled corpora, and then use that to improve the quality of machine learning from the parallel corpora,” he says. “So what we have done would also be useful if larger parallel corpora became available tomorrow.”

Another advantage of the transfer approach, says AT&T’s Bangalore, is its generalizability. “If you give me a parallel corpus dealing with financial news, and I train it up with millions of sentences of that sort, and two days later you say, ‘Translate a sports article,’ it’s not going to perform as well.”

But that kind of application domain change has been explicitly anticipated by Sampark’s developers. The first version, rolling out now language-by-language, is general purpose and optimized for tourism-related uses, but it will be made available to large users who wish to customize it for other domains, says Dipti Sharma, an associate professor at IIIT-H. That would involve building a new domain dictionary, incorporating rules that handle domain-specific grammatical structures, and perhaps retraining some modules such as Part of Speech Tagger and Named Entity Recognizer.

The effort required to make those changes is minimized by building on the existing multilingual dictionary, Sharma says. It is sense- or meaning-based, so that for one domain or language, “bank,” for example, would most likely represent a financial institution, but for another it might refer to the edge of a river, Sharma says. The dictionary currently allows translation among nine languages.

Sangal says the language-translation system has two especially noteworthy attributes. First, the linguistic analysis based on Panini is “extremely good,” he says. “It was initially chosen for Indian languages, but we find it is also suitable for other languages.” Initially, hard work is needed, he says, in setting it up by developing standards for parts-of-speech tags and dependency tree labels and for figuring out ways to handle unique language constructs.

The second attribute of special note is the system’s software architecture. It is an open architecture in which all modules produce output in Shakti Standard Format (SSF). The architecture allows modules written in different programming languages to be plugged in. Readability of SSF helps in development and debugging because the input and output of any module can be easily seen. Also, a dashboard tool supports the architecture in a variety of ways. Custom written, it is “extremely robust,” Sangal says. “If a module fails to perform a proper analysis, the next module will still work, albeit in a degraded mode. So the system never gives up; it always tries to produce something.”

Further Reading

Naskar, S. and Bandyopadhyay, S.
Use of machine translation in India: current status. ***Machine Translation Review 15*, Dec. 2005.**

Bharati, A., Sangal, R., Mishra, D., V., Sriram, T., Papi Reddy
Handling multi-word expressions explicit linguistic rules in an MT system. *Proceedings of the Seventh International Conference on Text, Speech and Dialogue, 2004.*

Lavie, A., Vogel, S., Levin, L., Peterson, E., Probst, K., Llitjos, A.F., Reynolds, R., Carbonell, J., Cohen, R.
Experiments with a Hindi-to-English transfer-based MT system under a miserly data scenario. *ACM Trans. on Asian Language Processing 2*, 2, June 2003.

Bharati, A., Chaitanya, V., Kulkarni, A., Sangal, R., Umamaheshwara Rao, G.
Anusaaraka: overcoming the language barrier in India. *Anuvad: Approaches to Translation*, Sage, New Delhi, 2001.

Manning, C.
***Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, 1999.**

Bharati, A. and Sangal, R..
Parsing free-order languages in the Paninian framework. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, June, 1993.

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

Milestones

SIGUCCS Hall of Fame and Other CS Awards

Select members of the computer science community were recently honored for their innovative service and research.

SIGUCCS HALL OF FAME

Recognized for their years of service, the 2009 inductees to the SIGUCCS Hall of Fame are Nancy Bauer-Runyan, Ross University; Jim Kerlin; Lynnell Lacy, University of Illinois at Urbana-Champaign; Teresa (Terry) Lockard, University of Virginia; and Glenn Ricart.

GORDON BELL PRIZES

The Gordon Bell Award was presented at SC09 to recognize outstanding achievement in high-performance computing applications. The purpose of the award is to track the progress over time of parallel computing, with particular emphasis on rewarding innovation in applying high-performance computing to applications in science.

A team led by Tsuyoshi Hamada of Nagasaki University won for its paper “42 TFlops Hierarchical N-body Simulations on GPUs with Applications in both Astrophysics and Turbulence” in the lower price/performance category.

There were two winners in the special category. A team led by David E. Shaw of D.E. Shaw Research won for its paper “Millisecond-Scale Molecular Dynamics Simulations on Anton” and a team led by Rajagopal Ananthanarayanan of IBM Almaden Research Center won for its paper “The Cat is Out of the Bag: Cortical Simulations with 10^9 Neurons, 10^{13} Synapses.”

In the peak performance category, a team led by Markus Eisenbach of Oak Ridge National Laboratory won for its paper “A Scalable Method for Ab Initio Computation of Free Energies in Nanoscale Systems.”

PENNY CRANE AWARD

Robert Paterson, vice president for information technology, planning and research at Molloy College, received the Penny Crane Award at SIGUCCS in recognition of significant contributions to SIGUCCS and computing in higher education.

DIANA AWARD

On behalf of Apple, Sandy Korzenny, director of Apple product documentation, received the Diana Award, which SIGDOC presents every two years to an organization, institution, or business for its long-term contribution to the field of communication design.

Science | DOI:10.1145/1629175.1629183

Neil Savage

New Search Challenges and Opportunities

If search engines can extract more meaning from text and better understand what people are looking for, the Web's resources could be accessed more effectively.

The Web is a huge, dynamic landscape of information, and navigating through it not an easy task. There are billions of Web pages, and the type of content is expanding dramatically, with blogs and Twitter feeds, maps and videos, photos and podcasts. People, typing on a computer in their cubicle or using their smartphone on a street corner, are trying to sift through this growing morass of data, looking for everything from car repair advice to a nearby Thai restaurant that's not too expensive. For search engines, this enormous variety of data and formats is providing both new challenges and new opportunities.

"The ability to produce information and store information has far outstripped human cognitive capacity, which is basically fixed," says Oren Etzioni, a professor of computer science and engineering at the University of Washington. "The haystack keeps getting bigger. Obviously we need better and better tools to find the proverbial needles."

Today's search engines do a fine job of cataloging text, counting links, and delivering lists of pages relevant to a user's search topic. But in the coming decade, Etzioni believes, search will move beyond keyword queries and automate the time-consuming task of sifting through those documents. With a better understanding both of what documents mean and what searchers are looking for, he predicts, some tasks could be reduced from hours to minutes.

Etzioni is attempting to get more information out of text using a technique called open information extraction, which is built on a long-used technology that examines natural language text and tries to derive data about the relationships between words. An algorithm looks for triples, which follow the structure of entity-relationship-entity, such as "Beijing is the capital of China" or "Franz Kafka was born in Prague." The system is open because it derives the relations from the structure of the language rather than relying on hand-labeled examples of relationships, which would not be scalable to the Web as a whole.

PHOTOGRAPH COURTESY OF THE UNIVERSITY OF WASHINGTON

Like many other computer scientists, the University of Washington's Oren Etzioni is developing new tools for searching the Web's growing morass of text, images, and other content.

Oren Etzioni's approach examines natural language text and tries to derive data about the relationships between words.

Etzioni developed a program called TextRunner that uses a general model of language to assign labels to words in a sentence, then to calculate the beginning and end of strings of words that contain the entity-relationship-entity structure. It extracts those triples so they can be indexed and searched. A searcher who asks "Where was Kafka born?" should quickly receive a precise answer, not just a list of pages that contain the words "Kafka" and "born." Given the vast number of Web pages, Etzioni says, the search engine should be able to notice errors such as one page saying Kafka's birthplace is Peking is less likely to be correct, for example, than the tens of thousands that say Prague.

It's more challenging for a computer to extract more subjective data from text, such as judgments about hotels or movies, but a well-designed algorithm can figure out cues, such as which descriptive phrase is stronger: clean, almost spotless, or sparkling. It should be able to distinguish the positive—"The room was nice and quiet"—from the negative—"I was disappointed the room wasn't quieter."

Blog and Twitter Searches

One growing area that poses new challenges for search engines is social media, such as blogs, Twitter feeds, and Facebook status updates. "I don't think we have really good blog search yet," says Marti A. Hearst, a professor in the University of California, Berkeley School of Information. Along with Microsoft researchers Susan T. Dumais and Matthew Hurst, Hearst says blog search should be able to accomplish three tasks: find out what people are thinking about a certain topic over time; suggest blogs that are good to read for their style, personality, and other criteria; and find useful information in older blog posts, along the lines of standard search of more static documents.

Blog search needs to take into account the differences between blogs and traditional documents, such as the former's use of more informal language, their different link topology, the importance of timeliness, and the fact that updates tend to not be full HTML pages. Blog search must also take into account that much of the information on blogs is subjective.

To accomplish these tasks, search engine designers look for representations of features that might belong to a particular class of posting, such as the readability level of a page. Machine learning algorithms can then figure out that particular distributions of features may be characteristic of a certain class.

"If you have labeled data and examples of things that you think have a particular attribute, then you can use that to find something similar," says Dumais, principal researcher in Microsoft Research's Adaptive Systems and Interaction Group. But rating postings as positive or negative, or figuring out whether they're aimed at an older or younger audience or have a left-leaning, right-leaning, or middle-of-the-road viewpoint, is challenging, she says. "They do involve a richer understanding of language than most search engines have," Dumais notes.

Search can be improved through a deeper understanding of a document's meaning and a better grasp of a searcher's intentions.

Twitter use has grown explosively in recent months, and in October the company made a deal to open its data to Microsoft's search engine. Dumais says that, with its 140-character limit leading to creative abbreviations of words and condensed hyperlinks, searching Twitter will pose some interesting challenges. But once those are tackled, Twitter users should be able to conduct more refined searches than the service currently allows, while the flow of Twitter data provides search designers with new information that may make search richer. "The volume of the content [on the Web] is actually very useful for some types of algorithms," Dumais says.

One useful fact is that people with Twitter feeds and Facebook pages are making public a lot of information about themselves that search engines can use to better understand their search queries. Just as search can be improved through a deeper understanding of what documents mean, it can also improve through a better grasp of the searcher's intentions. "The real issue with a search engine is not just to serve up results, but to help people accomplish what they're trying to do," says Jon Kleinberg, a professor of computer science at Cornell University.

Search engines trying to provide the right answer to a query might take into account what a user has previously searched for. If a user is looking for a restaurant or a movie recommendation, the search engine might look at the user's friends lists and see what those presumably trusted sources liked. And if the user is searching from a mobile device, that might provide additional clues.

If nothing else, a search from a mobile phone tells the search engine it is from a phone, so perhaps a search for a person is really a search for their phone number. And many mobile devices use GPS or cell phone towers to determine their location. A person typing "Yankees" in Manhattan may be looking for tickets to tonight's baseball game, whereas the same search in Seattle may represent a desire for last night's score. "In a relatively short time frame, we're going to think of geolocation as an integral component of a lot of the online activity we do," Kleinberg says.

Time is also becoming a characteristic to take into account, Kleinberg says. One way of judging the importance of a news story, for instance, is how quickly it spread and how long interest focused on it. Dumais points out that many facts have a time component as well. The gross national product of Norway, the population of Brazil, and the prime minister of Japan—all can have one factual answer in 2000 and a different one in 2010.

Dumais says future search engines will have both a better grasp of the intent of a query and a richer understanding of Web content. "We're looking at how we can support that in ways that go beyond 2.3 words typed into a search box and 10 blue links," she says. C

Further Reading

Etzioni, O., Banko, M., Soderland, S., Weld, D.
Open information extraction from the Web. ***Commun. of the ACM*** **51, 12, 2008.**

Hearst, M.A.
Emerging Trends in Search Interfaces. **Cambridge University Press, New York, NY, 2009.**

Backstrom, L., Kleinberg, J., Kumar, R., Novak, J.
Spatial variation in search engine queries. ***Proc. 17th Int'l Conf. on World Wide Web*****, 2008.**

Hearst, M.A., Hurst, M., Dumais, S.T.
What should blog search look like? ***Proc. of the 2008 ACM Workshop on Search in Social Media*****, 2008.**

Downey, D., Dumais, S.T., Liebling, D., Horvitz, E.
Understanding the relationship between searchers' queries and information goals. ***Proc. 17th ACM Conf. on Information and Knowledge Management*****, 2008.**

Neil Savage is a science and technology writer based in Lowell, MA.

Conference | DOI:10.1145/1629175.1629186

Kirk L. Kroeker

Future Internet Design Summit

The National Science Foundation's meeting on Internet architectures focused on designs related to emerging social and economic realities.

As part of its Future Internet Design initiative, the National Science Foundation (NSF) is establishing funding in 2010 for several multimillion-dollar research projects. From October 12–15, 2009 at the Waterview Conference Center in Arlington, VA, the NSF held a Future Internet Architectures Summit to gain input from the research community about developing calls for proposals related to this funding. Summit organizers designed the format of the four-day event to facilitate multidisciplinary collaboration in detailing new Internet architectures built in response to emerging social and economic realities.

Ninety researchers attended the multidisciplinary summit on future Internet architectures.

PHOTOGRAPH BY SCOTT MIDDLETON, KNOWINNOVATION LTD.

Altogether, the invitation-only summit drew 90 U.S.-based researchers with expertise in networking, communications, security, privacy, and the social and economic sciences. "The participants were engaged in the process and made progress toward the articulation of future network architectures," says Ty Znati, division director for computing and network systems in the NSF's Computer and Information Science and Engineering (CISE) division. "They also began to assemble and integrate coherent ideas and building blocks into candidate architectures."

Znati says that while CISE is especially interested in stimulating the multidisciplinary exploration of future Internet architectures, focusing specifically on one type of architecture or network design was not a goal of the summit. "The objective was to expand the scope of research from a component-focused agenda to the design of overarching architectures for future networks," says Znati. "It is not clear what the right architecture of future networks will be, so the summit encouraged participants to think about potential architectures from multiple perspectives."

Objectives discussed at the event for candidate architectures were far-ranging. One proposal suggested designing networks to function in the type of infrastructure-poor environments typical of developing nations. Another suggested designing network architectures for human identity, with the idea being that communication is person- rather than device-centric. A third suggested designing a network architecture that is more connected to the physical world so that it would be capable not only of information transport, but also of control, actuation, and sensing.

"The summit succeeded in paving the way to the development of interdisciplinary communities that currently do not exist, and in fostering collaborative free thinking that otherwise would not have been possible," says Znati.

In the end, there was no discussion about current Internet issues, such as IPv6 or Internet Corporation for Assigned Names and Numbers (ICANN) governance. "While governance will be an issue in 15 years, the current and short-term future of ICANN is not relevant," says David Clark, a senior research scientist at Massachusetts Institute of Technology. "I don't think anyone said 'ICANN' the whole time. Those are today's arguments, not tomorrow's arguments."

Clark has been helping the NSF organize its Future Internet Design program, the goal of which is to envision and then design what could be the Internet of 15 years from today. "In this meeting, people had many ideas about how to make that goal more specific, but there was no desire for consensus," he says. "This was a meeting to develop different ways of responding to that challenge."

Following a formal report on the summit proceedings, the NSF will send out calls for proposals. The proposals will be reviewed by an expert panel, which will make recommendations with the expectation that the NSF will fund up to four projects. C

Based in Los Angeles, **Kirk L. Kroeker** is a freelance editor and writer specializing in science and technology.

In Memoriam | DOI:10.1145/1629175.1629188

Karen A. Frenkel

Robert Lovett Ashenhurst 1929–2009

FORMER *COMMUNICATIONS* EDITOR-IN-CHIEF Robert L. Ashenhurst, who died last October at age 80, served ACM for 35 years with dedication, humor, and panache, according to fellow ACM volunteers. M. Stuart Lynn (*Communications* editor Jan. 1969–Mar. 1973) lauded him as a "distinguished computer scientist and dedicated editor." Much of Ashenhurst's thinking contributed to the current structure of ACM's journals, Transactions, SIG publications, and magazines, said Peter J. Denning (*Communications* editor Feb. 1983–Sept. 1992), who remembered him as "unassuming and unflashy, and yet one of the most influential people of all time in shaping ACM."

At the time of his death, Ashenhurst was professor emeritus of applied mathematics at the University of Chicago's Booth School of Business. He joined ACM in 1956; just months before joining the University's faculty. Indeed, he would become the first chair of the University's Committee on Information Systems, a predecessor of its Department of Computer Science.

Ashenhurst's ACM contributions were plentiful. Lynn recalls that it was during his editorship that he and Ashenhurst launched the popular Forum department, where many lively editorial debates took form. As Forum's first editor (a position he would hold for 20 years), Ashenhurst channeled his considerable diplomatic talents. "I always had the feeling that Bob received his biggest kicks in handling letters, which ranged from thoughtful to outrageous, from gravely serious to downright funny," said Lynn. "No opinion was too trivial to be published, but Bob always brought a helping hand to authors—and a sense of humor to sorting through the conflicts of hotly held opinions."

In those days, ACM elections for president were often contested, Denning reminisced. "There were frequent petition candidates and nasty campaigns," he said, "but Bob remained steadfastly impartial and was about the only one that candidates from any faction trusted to give them a fair shake in the Forum."

Ashenhurst served as editor-in-chief of *Communications* from 1973–1983. During that time he helped steer the transition of *Communications* from a research publication to a "journal for all members." He created a hierarchy of reviews for ACM periodicals—refereeing (peer review for research papers), formal reviewing (for articles that were not original research), reviewing (an informal, faster process for *Communications* articles), and unreviewed (for *The Guide* and SIG newsletters). He was also quick to recognize ACM's practitioner base, advocating the creation of the Computing Practices section within *Communications* to present real-world applications and industry-based articles.

Former ACM President Adele Goldberg recalled a controversy erupting within the ACM Publications Board over whether the government should review publications in parallel with the ACM review process and without author approval. Goldberg argued that if an author had submitted a paper, his or her employer had already cleared it in accord with any sponsor requirements. Many thought she was making too much of the issue, she said, but Ashenhurst fashioned a compromise: to advertise to authors the government's interest and willingness to review papers in advance of submitting. "That way, it was clearly up to the author, not us," Goldberg said, and it satisfied those at the ends of the political spectrum. When the U.S. government did black out sections of papers for a cryptography conference proceedings in the early 1980s, Goldberg remembered Ashenhurst smiling at her in recognition of a fight well fought.

Ashenhurst's magnanimity as an educator was also an ACM magnet. He was "enormously supportive, kind, and did not hold himself separate from students," said Goldberg, who was Ashenhurst's graduate student at the University of Chicago. When she expressed interest in education technology, for example, he not only suggested she spend a year at Stanford University, but also arranged for it. There she stayed, ultimately landing at Xerox PARC.

Ashenhurst was named an ACM Fellow in 1995 and in 1998 ACM recognized his years of service with its Outstanding Contribution to ACM Award. (See http://cacm.acm.org/news/49494-robert-l-ashenhurst-former-communications-editor-in-chief-dies/fulltext.)

Though soft-spoken and diplomatic, as he proved during his two decades as Council Parliamentarian, Ashenhurst also had a spontaneous, flamboyant side. Many remember his brilliant piano playing at après ACM Council parties. "He knew songs that seemed to stretch back to the dawn of light music," said Lynn. "Jim Adams (ACM's former Director of Membership) knew all the words, but Bob knew all the notes—and even when he didn't, he somehow managed to find them on the spot." C

Karen A. Frenkel is a freelance writer and editor in NYC specializing in science and technology

PHOTOGRAPH BY NANCY LEVNER

Chapters | DOI:10.1145/1629175.1645395

Samuel Greengard

ACM and India

ACM India aims to become a key information exchange and networking organization for the nation's professional and student communities.

Over the last decade, India has emerged as an economic and technology powerhouse. Although the nation of 1.2 billion has boasted ACM members as far back as the 1960s—it currently has about 1,800 professional and 1,300 student members participating in the organization—the country has lacked an official ACM presence. "Given that there are somewhere between 1.5 million and 2 million computing professionals in India it makes sense for ACM to play a key role," says Mathai Joseph, advisor for Tata Consultancy Services and a member of the ACM Council and the new ACM India Council.

SATELLITE MAP BY GOOGLE EARTH

ACM is in the process of establishing ACM India (http://india.acm.org/) as a legal entity and will hold its first conference in late January. Four A.M. Turing award winners, including Barbara Liskov of the Massachusetts Institute of Technology, will speak at the inaugural event in Bangalore. In the coming months, ACM is looking to hold more conferences and create chapters in areas with concentrations of technology and computing professionals. Potential cities include Hyderabad, Pune, Mumbai, Delhi, and Kolkata. ACM India has also introduced 28 student chapters at various educational institutions throughout the country.

The opportunity for professional development and networking offers a great deal of potential. More than 150,000 young people start work at Indian companies each year and approximately two-thirds of these information workers are below the age of 30. "There is a lot of energy and a keenness to learn new things," Joseph explains.

Yet there are also serious challenges to confront and ACM India hopes to play a lead role in affecting positive change. For one thing, India suffers from a dearth of qualified teachers in computer science, technology, and the application of IT skills. As a result, the organization will work to attract individuals to teaching and develop the skills and knowledge required to educate students in computer science, technology, and related practices.

Also, Indian industry has long focused on limited training at the expense of overall education. "There is far too much training that's associated with specific products or tools and too little of a focus on the long-term," Joseph observes. ACM India will aim for a more holistic and focused approach to learning by working with political leaders, educational institutions, and industry to develop better systems and methods.

ACM India will work to attract individuals to teaching and develop the skills and knowledge required to instruct students in computer science, technology, and related practices.

Finally, some cultural issues and biases exist. "In India, there is general acceptance of the traditional sciences like physics, chemistry, and biology, but general skepticism that there is any science associated with computing," Joseph says. "The general thinking is that computing is all about programming, so we have a lot of educating to do, and it will take time."

Ultimately, ACM India hopes to become the voice of the Indian computing community and influence public discourse and political decision-making. The Indian branch of ACM hopes to have 10,000-plus members within the next few years. The goal is to make ACM a key information exchange and networking center for the professional and student communities.

"Like many other parts of the world, India is undergoing rapid change," says Joseph. "There is enormous enthusiasm for creating an active ACM organization in India. It offers a great deal of potential."

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

DOI:10.1145/1629175.1629189

Michael Cusumano

Technology Strategy and Management
The Evolution of Platform Thinking

How platform adoption can be an important determinant of product and technological success.

In several of my prior publications, including my *Communications* columns on Microsoft, Apple, and Google, I have argued that companies in the information technology business are often most successful when their products become industrywide *platforms*. The term "platform," though, is used in many different contexts and can be difficult to understand. I am currently finishing a book on best-practice ideas in strategy and innovation, and include a chapter on how platform thinking has evolved.[1] This column summarizes some of my findings.

Most readers have probably heard the term platform used with reference to a foundation or base of common components around which a company might build a series of related products. This kind of in-house "product platform" became a popular topic in the 1990s for researchers exploring the costs and benefits of modular product architectures and component reuse.[2] I was among this group, having studied reusable components and design frameworks in Japanese software factories, reusable objects at Microsoft, and reusable underbody platforms at automobile manufacturers.[3]

Product versus Industry Platforms

In the mid- and late 1990s, various researchers and industry observers, including myself, also began discussing technologies such as Microsoft Windows and the personal computer, as well as the browser and the Internet, as "industrywide platforms" for information technology. Most of us saw the PC as competing with an older industry platform—the IBM System 360 family of mainframes. It took a few more years to devise frameworks to help managers use the concept of an industry platform more strategically. One of my doctoral students, Annabelle Gawer, took on this challenge for her MIT dissertation in the late 1990s, which became the basis for our 2002 book, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. In this book and subsequent articles we tried to clarify the characteristics of a product versus an industry platform.[4]

Gawer and I argued that an industry platform has two essential differences. One is that, while it provides a common foundation or core technology that a firm can reuse in different product variations, similar to an in-house product platform, an industry platform provides this function as part of a technology "system" whose components are likely to come from different companies (or maybe different departments of the same firm), which we called "complementors." Second, the industry platform has relatively little value to users without these complementary products or services. So, for example, the Windows-Intel personal computer or a smartphone are just boxes with relatively little or no value without software development tools and applications or wireless telephony

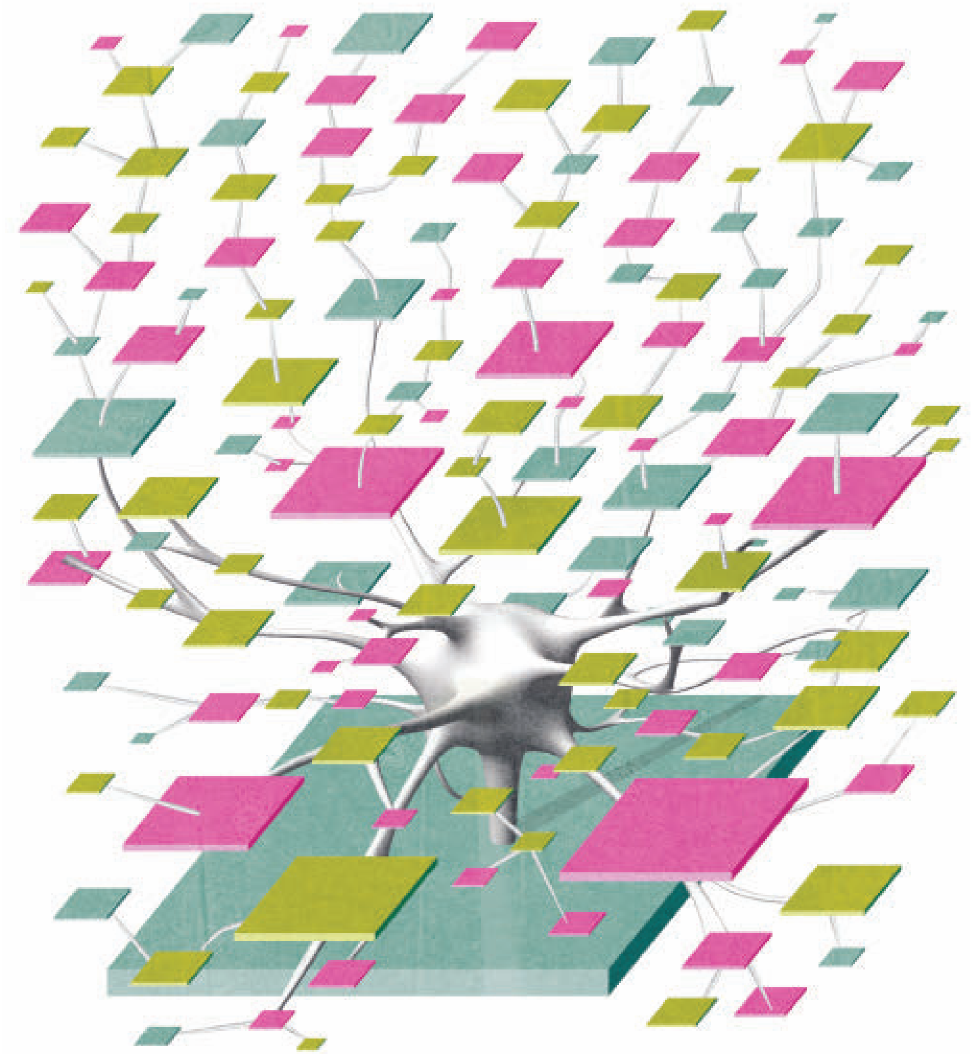


ILLUSTRATION BY JOHN HERSEY

and Internet services. The company that makes the platform is unlikely to have the resources or capabilities to provide all the useful applications and services that make platforms such as the PC or the smartphone so compelling for users. Hence, to allow their technology to become an industrywide platform, companies generally must have a strategy to open their technology to complementors and create economic incentives (such as free or low licensing fees, or financial subsidies) for other firms to join the same "ecosystem" and adopt the platform technology as their own.

A second key point is that, as various authors have noted, the critical distinguishing feature of an industry platform and ecosystem is the creation of "network effects." These are positive feedback loops that can grow at geometrically increasing rates as adoption of the platform and the complements rise. The network effects can be very powerful, especially when they are "direct," such as in the form of a technical compatibility or interface standard—which exists between the Windows-Intel PC and Windows-based applications or between VHS or DVD players and media recorded according to those formats. The network effects can also be "indirect," and sometimes these are very powerful as well—such as when an overwhelming number of application developers, content producers, buyers and sellers, or advertisers adopt a particular platform that requires complements to adopt a specific set of technical standards that define how to use or connect to the platform. We have seen these kinds of interface or format standards, and powerful network effects, with the Windows-Intel PC and application development services on the eBay, Google, Amazon, and Facebook social networking portals as well as new electronic book devices, among many others.

Most important with a network effect is that the more external adopters in the ecosystem that create or use complementary innovations, the more valuable the platform (and the complements) become. This dynamic, driven by direct or indirect network effects or both, encourages more users to adopt the platform, more complementors to enter the ecosystem, more users to adopt the platform and the complements, almost ad infinitum.

Standards Are Not Platforms

We have seen many platform-like battles and network effects in the history of technology, mainly in cases with incompatible and competing standards. It is important to realize, though, that standards by themselves are not platforms; they are rules or protocols specifying how to connect components to a platform, or how to connect different products and use them together. Prominent historical examples of platforms incorporating specific standards include the telegraph, telephone, electricity, radio, television, video recording and, of course, the computer. Understanding how standards initiatives are likely to play out is often an essential part of understanding which platform is likely to win the majority of a market, if one winner is likely to emerge.

Not surprisingly, there has been a growing amount of both theoretical and empirical research on industry platforms, particularly in economics but also in strategy and management of technology. Competition in the consumer electronics and computer industries spurred a great deal of thinking on this topic beginning in the early 1980s, just as the arrival of the Web did in the mid-1990s. Influential early work by economists mostly took the form of theory and models with few detailed case studies. This is still a relatively new topic and there are few large-sample studies. But the key concepts are all there—how platform industries or products are affected by standards and technical compatibility, the phenomenon of network or positive feedback effects, and the role of switching costs and bundling.[5] Switching costs and bundling have become strategically important because companies often can attract users to their platforms by offering many different features for one low price, and can retain users by making it technically difficult to move to another platform. This is why, for example, cable and telephone companies now compete to offer bundled voice, data, and video services to the home.

Another important insight for managers from the economics research is that platform industries tend to have more than one market "side" to them.[6] We can see this clearly in the personal computer industry. Microsoft and Apple compete not merely to attract end users to their products. They also have to attract software and hardware firms to build applications products and peripheral devices, such as printers and Webcams. In newer "multi-sided" platform markets such as social networking or Internet media, Google, Microsoft, Facebook, and other companies compete not simply for end users and application developers, but also for a third segment of the market—advertisers. Companies that like to sell video clips have an even more complicated market challenge. They have to attract not only end users, application developers, and advertisers, but also producers of content as well as aggregators of other people's content.

Even in simple two-sided markets, strategy and pricing can get complicated quickly.[7] In 1998, for example, David Yoffie and I wrote a book called *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft* that looked at how Netscape and Microsoft used one-sided subsidies, following the mantra of "free, but not free"—give one part of the platform away, such as the browser, but charge for the other part, such as the server or Windows.[8] Adobe has done the same thing by giving away the Acrobat Reader and charging for its servers and editing tools. Or firms can give one part of the platform away to some users (students or the general consumer) but charge others (corporate users). We also discussed the strategy of "open, but not open"—make access to the interfaces easily available but keep critical parts of the technology proprietary or very distinctive, such as Netscape did with the Navigator browser and its server, special versions of programming languages, and intranet and extranet combinations. Microsoft has done this with the entire set of Windows technologies, including Office and other applications.

We are still in the early stages of understanding how common and important industry platforms really are.

Other researchers have done important theoretical and empirical work on what makes for a "winner-take-all" market.[9] The conclusion seems to be that as long as there is room for companies to differentiate their platform offerings, and consumers can easily buy or use more than one platform, then it is unlikely for one dominant platform to emerge—unless the direct or indirect network effects are overwhelmingly strong. This is why the video game market has not seen one clear platform winner. The platforms (the consoles from Sony, Microsoft, and Nintendo) are different enough, most users can afford to buy more than one console (they are subsidized by the makers, who hope to make money from software fees), and truly hit complements (the games) often become available on all three platforms.

Conclusion

We are still in the early stages of understanding how common and important industry platforms really are. Apart from the examples I discussed in this column, new battles keep appearing in technologies ranging from electronic payment systems to electronic displays, automotive power systems, long-life batteries, and even the human genome database (for disease research and new drug discovery). The closer we look at modern technologies, the more likely we are to see platforms, and even platforms embedded within platforms. Who wins and who loses these competitions is not simply a matter of who has the best technology or the first product. It is often who has the best platform strategy and the best ecosystem to back it up.

References and Further Reading

1. M.A. Cusumano, *Staying Power: Six Enduring Ideas for Managing Strategy and Innovation in a Changing World* (Oxford University Press, forthcoming 2010), based on the 2009 Oxford Clarendon Lectures in Management Studies.
2. For managerial perspectives on product platforms, see M.H. Meyer and A.P. Lehnerd, *The Power of Product Platforms* (Free Press, 1997) and S.W. Sanderson and M. Uzumeri, *Managing Product Families* (Irwin, 1996). For more academic treatments, see M.H. Meyer and J.M. Utterback, "The Product Family and the Dynamics of Core Capability," *MIT Sloan Management Review 34*, 3 (1993), 29–47; K. Ulrich, "The Role of Product Architecture in the Manufacturing Firm," *Research Policy 24*, 3 (1995), 419–440; and C.Y. Baldwin and K.B. Clark, *Design Rules: The Power of Modularity, Volume 1* (MIT Press, 1999).
3. See M.A. Cusumano, *Japan's Software Factories* (Oxford University Press, 1991); M.A. Cusumano and K. Nobeoka, *Thinking Beyond Lean* (Free Press, 1998); and M.A. Cusumano and R.W. Selby, *Microsoft Secrets* (Free Press, 1995).
4. A. Gawer and M.A. Cusumano, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation* (Harvard Business School Press, 2002). Also, A. Gawer and M.A. Cusumano, "How Companies Become Platform Leaders," *MIT Sloan Management Review 49*, 2 (2008), 29–30.
5. See, for example, P. David, "Clio and the Economics of QWERTY," *American Economic Review 75*, 2 (1985), 332–337; J. Farrell and G. Saloner, "Installed Base and Compatibility: Innovation, Product Preannouncements and Predation," *American Economic Review 76*, 5 (1986), 940–955; W.B. Arthur, "Competing Technologies, Increasing Returns, and Lock-in by Historical Events," *Economic Journal 99* (Mar. 1989), 116–131; M. Katz and C. Shapiro, "Product Introduction with Network Externalities," *Journal of Industrial Economics 40*, 1 (1992), 55–83; C. Shapiro and H. Varian, *Information Rules: A Strategic Guide to the Network Economy* (Harvard Business School Press, 1998); and Y. Bakos and E. Brynjolfsson, "Bundling Information Goods: Pricing, Profits and Efficiency," Management Science 45, 12 (1999), 1613–1630.
6. See T. Bresnahan and S. Greenstein, "Technological Competition and the Structure of the Computer Industry," *Journal of Industrial Economics 47*, 1 (1999), 1–40; R. Schmalensee, D. Evans, and A. Hagiu, *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries* (MIT Press, 2006); C. Rochet and J. Tirole, "Platform Competition in Two-sided Markets," *Journal of the European Economic Association 1*, 4 (2003), 990–1029; and J.C. Rochet and J. Tirole, "Two-sided Markets: A Progress Report," *RAND Journal of Economics 37*, 3 (2006), 645–667.
7. See D.B. Yoffie and M. Kwak, "With Friends Like These: The Art of Managing Complementors," *Harvard Business Review 84*, 9 (2006), 89–98; and R. Adner (2006), "Match Your Innovation Strategy to Your Innovation Ecosystem," *Harvard Business Review 84*, 4 (2006), 98–107.
8. M.A. Cusumano and D.B. Yoffie, *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft* (Free Press, 1998), 97–100, 133–138.
9. See G. Parker and M.W. Van Alstyne, "Two-Sided Network Effects: A Theory of Information Product Design," *Management Science 51*, 10 (2005), 1494–1504; T. Eisenmann, "Internet Companies Growth Strategies: Determinants of Investment Intensity and Long-Term Performance," *Strategic Management Journal 27*, 12 (2006), 1183–1204; T. Eisenmann, G. Parker, and M.W. Van Alstyne, "Strategies for Two-Sided Markets," *Harvard Business Review 84*, 10 (2006), 92–101; and T. Eisenmann, G. Parker, and M.W. Van Alstyne, "Platform Envelopment," Unpublished working paper, 2007. Also, for a collection of articles, see A. Gawer, Ed., *Platforms, Markets and Innovation* (Edward Elgar, 2009).

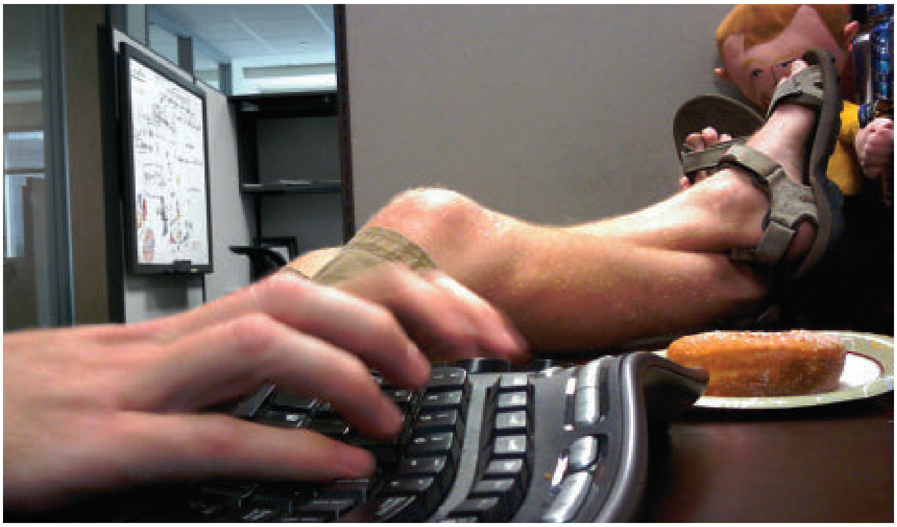
Michael Cusumano (cusumano@mit.edu) is Sloan Management Review Distinguished Professor of Management and Engineering Systems at the MIT Sloan School of Management and School of Engineering in Cambridge, MA.

DOI:10.1145/1629175.1629190

Phillip G. Armour

The Business of Software
In Praise of Bad Programmers

A tale illustrating the difference between individual and team skills.



PHOTOGRAPH BY FLICKR USER SLWORKING2

HAROLD WAS A bad programmer—a *really* bad programmer—the kind who owed it to himself and all around him to find another profession. But Harold was a nice guy and he was a lifer; he'd been with the company for a long, long time. He had been a low-level programmer forever, he never got promoted, he received minimum salary increases each year, and he got moved around a lot. But nobody wanted to fire him. So every time a project started up and needed headcount the manager who had Harold on his team at the time took the opportunity to unload him onto the next person unfortunate enough to have to supervise Harold. One time that person was me.

The scene was in the 1980s when longevity with a company meant more than it does today. The team I managed was very keen. Our defining characteristic was that we wanted to become a better team and we worked toward this goal. We had well-defined and practiced processes, we developed and implemented requirements and design modeling approaches, we applied and refined a comprehensive review methodology that inspected requirements, designs, code, test plans, test cases, everything. We even invented an innovative new testing method. And when we became more effective, we worked hard at becoming even more effective. Then Harold joined the team.

Long before Harold came on board, the team had made a pact: we would allow no bad work, not from anyone, not the chief designer, not the test coordinator, not the manager. Not even Harold. We would provide all the processes, templates, support, resources, training, reviews, feedback, and discipline necessary for every team member to be 100% successful. We even defined the metrics that measured this success. When Harold joined us, we laid this system out before him and explained that following these processes was a condition of being a member of the team. This discipline applied to everyone, even Harold.

A Program Gets Redesigned

As his first task, Harold was given one of the easier programs to write. The package he received prior to coding contained good reviewed requirements, an excellent program design complete with graphics, a proven process for developing test conditions, and a schedule for the ultimate inspection of his code. While he was coding, other project issues arose and we couldn't keep to the review schedule. When Harold was finally able to bring his code for inspection he had already unit tested it. This was in conflict with the normal and expected process order, but it wasn't Harold's fault; and the program worked, at least according to Harold. However, in the code review we saw right away that the program he wrote did not follow the reviewed and approved design at all. Harold had not submitted (and defended) an alternative design to the team's systems designers; he had just written what he wanted to write, the way he wanted to write it. Accordingly, his code failed the inspection immediately. Harold was shocked: "…but it passed all the tests!…" he cried. His complaints reached all the way up to the director of the installation, protest-

Sometimes it seems the software process is implemented with the intention of squeezing out any trace of individuality and creativity.

ing: "...they're making me rewrite something that already works!..." Fortunately, his griping was ignored, the director backed the team, and Harold had to rewrite the program to the original design. When the code finally came back for inspection, it matched the design, worked well, and passed inspection easily with relatively few comments and issues.

A Better Team, A Better System

As a team, we noticed an interesting thing: *because* Harold was a low-achieving programmer, we had become a better team: we had defined and enforced our process more tightly, we had set up and captured metrics more consistently, we had learned to provide better counseling and support to the programmers (especially Harold). Most importantly, we had adopted a team goal and ethic: we would never accept bad work and never simply discard a weaker member of the team who could not perform—we would do what was necessary to allow the person to meet the team's standards. Provided team members supported these goals, availed themselves of the resources, and produced viable products, they could be a member of our team and the team would take care of them professionally. This higher ethic made us a better team.

Modern physics is learning that the behavior of systems may not be a direct function of the behavior of the parts of a system. Not all cogs in a watch have to be big cogs. Sometimes more optimal behavior in a system can be obtained from less optimal parts. Sometimes the best teams are not made up of all superheroes and when a team asserts itself to overcome some limitation, even with its own personnel, it can become better than it would have been without that limitation. This is not to say we should intentionally seek to employ bad programmers; but teams are not disconnected parts like nuts and bolts in a bucket—they are systems. The dynamics of an effective team are more complex and subtle than they sometimes appear and low-performing people may have an effect on a project other than through the products they produce.[a]

Sometimes it seems the software process is implemented with the intention of squeezing out any trace of individuality and creativity—to make building systems a mindless process. In this team, we were careful not to do this; but we were also careful not to simply allow anyone to change anything, anytime, without consideration of the value and return. We certainly never allowed anyone to make changes simply out of laziness, misunderstanding, or incompetence. And we never let anyone fail.

Harold Redeemed

When Harold had his code approved, he (to the surprise of everyone on the team) remarked "You know, this design is much better than the one I did." In fact, from that point onward, Harold became a believer in our approach and (to the surprise of everyone in the installation) a vocal evangelist for our team and our processes. He even made a significant contribution to the team's processes and tools—with the appropriate team reviews and approvals, of course.

The team had defined the highest level of inspection acceptance as an "AS-IS," meaning no errors of any sort, no minor functional issues, no design questions, no documentation inaccuracies, not even spelling or grammar mistakes. AS-IS was perfection. In nearly 500 inspections we conducted during the project life cycle, only one person on the team ever got an AS-IS on any work product submitted for inspection. That was Harold. C

Phillip G. Armour (armour@corvusintl.com) is a senior consultant at Corvus International Inc., Deer Park, IL.

[a] See J.B. Harvey, *How Come Every Time I Get Stabbed in the Back, My Fingerprints are on the Knife? And Other Meditations on Management*, Jossey-Bass, 1999.

DOI:10.1145/1629175.1629191

Arti Rai

Law and Technology

Unstandard Standardization: The Case of Biology

How applicable are the approaches adopted by information and communication technology standards-setting organizations to biological standards?

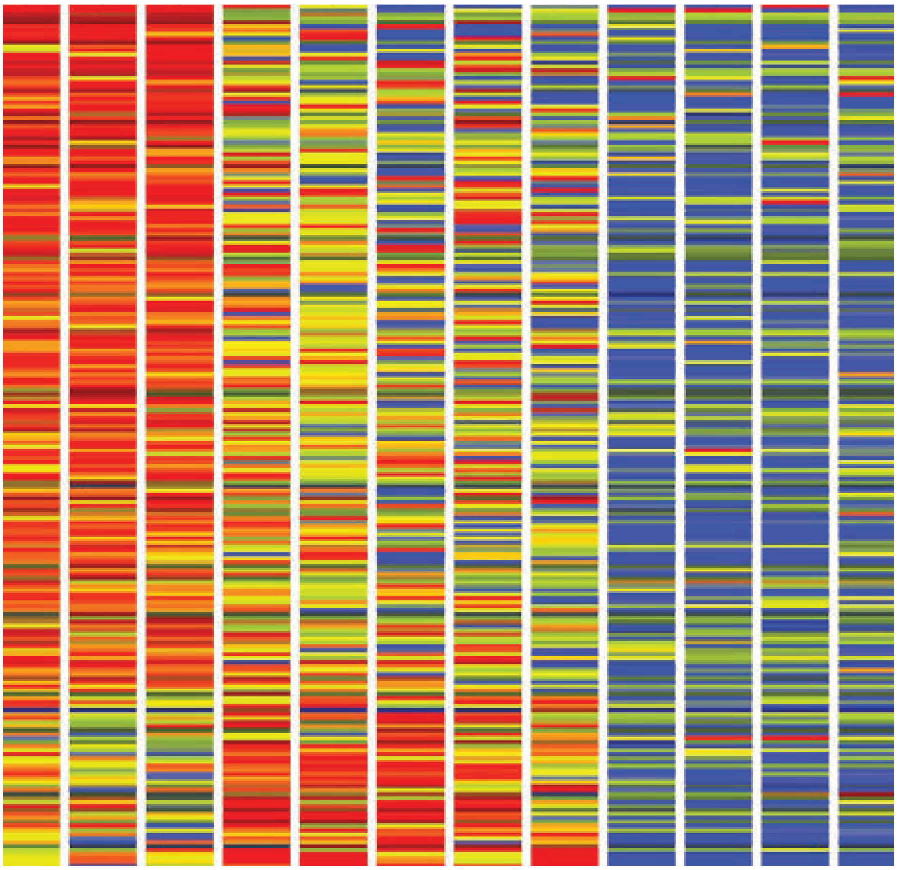
MOST ENGINEERING-BASED industries construct products from standard, well-understood components. By contrast, despite the early attachment of the moniker "genetic engineering" to biotechnology, standardization in the biological sciences has been relatively rare. In 2004, MIT computer scientist Tom Knight offered this colorful characterization of the difference between a biologist and an engineer: "A biologist goes into the lab, studies a system, and finds that it is far more complex than anyone suspected. He's delighted; he can spend a lot of time exploring that complexity and writing papers. An engineer goes into the lab and makes the same finding. His response is 'How can I get rid of this?'"[2]

Knight's insightful observation notwithstanding, efforts are currently being made to standardize biology. What lessons (if any) can biology learn from engineering?

Standard-Setting Organizations

The area of engineering where standard setting has been most discussed is information and communication technology (ICT). In the ICT industries, standards often have the potential to read on dozens if not hundreds of patents. Thus standard-setting organizations (SSOs) that make choices among potential standards generally have policies concerning patent disclosure and licensing. The most elaborate policies require disclosure of patents not only by those entities that actually submit technology to the standard but also by other members of the standards organization. As for licensing, patent owners may be required to license royalty-free or, more frequently, on "reasonable and nondiscriminatory terms." At least in theory, such deliberate decision making should lead to the adoption of standards that balance payment of patent licensing royalties with technological superiority.

Through rigorous disclosure and licensing policies, SSOs also hope to avoid future lawsuits in which previously unknown patent owners make assertions of infringement against



A detail of hierarchical clustering and ANOVA of single-cell gene expression data.

IMAGE SOURCE: MARK D. LEVIN , J. CLIN. INVEST. 119:11 DOI:10.1172/JCI39109

product manufacturers that use a widely adopted standard. In such circumstances, the patent holder could arguably extract a royalty in excess of the technical contribution made by the patent.

How applicable are the approaches adopted by SSOs in the ICT industries to biological standards? To a significant extent, the answer depends on the type of standard.

Currently, some of the most advanced standardization efforts involve specifications for the development and presentation of biological data. The Microarray Gene Expression Data Society (MGED) was an early leader in the field. MGED's "Minimum Information About a Microarray Experiment" (MIAME) standard has inspired similar efforts in many other biological fields, including proteomics, metabolomics, and RNA interference.[4] The Minimum Information for Biological and Biomedical Investigations (MIBBI) project takes standardization one step further by attempting to rationalize the varying data standards that have developed in different biological fields. MIBBI's goal is interoperability across data sets from different biological communities.[5]

These data standardization efforts, which emerged from academic institutions, do not appear to have adopted formal policies on patents. But in the case of data standards, the administrative costs associated with establishing an SSO-type apparatus may exceed any challenges that patents pose. At least at this stage, the numbers of patents that could be asserted may not be particularly large.

Biomarker Standards

Another important category of biological standardization efforts involves biomarkers. Biomarkers are biological signs of drug toxicity and efficacy, and the pharmaceutical industry has high hopes that improved biomarkers will yield expedited preclinical drug safety evaluation as well as early indicators of clinical safety and efficacy. With such indicators, firms should be able to reduce the costly drug trial failures that are currently a major contributing factor to diminished biopharmaceutical innovation.

Pharmaceutical companies have formed a number of consortia that pool information and conduct collaborative research to identify consensus biomarker standards. Prominent consortia include the Predictive Safety Testing Consortium (PSTC), which comprises 17 major multinational pharmaceutical firms. The PSTC has already been successful in securing U.S. Food and Drug Administration and European Medicines Evaluation Agency approval for seven new biomarkers that signal kidney injury at the preclinical stage.

The various biomarker standards consortia set up by pharmaceutical firms deal very explicitly with patent rights. To some extent, these consortia adopt policies similar to those of SSOs in the ICT industries. For example, although the PSTC policy does not address the licensing of "background" patents that firms may bring to the collaborative research, it addresses with great care future patents on biomarker standards that may emerge. Specifically, PSTC members assign any future patent rights to a non-profit trusted intermediary, Critical Path. Critical Path, in turn, is obliged to license the rights on "fair, neutral, and commercially reasonable" terms to members of the Consortium as well as third parties. Described another way, in the PSTC, *future* patents are addressed in terms somewhat similar to those used by ICT SSOs for background patents.

The emerging discipline of synthetic biology aims for what is arguably the most comprehensive form of standardization. It hopes to make all of biotechnology a science that relies on standardized, well-characterized DNA "parts." These parts could then be assembled into composite devices and systems with similarly well-defined behavior. When transplanted into suitable model organism "chassis" (which had themselves been standardized), the composite systems could yield outputs ranging from drug therapies to environmentally friendly fuels. Standards would cover not only parts and chassis but, perhaps even more importantly, the interfaces used to assemble parts and the interactions between parts and host cells.

Standardization in Synthetic Biology

The synthetic biology community is still debating precisely how much information about a biological standard is necessary before full standardization can be said to have been achieved.[1] Even so, some progress has been made. The Registry of Standard Biological Parts (http://www.partsregistry.org), an academic effort that receives significant federal funding, now contains about 3,200 parts. Each of these parts adheres to the so-called BioBricks protocol for cloning and physical linking and has specific as-

Some of the most advanced standardization efforts involve specifications for the development and presentation of biological data.

sociated inputs and outputs.

The Registry of Standard Biological Parts presents what may be the most interesting, and difficult, challenge for patents on biological standards. As currently constituted, the Registry may well read on a large number of patents. Tens of thousands of U.S. patents have been granted on DNA sequences. Although these patents are not specific to synthetic biology, they could certainly read on various standardized parts. Preliminary patent mapping also reveals a significant number of patents highly relevant to synthetic biology in particular.[3]

Thus far the Registry essentially puts results in the public domain, albeit with a hortatory suggestion that users should contribute back information and data, so as to improve the "community resource." As for background patents that the Registry may infringe, the academic scientists involved appear to be proceeding under the assumption that they will be not be sued because potential plaintiffs will not foresee significant monetary payoffs from such suits. As for potential industry defendants, at this stage it does not appear that Registry parts are being used by industry to make commercially valuable products.

At some point, however, Registry parts may begin to be used by industry. In addition, use of such standardized parts may be difficult to conceal. Thus one apparently common biopharmaceutical industry strategy for avoiding patents on research inputs—secret infringement—may not be possible.[6] Industry users that are contemplating using Registry parts might therefore consider organizing patent mapping efforts to determine whether patents do in fact read on key standards.

The situation the Registry faces arguably bears some similarity to that faced by standards developers for the Web in its early days. For example, in the case of the XML standard for structured data presentation, the critical early work was done by developers from academic and commercial organizations, as well as independent contributors, without any significant thought being given to patents.

As the Web matured, however, the issue of background patents on core technical standards had to be addressed. By 1999, the World Wide Web Consortium had created a patent policy working group. Participants in that group included representatives from the major software, hardware, and telecommunications firms (Apple, AT&T, Hewlett-Packard, IBM, Intel, Motorola, Nokia, Nortel, Sun Microsystems, and Xerox).

The emerging discipline of synthetic biology aims for what is arguably the most comprehensive form of standardization.

Conclusion

At this stage in the evolution of synthetic biology, it is probably too early to determine whether any of the work done thus far has yielded key standards upon which the community will eventually converge. But as synthetic biologists and other biologists continue work on standardization, they should carefully examine mechanisms (both successful and unsuccessful) for addressing patent issues that have been invoked in the ICT industries. C

References

1. Arkin, A. Setting the standard in synthetic biology. *Nature Biotechnology 26* (2008), 771–774.
2. Brown, C. BioBricks to help reverse-engineer life. *EE Times* (Nov. 6, 2004).
3. Kumar, S. and Rai, A. Synthetic biology: The intellectual property puzzle. *Texas Law Review 85* (2007), 1745–1768.
4. Standard operating procedures. *Nature Biotechnology 24* (2006), 1299.
5. Taylor et al. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: The MIBBI project. *Nature Biotechnology 26* (2008), 889–896.
6. Walsh, J., Arora, A., and Cohen, W. Working through the patent problem. *Science 299* (2003), 1021.

Arti Rai (RAI@law.duke.edu) is Elvin R. Latty Professor of Law at Duke University and the chair of the Intellectual Property Committee of the Administrative Law Section of the American Bar Association.

This column was written before the author entered U.S. government service and does not represent the views of the U.S. government.

Calendar of Events

January 17–23
POPL '10: The 37th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Lanuguages
Madrid, Spain,
Sponsored: SIGPLAN,
Contact: Manuel V. Hermenegildo,
Phone: 34-91-336-7435,
Email: herme@fi.upm.es

January 18–19
PEPM '10: Partial Evaluation and Program Manipulation (co-located with POPL 2010)
Madrid, Spain,
Sponsored: SIGPLAN,
Contact: John P. Gallagher,
Phone: 45 46742196,
Email: jpg@ruc.dk

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

January 22–24
International Conference on Agents and Artificial Intelligence
Valencia, Spain,
Contact: Joaquim B. Filipe,
Phone: 351-91-983-3996,
Email: jfilipe@insticc.org

January 25–27
Fourth International Conference on Tangible, Embedded, and Embodied Interaction
Cambridge, MA,
Sponsored: SIGCHI,
Contact: Jamie Zigelbaum,
Phone: 617-877-4970,
Email: zig@media.mit.edu

January 27–29
Multimedia Systems Conference
Phoenix, AZ,
Sponsored: SIGMM,
Contact: Wu-Chi Feng,
Phone: 503-725-2408,
Email: wuchi@cs.pdx.eduv

DOI:10.1145/1629175.1629192

Bjarne Stroustrup

Viewpoint
What Should We Teach New Software Developers? Why?

Fundamental changes to computer science education are required to better address the needs of industry.

COMPUTER SCIENCE MUST be at the center of software systems development. If it is not, we must rely on individual experience and rules of thumb, ending up with less capable, less reliable systems, developed and maintained at unnecessarily high cost. We need changes in education to allow for improvements of industrial practice.

The Problem

In many places, there is a disconnect between computer science education and what industry needs. Consider the following exchange:

Famous CS professor (proudly): *"We don't teach programming; we teach computer science."*

Industrial manager: *"They can't program their way out of a paper bag."*

In many cases, they are both right, and not just at a superficial level. It is not the job of academia just to teach run-of-the-mill programmers and the needs of industry are not just for "well-rounded high-level thinkers" and "scientists."

Another CS professor: *"I never code."*

Another industrial manager: *"We don't hire CS graduates; it's easier to teach a physicist to program than to teach a CS graduate physics."*

Both have a point, but in an ideal world, both would be fundamentally misguided. The professor is wrong in that you can't teach what you don't practice (and in many cases, never have practiced) and therefore don't understand, whereas the industrial manager is right only when the requirements for software quality are set so absurdly low that physicists (and others untrained in CS) can cope. Obviously, I'm not referring to physicists who have devoted significant effort to also master computer science—such combinations of skills are among my ideals.

CS professor (about a student): *"He accepted a job in industry."*

Another CS professor: *"Sad; he showed so much promise."*

This disconnect is at the root of many problems and complicates attempts to remedy them.

Industry wants computer science graduates to build software (at least initially in their careers). That software is often part of a long-lived code base and used for embedded or distributed systems with high reliability requirements. However, many graduates have essentially no education or training in software development outside their hobbyist activities. In particular, most see programming as a minimal effort to complete homework and rarely take a broader view that includes systematic testing, maintenance, documentation, and the use of their code by others. Also, many students fail to connect what they learn in one class to what they learn in another. Thus, we often see students with high grades in algorithms, data structures, and software engineering who nevertheless hack solutions in an operating systems class with total disregard for data structures, algorithms, and the structure of the software. The result is a poorly performing unmaintainable mess.

For many, "programming" has become a strange combination of unprincipled hacking and invoking other people's libraries (with only the vaguest idea of what's going on). The notions of "maintenance" and "code quality" are typically forgotten or poorly understood. In industry, complaints about the difficulty of finding graduates who understand "systems" and "can architect software" are common and reflect reality.

But My Computer Hasn't Crashed Lately

Complaining about software is a popular pastime, but much software has become better over the last decade, exactly as it improved in previous decades. Unfortunately, the improvements have come at tremendous cost in terms of human effort and computer resources. Basically, we have learned how to build reasonably reliable systems out of unreliable parts by adding endless layers of runtime checks and massive testing. The structure of the code itself has sometimes changed, but not always for the better. Often, the many layers of software and the intricate dependencies common in designs prevent an individual—however competent—from fully understanding a system. This bodes ill for the future: we do not understand and cannot even measure critical aspects of our systems.

There are of course system builders who have resisted the pressures to build bloated, ill-understood systems. We can thank them when our computerized planes don't crash, our phones work, and our mail arrives on time. They deserve praise for their efforts to make software development a mature and trustworthy set of principles, tools, and techniques. Unfortunately, they are a minority and bloatware dominates most people's impression and thinking.

Similarly, there are educators who have fought the separation of theory and industrial practice. They too deserve praise and active support. In fact, every educational institution I know of has programs aimed at providing practical experience and some professors have devoted their lives to making successes of particular programs. However, looking at the larger picture, I'm unimpressed—a couple of projects or internships are a good start, but not a substitute for a comprehensive approach to a balanced curriculum. Preferring the labels "software engineering" or "IT" over "CS" may indicate differences in perspective, but problems have a nasty way of reemerging in slightly different guises after a move to a new setting.

My characterizations of "industry" and "academia" border on caricature, but I'm confident that anyone with a bit of experience will recognize parts of reality reflected in them. My perspective is that of an industrial researcher and manager (24 years at AT&T Bell Labs, seven of those as department head) who has now spent six years in academia (in a CS department of an engineering school). I travel a lot, having serious discussions with technical and managerial people from several dozen (mostly U.S.) companies every year. I see the mismatch between what universities produce and what industry needs as a threat to both the viability of CS and to the computing industry.

Many organizations that rely critically on computing have become dangerously low on technical skills.

The Academia/Industry Gap

So what can we do? Industry would prefer to hire "developers" fully trained in the latest tools and techniques whereas academia's greatest ambition is to produce more and better professors. To make progress, these ideals must become better aligned. Graduates going to industry must have a good grasp of software development and industry must develop much better mechanisms for absorbing new ideas, tools, and techniques. Inserting a good developer into a culture designed to prevent semi-skilled programmers from doing harm is pointless because the new developer will be constrained from doing anything significantly new and better.

Let me point to the issue of scale. Many industrial systems consist of millions of lines of code, whereas a student can graduate with honors from top CS programs without ever writing a program larger than 1,000 lines. All major industrial projects involve several people whereas many CS programs value individual work to the point of discouraging teamwork. Realizing this, many organizations focus on simplifying tools, techniques, languages, and operating procedures to minimize the reliance on developer skills. This is wasteful of human talent and effort because it reduces everybody to the lowest common denominator.

Industry wants to rely on tried-and-true tools and techniques, but is also addicted to dreams of "silver bullets," "transformative breakthroughs," "killer apps," and so forth. They want to be able to operate with minimally skilled and interchangeable developers guided by a few "visionaries" too grand to be bothered by details of code quality. This leads to immense conservatism in the choice of basic tools (such as programming languages and operating systems) and a desire for monocultures (to minimize training and deployment costs). In turn, this leads to the development of huge proprietary and mutually incompatible infrastructures: Something beyond the basic tools is needed to enable developers to produce applications and platform purveyors want something to lock in developers despite the commonality of basic tools. Reward systems favor both grandiose corporate schemes and short-term results. The resulting costs are staggering, as are the failure rates for new projects.

Faced with that industrial reality—and other similar deterrents—academia turns in on itself, doing what it does best: carefully studying phenomena that can be dealt with in isolation by a small group of like-minded people, building solid theoretical foundations, and crafting perfect designs and techniques for idealized problems. Proprietary tools for dealing with huge code bases written in archaic styles don't fit this model. Like industry, academia develops reward structures to match. This all fits perfectly with a steady improvement of smokestack courses in well-delineated academic subjects. Thus, academic successes fit industrial needs like a square peg in a round hole and industry has to carry training costs as well as development costs for specialized infrastructures.

Someone always suggests "if industry just paid developers a decent salary, there would be no problem." That might help, but paying more for the same kind of work is not going to help much; for a viable alternative, industry needs better developers. The idea of software development as an assembly line manned by semi-skilled interchangeable workers is fundamentally flawed and wasteful. It pushes the most competent people out of the field and discourages students from entering it. To break this vicious circle, academia must produce more graduates with relevant skills and industry must adopt tools, techniques, and processes to utilize those skills.

Dreams of Professionalism

"Computer science" is a horrible and misleading term. CS is not primarily about computers and it is not primarily a science. Rather it is about uses of computers and about ways of working and thinking that involves computation ("algorithmic and quantitative thinking"). It combines aspects of science, math, and engineering, often using computers. For almost all people in CS, it is an applied field—"pure CS," isolated from application, is typically sterile.

What distinguishes a CS person building an application from a professional in some other field (such as medicine or physics) building one? The answer must "be mastery of the core of CS." What should that "core" be? It would contain much of the established CS curriculum—algorithms, data structures, machine architecture, programming (principled), some math (primarily to teach proof-based and quantitative reasoning), and systems (such as operating systems and databases). To integrate that knowledge and to get an idea of how to handle larger problems, every student must complete several group projects (you could call that basic software engineering). It is essential that there is a balance between the theoretical and the practical—CS is not just principles and theorems, and it is not just hacking code.

This core is obviously far more "computer-oriented" than the computing field as a whole. Therefore, nobody should be called a computer scientist without adding a specialization within CS (for example, graphics, networking, software architecture, human-machine interactions, security). However that's still not enough. The practice of computer science is inherently applied and interdisciplinary, so every CS professional should have the equivalent to a minor in some other field (for example, physics, medical engineering, history, accountancy, French literature).

Experienced educators will observe: "But this is impossible! Hardly any students could master all that in four years." Those educators are right: something has to give. My suggestion is that the first degree qualifying to practice as a computers scientist should be a master's—and a master's designed as a whole—not as a bachelor's degree with an appended final year or two. People who plan to do research will as usual aim for a Ph.D.

Many professors will object: "I don't have the time to program!" However, I think that professors who teach students who want to become software professionals will have to make time and their institutions must find ways to reward them for programming. The ultimate goal of CS is to help produce better systems. Would you trust someone who had not seen a patient for years to teach surgery? What would you think of a piano teacher who never touched the keyboard? A CS education must bring a student beyond the necessary book learning to a mastery of its application in complete systems and an appreciation of aesthetics in code.

I use the word "professional." That's a word with many meanings and implications. In fields like medicine and engineering, it implies licensing. Licensing is a very tricky and emotional topic. However, our civilization depends on software. Is it reasonable that essentially anyone can modify a critical body of code based on personal taste and corporate policies? If so, will it still be reasonable in 50 years? Is it reasonable that pieces of software on which millions of people depend come without warranties? The real problem is that professionalism enforced through licensing depends on having a large shared body of knowledge, tools, and techniques. A licensed engineer can certify that a building has been constructed using accepted techniques and materials. In the absence of a widely accepted outline of CS competence (as I suggested earlier), I don't know how to do that for a software application. Today, I wouldn't even know how to select a group of people to design a licensing test (or realistically a set of tests for various subspecialties, like the medical boards).

What can industry do to close the gap? It is much more difficult to characterize "industry" and "industrial needs" than to talk about academia. After all, academia has a fairly standard structure and fairly standard approaches to achieving its goals. Industry is far more diverse: large or small, commercial or non-profit, sophisticated or average in their approach to system building, and so forth. Consequently, I can't even begin to prescribe remedies. However, I have one observation that related directly to the academia/industry gap: Many organizations that rely critically on computing have become dangerously low on technical skills:

Industrial manager: *"The in-sourcing of technical expertise is critical for survival."*

No organization can remain successful without an institutional memory and an infrastructure to recruit and develop new talent. Increasing collaboration with academics interested in software development might be productive for both parties. Collaborative research and an emphasis on lifelong learning that goes beyond mere training courses could play major roles in this.

Conclusion

We must do better. Until we do, our infrastructure will continue to creak, bloat, and soak up resources. Eventually, some part will break in some unpredictable and disastrous way (think of Internet routing, online banking, electronic voting, and control of the electric power grid). In particular, we must shrink the academia/industry gap by making changes on both sides. My suggestion is to define a structure of CS education based on a core plus specializations and application areas, aiming eventually at licensing of software artifacts and at least some of the CS professionals who produce them. This might go hand-in-hand with an emphasis on lifelong industry/academia involvement for technical experts. C

Bjarne Stroustrup (bs@cs.tamu.edu) is the College of Engineering Chair in Computer Science Professor at Texas A&M University in College Station, TX.

DOI:10.1145/1629175.1629193

William Aspray

Computer Museum Series
Great Computing Museums of the World, Part One

The first of a two-part series highlighting several of the world's museums dedicated to preserving, exhibiting, and elucidating computing history.

Some of the science and technology museums around the world are devoted to science discovery—to teaching their visitors, especially children, about the principles of science and technology. Other science and technology museums are more focused on the history and cultural significance of particular scientific discoveries and technological inventions. Some museums include a blend of the two functions.

This is the first installment of a two-part *Communications* series featuring five of world's greatest computing museums. These museums have been chosen for their contributions to the history and culture mission, though most of them have some elements of the science discovery mission as well. There are perhaps hundreds of small and not-so-small museums around the world either devoted entirely to computing or at least having significant computing exhibits. The five museums highlighted in this series have been selected because of the large size of their exhibits, the importance and quality of the artifacts shown, and the quality of their interpretations.

An exhibit is not simply a collection of artifacts; it includes signage and other accompanying information (films, lectures, guided tours) that help to interpret the artifacts and set them in context. Each of the exhibits described in this series is the result of years of human labor in preparation: designing the exhibit, selecting and securing exactly the right artifacts, and giving them the right interpretation. This work has been carried out by some of the best historians of science and technology, who work in these museums collecting artifacts and the associated information and documentation about them, answering queries from all kinds of people about their collections and about the science and its history, undertaking scholarly research, preparing educational materials, and doing much more. The exhibits are only one facet of what these museums do.

The museums featured in this issue are the Computer History Museum, located in Mountain View, CA, and the Heinz Nixdorf Forum in Paderborn, Germany. We hope you enjoy the accounts of these museums and that these stories will whet your appetite to explore the museums' Web sites and to visit the museums in person.

IBM International Computer Chess Championship
1996 SOFTWARE TOOLKIT BASICS

The Computer History Museum exhibit "Mastering the Game: A History of Computer Chess."

PHOTOGRAPH COURTESY OF THE COMPUTER HISTORY MUSEUM

William Aspray (bill@ischool.utexas.edu) is Bill and Lewis Suit Professor of Information Technologies at the University of Texas, Austin and a *Communications* Viewpoints section board member.

The Computer History Museum
Len Shustek

For most of the 10,000 years of recorded history, there were no computers. We are privileged to be living through the brief transitional period: from now on, and forever more, computers will be an intimate and inseparable part of our life and work. The engines of the 19th century industrial revolution were amplifiers for our physical bodies. The computers of the 20th century information revolution are amplifiers for our minds.

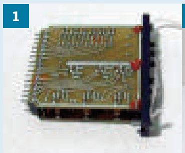
Viewed from 1,000 years from now, the 50 years that elapsed from the invention of the computer to its ubiquitous use will seem like a point in time. We owe it to ourselves as current participants, and to future generations as our beneficiaries, to document and explain how the information revolution came to be.

This perspective motivates the mission of the Computer History Museum (CHM; http://www.computerhistory.org): "To preserve and present for posterity the artifacts and stories of the information age." Therefore, the Computer History Museum is an evolving institution with three primary initiatives:

Collecting. At the core of CHM is the computing collection, which was started 30 years ago in Boston, MA, by Gordon and Gwen Bell within Digital Equipment Corporation. It then became part of The Computer Museum, a public museum in Boston, and when that institution closed in 1999 the collection became part of CHM. This ever-growing repository, whose catalog is online, now has about 70,000 objects in six categories:

- ▸ Physical artifacts: from microscopic chips to room-sized mainframes;
- ▸ Software: source code, executable code, and documentation for systems and applications, both in original formats and converted to modern digital formats;
- ▸ Documents: 30 million pages of primary reference material useful for the technical, business, and social history of computing, much of which is unpublished or near-print;
- ▸ Photographs: tens of thousands of prints, negatives, and digital images of items, locations, and people related to the history of computing
- ▸ Moving images: films and videos stored on many kinds of media, most of which have been converted to digital format; and
- ▸ Oral histories: interviews of computing pioneers, most done using

Highlights of the Computer History Museum collection



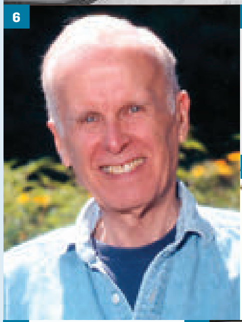
1. ***CDC 6600 transfer board, serial number 1:*** **The CDC 6600 was a Control Data Corporation mainframe computer designed by legendary computer architect Seymour Cray. It is considered the first successful supercomputer, and was the world's fastest computer from 1964 to 1969.**

2. ***Busicom calculator prototype:*** **The first device to use the first microprocessor, the Intel 4004 from 1971.**

3. ***Altair BASIC paper tape:*** **An original tape of the BASIC language interpreter written by Bill Gates for the Altair 8800 computer, and signed by him.**

4. ***Apollo Guidance Computer:*** **The computer which, with less computing power than a typical digital watch, guided the Apollo lunar module through its descent to the moon's surface in 1969.**

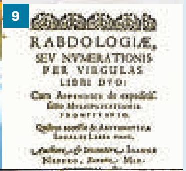
5. ***SAGE:*** **A huge and amazingly reliable air defense computer built in the 1950s out of 51,000 vacuum tubes and located in an underground concrete bunker.**



6. ***John Backus interview:*** **Videotape and transcript of a long interview with Fortran pioneer John Backus, made the year before he died.**

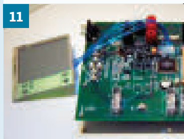
7. ***Apple Lisa ephemera:*** **Button, hat, t-shirt, and poster for the 1983 release of the graphics-oriented Lisa computer.**

8. ***Johnniac:*** **Built in 1954 and named for John von Neumann, this was one of 17 custom-built machines inspired by his design, and is the only complete one that has survived.**


9
RABDOLOGIÆ,
SEV NVMERATIONIS
PER VIRGVLAS
LIBRI DVO:

9. ***Rabdologia:*** **An original copy of Napier's 1617 book on calculating methods, including a description of his eponymous "bones."**

10. ***IBM card sorter:*** **A model 080 punched card sorter from 1925. Although over 10,000 were made, few have survived.**



11. ***Palm Pilot prototype:*** **the engineering model of the first highly successful personal digital assistant.**

To search the online catalog, see http://www.computerhistory.org/collections/search/

To view historic videos and recent lecture events, see http://www.youtube.com/computerhistory

high-quality video and subsequently transcribed and edited. It is critical that we collect the first-person stories of our pioneers while we still can.

Educational Activities. Expanding the public presence of the museum is the current highest priority. This includes:

- Exhibits: There are about a dozen physical and online exhibits currently on display, such as "The Silicon Engine," "Mastering the Game: A History of Computer Chess," "The Babbage Engine," and "Visible Storage". The physical exhibits are open to the public four days a week. A major 25,000 square foot signature exhibition on computing history is scheduled to open later this year, in conjunction with a comprehensive Web-based version that will also provide digitized access to related objects from the deep collection.
- Programs: A public lecture series that attracts over 300 attendees is held each month and is available afterward on the Web. A black-tie Fellows Awards ceremony to honor outstanding computing pioneers is held yearly. An annual award-winning magazine (*CORE*) is published as well as commemorative booklets that highlight important computing milestones.
- Restorations: Historical computer systems, both hardware and software, are selectively restored and demonstrated. A restoration of an IBM 1620 and a DEC PDP-1 have been completed; restorations currently under way include an IBM 1401 system (complete with card equipment, printer, and tape drives), and the world's first disk drive, the RAMAC 350. The restored systems are on exhibit and are demonstrated by trained volunteer docents.

We owe it to ourselves as current participants, and to future generations as our beneficiaries, to document and explain how the information revolution came to be.

Research Activities. The CHM wishes to become an important part of the academic research community on computing history, but it has only taken small steps so far: organizing topical conferences and workshops, collecting oral histories, and publishing papers and articles.

The CHM scope (and collection) is international, but the museum's physical presence is in the heart of Silicon Valley in California. The CHM owns a 120,000 square foot modern building on seven acres—lots of free parking is a real asset here!—in a prominent location in Mountain View. The CHM also owns a 25,000 square foot warehouse 20 minutes away, where most of the 90% of the collection that is not on exhibit at any particular time is stored in climate-controlled conditions and is available to researchers.

The Computer History Museum is a work in progress. We like to think of ourselves as a startup with a 30-year history. We welcome the opportunity to work with people and organizations that resonate with our mission and our goals. For more information, see www.computerhistory.org.

Len Shustek (shustek@computerhistory.org) is the chairman of the Computer History Museum.

The Heinz Nixdorf MuseumsForum

Norbert Ryksa

The Heinz Nixdorf MuseumsForum (HNF; www.hnf.de) in Paderborn, Germany, is the world's largest computer museum. The museum, which is also an established conference center, showcases the history of information technology—beginning with cuneiform writing and going right through to the latest developments in robotics, artificial intelligence, and ubiquitous computing.

The multimedia journey through time takes visitors through 5,000 years of history, starting with the origins of numbers and writing in Mesopotamia in 3000 B.C. and covering the entire cultural history of writing, calculating, and communications. Alongside typewriters and calculating machines, the exhibition shows punched card systems, a fully functioning automatic telephone exchange system from the 1950s, components from the earliest computer (which filled a whole room), over 700 pocket calculators, and the first PCs. Work environments from different centuries are also staged in the exhibition.

The exhibition highlights include fully functioning replicas of the Leibniz calculating machine and the Hollerith tabulating machine, a Thomas Arithmometer dating from 1850, a Jacquard loom operated with punched tape, components of the ENIAC from

The Heinz Nixdorf MuseumsForum in Paderborn, Germany.

PHOTOGRAPH COURTESY OF THE HEINZ NIXDORF MUSEUM

The Chess Turk, the world's most famous automaton, at the Heinz Nixdorf MuseumsForum.

PHOTOGRAPH COURTESY OF THE HEINZ NIXDORF MUSEUM

1945, the on-board computer from the Gemini space capsule, the Apple 1, a LEGO Turing machine, and Europe's largest collection of cipher machines. One of the current attractions at HNF is the world's most famous automaton: Wolfgang von Kempelen's chess playing machine, the Chess Turk, which dates from the 18th century.

The exhibition was updated in 2004 with the addition of new themes such as robotics and artificial intelligence, mobile communications, and digitization. The new galleries present the latest information technology themes in an interactive, multimedia exhibit. Visitors can try their skills at old and new computer games, test advanced man-machine interfaces, and experiment with the latest applications and products from research and industry in the showroom. A multimedia scenario presents 150 pioneers of computer history from 1940–2009. Along with conventional museum formats, HNF has chosen to use a range of interactive multimedia applications and videos: approximately 100 special interactive multimedia application developments and video installations introduce visitors to the functions of the objects that are on display as well as to the life stories of famous historical figures.

Themes relating to the present and the future are also presented in HNF's Software Theatre, which offers virtual tours through cyberspace. Visitors can test the latest computer applications and software developments at the Digital Workbench. The games booths offer educational games and games of skill and strategy for guests to try out.

In 2005, the HNF marked the 40th anniversary of Moore's Law by presenting a huge illuminated "chip pagoda" demonstrating the continuous minimization of the chip surface area over the years in 20 stages. The individual levels of the pagoda consist of illuminated plexiglass panes and the display is lit by some 3,500 LEDs.

In 2007, the HNF opened the world's first gallery on software and computer science (informatics). A black cube is decorated on its "shell" with early instances of computer programs and 13 small "miracle chambers of computer science." These provide concrete three-dimensional examples of ostensibly abstract topics: Russian nested dolls are used to explain the method of recursion, while a tin of English luncheon meat demonstrates the origins of the term "spam," and toy robots illustrate an important software application area.

HNF has compiled a varied educational museum program to motivate children and young people to take an active approach to the exhibits and their history. At workshops children can, for example, build robots, encrypt messages or learn how to 'make' paper. Teachers and pupils are given numerous ideas for study content. Besides a guided tour of the permanent exhibition, special tours can be booked on such topics as arithmetic, writing, inventors and entrepreneurs, women's work in information technology, and cryptology.

Special HNF events focus on people in the information age. Numerous presentations, discussions, conferences and workshops deal with current concerns in today's information society and information technology. A quarterly newsletter on HNF activities is published in two languages and distributed free of charge.

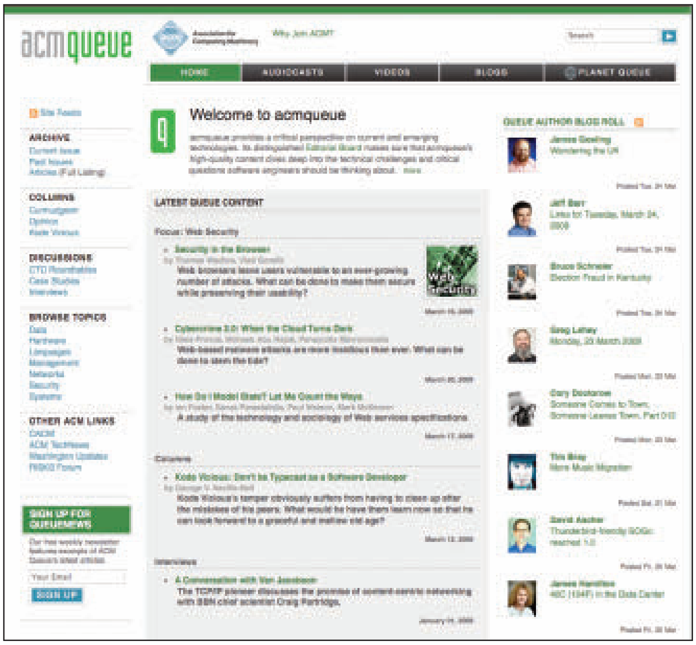
Alongside the permanent exhibition, the museum has two additional areas that cater to an ambitious special exhibition program, such as focusing on historical slot machines or the world of computer games.

The HNF's "Computer.Medicine" exhibition, designed to provide a broad overview of the computer's importance in present-day medicine and featuring many exhibits on loan from abroad, proved to be the most successful exhibition in the history of the HNF, attracting more than 93,000 visitors. The exhibition is currently being shown in Vienna (until mid-April) on the occasion of the 100th anniversary of the opening of the Austrian Technology Museum.

Until the end of February, the HNF in cooperation with the MIT Museum is presenting the exhibition "Codes and Clowns: Claude Shannon, The Juggling Scientist," which will showcase Claude Shannon's scientific work as well as his famous toy collection. The selection of Shannon's inventions range from the highly practical to the downright useless and the presentation sets Shannon's inventions in the context of his biography and the history of information technology, shedding light on the relevant scientific relationships and implications. The exhibits are on loan from the MIT Museum in Boston, MA, the first time they have been on display a different location.

The Forum part of the HNF organizes over 800 events every year, ranging from scientific congresses to popular lecture series, senior's IT workshops, and business fairs. C

Norbert Ryska (NRyska@hnf.de) is the managing director of the Heinz Nixdorf MuseumsForum.



Welcome to acmqueue

acmqueue provides a critical perspective on current and emerging technologies. Its distinguished Editorial Board makes sure that acmqueue's high-quality content dives deep into the technical challenges and critical questions software engineers should be thinking about. more

LATEST QUEUE CONTENT

Focus: Web Security

- Security in the Browser
  by Thomas Wadlow, Vlad Gorelik
  Web browsers leave users vulnerable to an ever-growing number of attacks. What can be done to make them secure while preserving their usability?
  March 16, 2009
- Cybercrime 2.0: When the Cloud Turns Dark
  by Niels Provos, Moheeb Abu Rajab, Panayiotis Mavrommatis
  Web-based malware attacks are more insidious than ever. What can be done to stem the tide?
  March 20, 2009
- How Do I Model State? Let Me Count the Ways
  by Ian Foster, Savas Parastatidis, Paul Watson, Mark McKeown
  A study of the technology and sociology of Web services specifications.
  March 17, 2009

Columns

- Kode Vicious: Don't be Typecast as a Software Developer
  by George V. Neville-Neil
  Kode Vicious's temper obviously suffers from having to clean up after the mistakes of his peers. What would he have them learn now so that he can look forward to a graceful and mellow old age?
  March 13, 2009

Interviews

- A Conversation with Van Jacobson
  The TCP/IP pioneer discusses the promise of content-centric networking with BBN chief scientist Craig Partridge.
  January 19, 2009

Our free weekly newsletter featuring excerpts of ACM Queue's latest articles

James Gosling
Wondering the UI
Posted Tue, 24 Mar

Jeff Barr
Links for Tuesday, March 24, 2009
Posted Tue, 24 Mar

Bruce Schneier
Election Fraud in Kentucky
Posted Tue, 24 Mar

Greg Lehey
Monday, 23 March 2009
Posted Mon, 23 Mar

Cory Doctorow
Someone Comes to Town, Someone Leaves Town, Part 010
Posted Mon, 23 Mar

Tim Bray
More Music Migration
Posted Sat, 21 Mar

David Ascher
Thunderbird-friendly BOGs: reached 1.0
Posted Fri, 20 Mar

James Hamilton
40C (104F) in the Data Center
Posted Fri, 20 Mar

DOI:10.1145/1629175.1629195

Article development led by acmqueue
queue.acm.org

How streaming SQL technology can help solve the Web 2.0 data crunch.

BY JULIAN HYDE

Data in Flight

WEB APPLICATIONS PRODUCE data at colossal rates, and those rates compound every year as the Web becomes more central to our lives. Other data sources such as environmental monitoring and location-based services are a rapidly expanding part of our day-to-day experience. Even as throughput is increasing, users and business owners expect to see their data with ever-decreasing latency. Advances in computer hardware (cheaper memory, cheaper disk, and more processing cores) are helping somewhat, but not enough to keep pace with the twin demands of rising throughput and decreasing latency.

The technologies for powering Web applications must be fairly straightforward for two reasons: first, because it must be possible to evolve a Web application rapidly and then to deploy it at scale with a minimum of hassle; second, because the people writing Web applications are generalists and are not prepared to learn the kind of complex, hard-to-tune technologies used by systems programmers.

The streaming query engine is a new technology that excels in processing rapidly flowing data and producing results with low latency. It arose out of the database research community and therefore shares some of the characteristics that make relational databases popular, but it is most definitely not a database. In a database, the data arrives first and is stored on disk; then users apply queries to the stored data. In a streaming query engine, the queries arrive before the data. The data flows through a number of continuously executing queries, and the transformed data flows out to applications. One might say that a relational database processes data at rest, whereas a streaming query engine processes data in flight.

Tables are the key primitive in a relational database. A table is populated with records, each of which has the same record type, defined by a number of named, strongly typed columns. Records have no inherent ordering. Queries, generally expressed in SQL, retrieve records from one or more tables, transforming them using a small set of powerful relational operators.

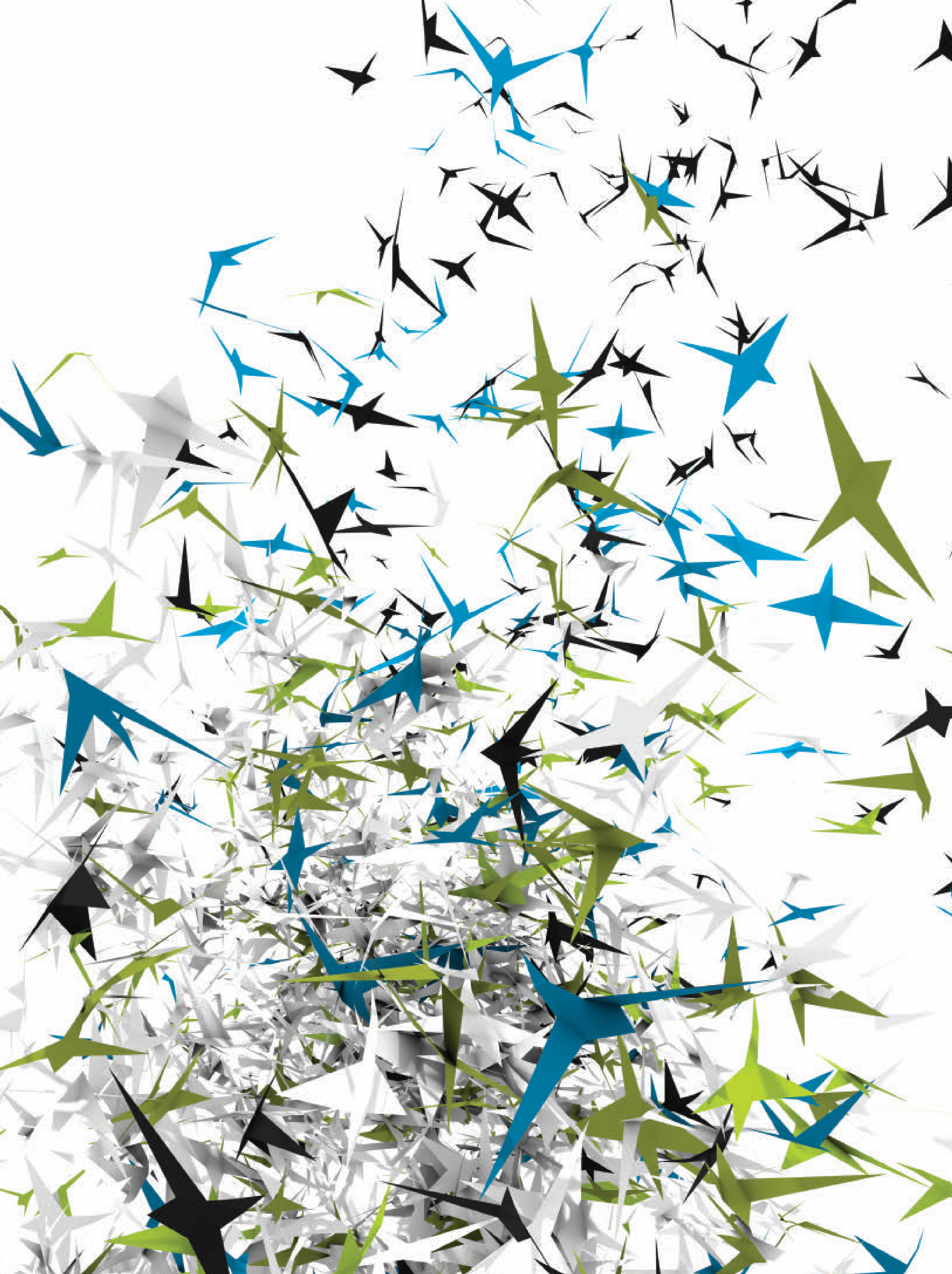
Streams are the corresponding primitive in a streaming query engine. A stream has a record type, just like a table, but records flow through a stream rather than being stored. Records in a streaming system are inherently ordered; in fact, each record has a time stamp that indicates when it was created. The relational operations supported by a relational database have analogues in a streaming system and are sufficiently similar that SQL can be used to write streaming queries.

To illustrate how a streaming query engine can solve problems involving data in flight, consider the following example.

Streaming Queries for Click-Stream Processing

Suppose we want to monitor the most popular pages on a Web site. Each Web server request generates a line to the

ILLUSTRATION BY RYAN ALEXANDER



Web server's log file describing the time, the URI of the page, and the IP address of the requester; and an adapter can continuously parse the log file and populate a stream with records. This query computes the number of requests for each page each minute, as shown in the accompanying table.

The example here is expressed in SQLstream's query language, as are others in this article. The language is standard SQL plus streaming extensions.[4] Other streaming query engines have similar capabilities.

```
SELECT STREAM ROWTIME,
  uri,
  COUNT(*)
FROM PageRequests
GROUP BY
  FLOOR(ROWTIME TO MINUTE),
  uri;
```

The only SQL extensions used in this particular query are the STREAM keyword and the ROWTIME system column. If you removed the STREAM keyword and converted PageRequests to a table with a column called ROWTIME, you could execute the query in a conventional database such as Oracle or MySQL. That query would analyze all requests that have ever occurred up until the current moment. If PageRequests is a stream, however, the STREAM keyword tells SQLstream to attach to the PageRequests stream and to apply the operation to all future records. Streaming queries run forever.

Every minute this query emits a set of rows, summarizing the traffic for each page during that minute. The output rows time-stamped 10:00:00 summarize all requests between 10:00 and 10:01 (including the 10:00 end point but not including 10:01). Rows in the PageRequests stream are sorted by their ROWTIME system column, so the 10:00:00 output rows are literally pushed out by the arrival of the first row time-stamped 10:01:00 or later. A streaming query engine tends to process data and deliver results only when new data arrives, so it is said to use push-based processing. This is in contrast to a relational database's pull-based approach where the application must poll repeatedly for new results.

The example in Figure 1 computes URIs for which the number of requests is much higher than normal. First, the PageRequestsWithCount view computes the number of requests per hour for each URI over the past hour and averaged over the past 24 hours. Then a query selects URIs for which the rate over the past hour was more than three times the hourly rate over the past 24 hours.

Unlike the previous query that used a GROUP BY clause to aggregate many records into one total per time period, this query uses windowed aggregate expressions (*aggregate-function* OVER *window*) to add analytic values to each row. Because each row is annotated with its trailing hour's and day's statistics, you need not wait for a batch of rows to be complete. You can use such a query to continuously populate a "Most popular pages" list on your Web site, or an e-commerce site could use it to detect products selling in higher than normal volumes.

Figure 1. Streaming query to find Web pages with higher than normal volume.

```
CREATE VIEW PageRequestsWithCount AS
SELECT STREAM ROWTIME,
    uri,
    COUNT(*) OVER lastHour AS hourlyRate,
    COUNT(*) OVER lastDay / 24 AS hourlyRateL-
astDay
FROM PageRequests
WINDOW lastHour AS (
        PARTITION BY uri
        RANGE INTERVAL '1' HOUR PRECEDING)
    lastDay AS (
        PARTITION BY uri
        RANGE INTERVAL '1' DAY PRECEDING);

SELECT STREAM *
FROM PageRequestsWithCount
WHERE rate > hourlyRateLastDay * 3;
```

Comparing Databases and Streaming Query Engines

A database and a streaming query engine have similar SQL semantics, but if you use the two systems for problems involving data in flight, they behave very differently. Why is a streaming query engine more efficient for such problems? To answer that question, it helps to look at its pedigree.

Some use the term *streaming database*, which misleadingly implies that the system is storing data. That said, streaming query engines have very strong family connections with databases. Streaming query engines have roots in database research, in particular the Stanford STREAMS project,[1] the Aurora project at MIT/Brown/Brandeis,[2] and the Telegraph project at Berkeley.[3] Streaming query engines are based on the relational model that underlies relational databases and, as we shall see, those underpinnings give them power, flexibility, and industry acceptance.

The relational model, first described by E.F. Codd in 1970, is a simple and uniform way of describing the structure of databases. It consists of relations (named collections of records) and a set of simple operators for combining those relations: *select, project, join, aggregate*, and *union*. A relational database naturally enforces data independence, the separation between the logical structure of data and the physical representation. Because the query writer does not know how data is physically organized, a query optimizer is an essential component of a relational database, to choose among the many possible algorithms for a query.

SQL was first brought to market in the late 1970s. Some say it is not theoretically pure (and it has since been extended to encompass nonrelational concepts such as objects and nested tables), but SQL nevertheless embodies the key principles of the relational model. It is declarative, which enables the query to be optimized, so you (or the system) can tune an application without rewriting it. You can therefore defer tuning a new database schema until the application is mostly written, and you can safely refactor an existing database schema. SQL is simple, reliable, and forgiving, and many developers understand it.

Streams introduce a time dimension into the relational model. You can still apply the basic operators (*select, project, join*, and so forth), but you can also ask, "If I executed that *join* query a second ago, and I execute it again now, what would be the difference in the results?"

This allows us to approach problems in a very different way. As an analogy, consider how you would measure the speed of a car traveling along the freeway. You might look out the window for a mile marker, write down the time, and when you reach the next mile marker, divide the distance between the mile markers by the elapsed time. Alternatively, you might use a speedometer, a device where a needle is moved based on a generated current that is proportional to the angular velocity of the car's wheels, which in turn is proportional to the speed of the car. The mile-marker method converts position and time into speed, whereas the speedometer measures speed directly using a set of quantities proportional to speed.

Position and speed are connected quantities; in the language of calculus, speed is the differential of position with respect to time. Similarly, a stream is the time differential of a table. Just as the speedometer is the more appropriate solution to the problem of measuring a car's speed, a streaming query engine is often much more efficient than a relational database for data-processing applications involving rapidly arriving time-dependent data.

Output from query.

ROWTIME	uri	COUNT(*)
10:00:00	/index.html	15
10:00:00	/images/logo.png	19
10:00:00	/orders.html	6
10:01:00	/index.html	20
10:01:00	/images/logo.png	18
10:01:00	/sitemap.html	2
...		

Streaming Advantage

Why is a streaming query engine more efficient than a relational database for data-in-flight problems?

First, the systems express the problems in very different ways. A database stores data and applications fire queries (and transactions) at the data. A streaming query engine stores queries, and the outside world fires data at the queries. There are no transactions as such, just data flowing through the system.

The database needs to load and index the data, run the query on the whole dataset, and subtract previous results. A streaming query system processes only new data. It holds only the data that it needs (for example, the latest minute), and since that usually fits into memory easily, no disk I/O is necessary.

A relational database operates under the assumption that all data is equally important, but in a business application, what happened a minute ago is often more important than what happened yesterday, and much more important than what happened a year ago. As the database grows, it needs to spread the large dataset across disk and create indexes so that all of the data can be accessed in constant time. A streaming query engine's working sets are smaller and can be held in memory; and because the queries contain window specifications and are created before the data arrives, the streaming query engine does not have to guess which data to store.

A streaming query engine has other inherent advantages for data in flight: reduced concurrency control overhead and efficiencies from processing data asynchronously. Since a database is writing to data structures that other applications can read and write, it needs mechanisms for concurrency control; in a streaming query engine there is no contention for locks, because incoming data from all applications is placed on a queue and processed when the streaming query engine is ready for it.

In other words, the streaming query engine processes data asynchronously. Asynchronous processing is a feature of many high-performance server applications, from transaction processing to email processing, as well as Web crawling and indexing. It allows a system to vary its unit of work—from a record at a time when the system is lightly loaded to batches of many rows when the load is heavier—to achieve efficiency benefits such as locality-of-reference. One might think an asynchronous system has a slower response time, because it processes the data "when it feels like it," but an asynchronous system can achieve a given throughput at much lower system load, and therefore have a better response time than a synchronous system. Not only is a relational database synchronous, but it also tends to force the rest of the application into a record-at-a-time mode.

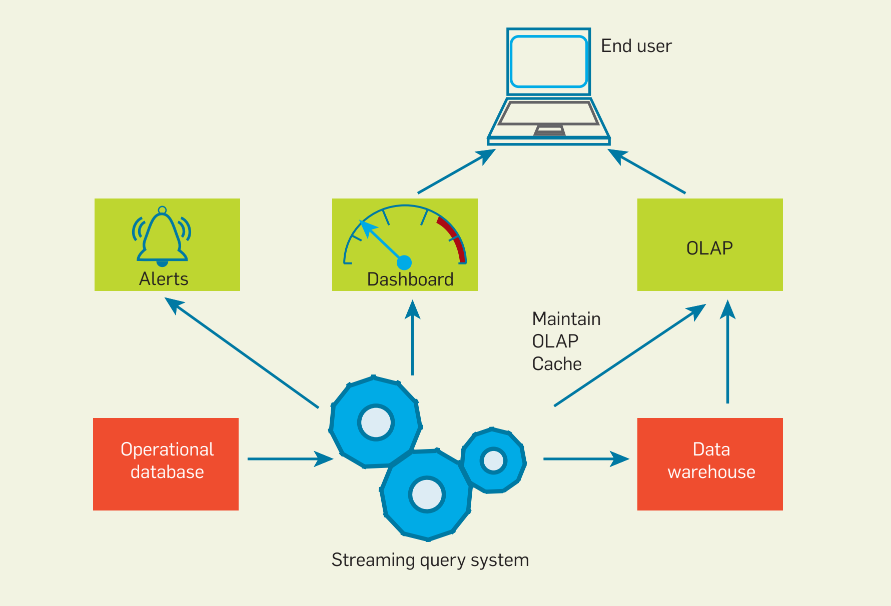
It should be clear by now that push-based processing is more efficient for data in flight; however, a streaming query engine is not the only way to achieve it. Streaming SQL does not make anything possible that was previously impossible. For example, you could implement many problems using a message bus, messages encoded in XML, and a procedural language to take messages off the bus, transform them, and put them back onto the bus. You would, however, encounter problems of performance (parsing XML is expensive), scalability (how to split a problem into sub-problems that can be handled by separate threads or machines), algorithms (how to combine two streams efficiently, correlate two streams on a common key, or aggregate a stream), and configuration (how to inform all of the components of the system if one of the rules has changed). Most modern applications choose to use a relational database management system to avoid dealing with data files directly, and the reasons to use a streaming query system are very similar.

Other Applications of Streaming Query Systems

Just as relational databases are a horizontal technology, used for everything from serving Web pages to transaction processing and data warehousing, streaming SQL systems are being applied to a variety of problems.

Application areas include complex event processing (CEP), monitoring, population data warehouses, and middleware. A CEP query looks for sequences of events on a single stream or on multiple streams that, together, match a pattern and create a "complex event" of interest to the business. Applications of CEP include fraud detection and electronic trading.

Figure 2. Continuous ETL using a streaming query system.



CEP has been used within the industry as a blanket term to describe the entire field of streaming query systems. This is regrettable because it has resulted in a religious war between SQL-based and non-SQL-based vendors and, in overly focusing on financial services applications, has caused other application areas to be neglected.

The click-stream queries here are a simple example of a monitoring application. Such an application looks for trends in the transactions that represent the running business and alerts the operations staff if things are not running smoothly. A monitoring query finds insights by aggregating large numbers of records and looking for trends, in contrast to a CEP query that looks for patterns among individual events. Monitoring applications may also populate real-time dashboards, a business's equivalent of your car's speedometer, thermometer, and oil pressure gauge.

Because of their common SQL language, streaming queries have a natural synergy with data warehouses. The data warehouse holds the large amount of historical data necessary for a "rear-view mirror" analysis of the business, while the streaming query system continuously populates the data warehouse and provides forward-looking insight to "steer the company."

The streaming query system performs the same function as an ETL (extract, transform, load) tool but operates continuously. A conventional ETL process is a sequence of steps invoked as a batch job. The cycle time of the ETL process limits how current the data warehouse is, and it is difficult to get that cycle time below a few minutes. For example, the most data-intensive steps are performed by issuing queries on the data warehouse: looking up existing values in a dimension table, such as customers who have made a previous purchase, and populating summary tables. A streaming query system can cache the information required to perform these steps, offloading the data warehouse, whereas the ETL process is too short-lived to benefit from caching.

Figure 2 shows the architecture of a real-time business intelligence system. In addition to performing continuous ETL, the streaming query system populates a dashboard of business metrics, generates alerts if metrics fall outside acceptable bounds, and proactively maintains the cache of an OLAP (online analytical processing) server that is based upon the data warehouse.

Today, much "data in flight" is transmitted by message-oriented middleware. Like middleware, streaming query systems can deliver messages reliably, and with high throughput and low latency; further, they can apply SQL operations to route, combine, and transform messages in flight. As streaming query systems mature, we may see them stepping into the role of middleware and blurring the boundaries between messaging, continuous ETL, and database technologies by applying SQL throughout.

Conclusion

Streaming query engines are based on the same technology as relational databases but are designed to process data in flight. Streaming query engines can solve some common problems much more efficiently than databases because they match the time-based nature of the problems, they retain only the working set of data needed to solve the problem, and they process data asynchronously and continuously.

Because of their shared SQL language, streaming query engines and relational databases can collaborate to solve problems in monitoring and real-time business intelligence. SQL makes them accessible to a large pool of people with SQL expertise.

Just as databases can be applied to a wide range of problems, from transaction processing to data warehousing, streaming query systems can support patterns such as enterprise messaging, complex event processing, continuous data integration, and new application areas that are still being discovered.

Related articles on queue.acm.org

A Call to Arms
Jim Gray and Mark Compton
http://queue.acm.org/detail.cfm?id=1059805

Beyond Relational Databases
Margo Seltzer
http://queue.acm.org/detail.cfm?id=1059807

A Conversation with Michael Stonebraker and Margo Seltzer
http://queue.acm.org/detail.cfm?id=1255430

References

1. Arasu, A., Babu, S., Widom, J. The CQL Continuous Query Language: Semantic Foundations and Query Execution. Technical Report. Stanford University, Stanford, CA, 2003.
2. Aurora project; http://www.cs.brown.edu/research/aurora.
3. Chandrasekaran, S., et al. TelegraphCQ: Continuous dataflow processing for an uncertain world. In *Proceedings of Conference on Innovative Data Systems Research* (2003).
4. SQLstream Inc.; http://www.sqlstream.com.

Julian Hyde is chief architect of SQLstream, a streaming query engine. He is also the lead developer of Mondrian, the most popular open source relational OLAP engine and a part of the Pentaho open source BI suite. An expert on relational technology, including query optimization and streaming execution, Hyde introduced bitmap indexes into Oracle and led development of the Broadbase analytic DBMS.

DOI:10.1145/1629175.1629196

Article development led by acmqueue
queue.acm.org

Companies have access to more types of external data than ever before. How can they integrate it most effectively?

BY STEPHEN PETSCHULAT

Other People's Data

EVERY ORGANIZATION BASES some of its critical decisions on external data sources. In addition to traditional flat file data feeds, Web services and Web pages are playing an increasingly important role in data warehousing. The growth of Web services has made data feeds easily consumable at the departmental and even end-user levels. There are now more than 1,500 publicly available Web services and thousands of data mashups ranging from retail sales data to weather information to U.S. census data.[3] These mashups are evidence that when users need information, they will find a way to get it. An effective enterprise information management strategy must take into account both internal and external data.

External data sources vary in their structure and methods of access. Some are comprehensive and have been a part of data-warehousing flows for many years: securities data, corporate information, credit risk data, and address/postal code lookup. These are typically structured in a formal manner, contain the "base" (most detailed) level of data, and are available through established data service providers in multiple formats. The most common access method is still flat files over FTP.

Web services are well understood

from a software development perspective, but their relevance to enterprise data management is just emerging. Traditional data service providers such as Dunn & Bradstreet and Thomson Reuters have started offering most of their products via Web services. Hundreds of smaller Web service companies are also providing data in areas such as retail sales, Web trends, securities, currency, weather, government, medicine, current events, real estate, and competitive intelligence.

With Web services comes the ability to add value in the form of functional services. Instead of allowing the retrieval of a flat file of data—effectively a `fetch()` function—it is easy for a data provider to add services on top of the data: conversions, calculations, searching, and filtering. In fact, for most of the smaller upstarts the emphasis has been heavily on the functional services, which typically present a more highly processed subset of the overall data. This can save time and effort when the functions provided are what you need. The data is precleansed, aggregated at the right level, and you don't have to implement your own search.

This ability can also lead to challenges, however, when the functional interfaces don't match your exact needs. EBay provides marketplace research data on the daily best-selling products in various categories, top vendors, and bid prices for products. This works well if these specific queries are what you want, but if you require a query that eBay has not thought of, you don't have access to the base data to create that custom query yourself.

Another important data source is the Web itself. A great deal of unstructured data exists in Web pages and behind search engines. The area of competitive intelligence has been driving the merging of unstructured and semistructured Web sources with the data warehouse. Competitive intelligence is also an area that is driving the shift from solely back-end data feeds to those all the way through the stack.[1] Structured Web services from Amazon and eBay are one source of specific market and sales information, while technologies from companies such as Kapow and Dapper allow users to turn any external Web page content into semistructured data feeds by mapping some of the visual fields on the page to data fields in the dynamic feed.

Although these tools are beginning to make Web scraping easier, most end users still resort to cutting and pasting from competitors' Web sites into spreadsheets in order to gain the insights they need to do their jobs. This is a manually intensive and error-prone approach, but the alternative—sourcing market information, talking to IT, integrating the datasets into the core data-warehouse model, staging, testing, deploying—takes too long, particularly when sources may be changing on a weekly or monthly basis.

Architectural Considerations

External data should be considered and planned for differently from internal data. Much the way distributed computing architectures must account for latency and data failures, a robust data-warehousing plan must take into account the fact that external sources are not, by definition, in the sphere of control of the receiving organization. They are more prone to unpredictable delays, data anomalies, schema changes, and semantic data changes. That is not to say they are lower quality; plenty of internal sources have the same issues, and data-service companies are paid to provide top-quality data. However, the communication channel and processing systems are inter- rather than intracompany, creating an additional source of issues and delays.

Competitive intelligence data (legally) scraped off of publicly available sites must not contend with cleanliness issues, but the schema can also change at any time and the publisher has no obligation to notify consumers of that change. If a scheduled report relies on this information, then it will often break, resulting in a blank or incomprehensible report. Even worse, it could result in incorrect numbers, leading to bad business decisions. Data reliability and accuracy must be considered fundamental attributes throughout the data's flow in the organization.

International Securities Exchange historical options data ticker.

```
TRADE_DT,UNDLY,SEC_TYPE,SYM_ROOT,...80+  fields...,ISE_VOL,TOTAL_VOL
20090810,AAPL,1,QAA , ... ,2241,164.72,0.01,1,2
20090810,AAPL,1,QAA, ... ,2347,164.72,0.02,1,2
20090810,AAPL,1,QAA, ... ,3591,164.72,0.03,7,130
20090810,AAPL,1,APV, ... ,2714,164.72,40.7,10,15
…
```

Flexibility, Quality, and Cost

Not all data needs to go through the entire data-warehouse workflow. In information-intensive organizations the IT group can rarely accommodate every user's data needs in a timely manner. Trade-offs must be made. These trade-offs can be considered along the dimensions of flexibility, quality, and cost. In most cases, you can pick two and trade off the third.

Flexibility refers to how easily you can purpose the data for the end users' needs. Getting base data from raw flat files maximizes your ability to massage the data further; however, the effort involved is much higher than getting a highly summarized feed from a Web service vendor. For example, the International Securities Exchange historical options daily ticker data for a single stock symbol has more than 80 fields[2] (see the accompanying box).

Having all of the base data in CSV (comma-separated values) format provides maximum flexibility; you can derive any information you want from it. However, if all you require is the high-level summary information, you would be better off giving up that flexibility in exchange for a simpler API from a Web service provider such as StrikeIron or Xignite. For example, GET http://www.xignite.com/xquotes.asmx/GetSingleQuote?Symbol=AAPL

```
<QuickQuote>
   <Symbol>AAPL</Symbol>
   <Last>188.50</Last>
   <Change>7.85</Change>
   <Volume>25,094,395</Volume>
   <Time>4:01pm ET</Time>
</QuickQuote>
```

In the case of securities information, very few IT shops can afford to manage a raw stock feed directly from the exchanges. The monthly uncompressed data for the International Securities Exchange historical options ticker data is more than a terabyte, so even handling daily deltas can be multiple gigabytes on a full financial stream.[2]

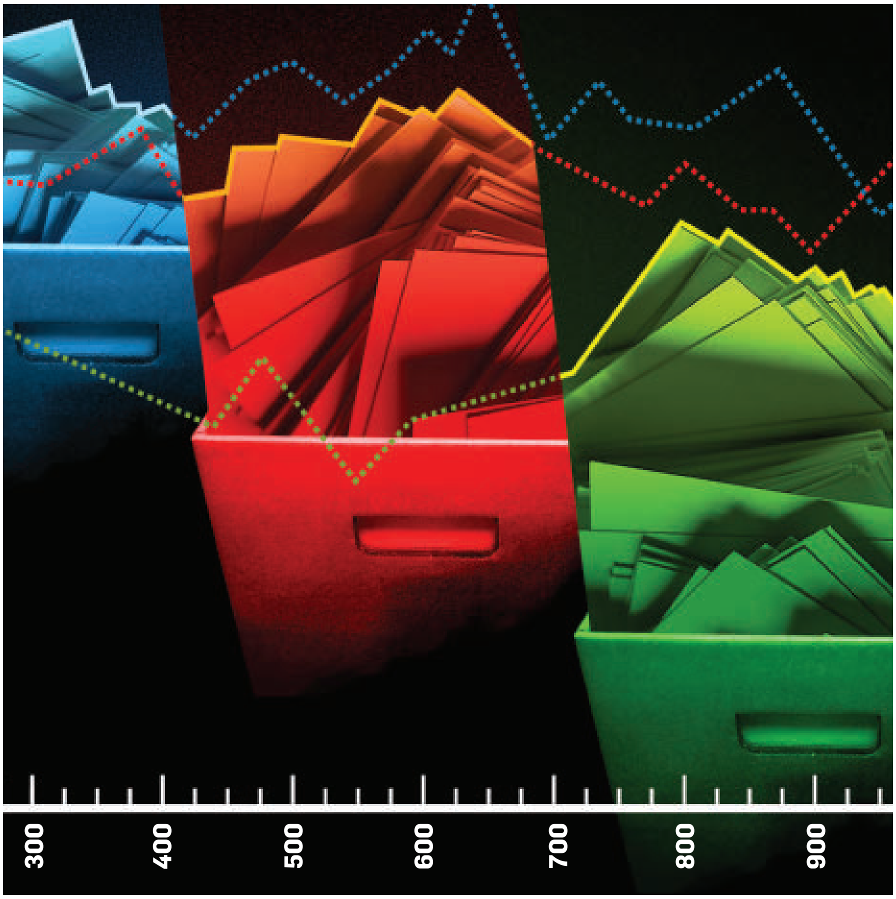
Ticker information alone is not very useful without symbol lookup tables and other corporate data with which to cross-reference it. These may come from separate sources and may or may not handle data changes for you—someone has to recognize the change from SUNW to JAVA to ORCL and ensure it is handled meaningfully. Different feeds also come in at different rates. Data can change for technical, business, and regulatory reasons, so keeping it in sync with a usable data model is a full-time job.

Quality is both a function of the source of the data and the process by which it flows through the organization. If a complex formula shows up in hundreds of different reports authored by dozens of different people, the chance of introducing errors is almost certain. Adding up all of the invoices will produce a revenue number, but it may not take into account returns, refunds, volume discounts, and tax rebates. Calculations such as revenue and profit typically rely upon specific assumptions of the underlying data, and any formulas based on them need to know what these assumptions are:

- Do all of the formulas use the exact same algorithm?
- How do they deal with rounding errors?
- Have currency conversions been applied?
- Has the data been seasonally adjusted?
- Are nulls treated as zeros or a lack of data?

The more places a formula is managed, the more likely errors will be introduced.

Cost can be traded off against both quality and flexibility. With enough people hand-inspecting every report, it doesn't matter how many times things are duplicated—quality can be maintained through manual quality assurance. However, this is not the most efficient way to run a data warehouse. Cost and flexibility typically trade off based on the effort necessary to take raw data and turn it into useful data. Each level of processing takes effort and loses flexibility unless you are willing to invest even more effort to maintain both base and processed data.



Of course, you can always keep all of the base data around forever, but the cost of maintaining this can be prohibitive. Having everything in the core data warehouse at the lowest possible level of detail represents the extreme of maximizing flexibility and quality while trading off cost.

External Web services typically trade off flexibility in exchange for quality and cost. Highly summarized, targeted data services with built-in assumptions, calculations, and implicit filters are not as flexible, but they are often all that is needed to solve a specific problem. It doesn't matter that the daily weather average for a metropolitan area is actually sampled at the airport and the method of averaging isn't explicit. This loss of flexibility is a fair trade when all you care about is what the temperature was that day +/-3 degrees.

The following are questions to ask when determining which trade-offs make sense for a given data source:

- What is the business impact of incorrect data?
- What is the cost of maintaining the data feed?
- How large are the datasets?
- How often does the data change?
- How often does the data schema change?
- How complex is the data?
- How complex and varied are the consumption scenarios?
- What is the quality of the data (how many errors expected, how often, magnitude of impact)?
- How critical is the data to decision making?
- What are the auditing and traceability requirements?
- Are there any regulatory concerns?
- Are there any privacy or confidentiality concerns?

Enterprise Data Mashups

Traditional warehouse life cycle, topology, and modeling approaches are not well suited for external data integration. The data warehouse is often considered a central repository; the single source of truth. In reality, it can rarely keep up with the diversity of information needed by the organization. This results in users resorting to sourcing their own external data, exporting to Excel, and doing their own data joins in spreadsheets.

Self-sourcing of data and user-driven data integration can alleviate some of the burden on IT, but it can also cause problems. It is not always optimal to have mission-critical decisions being made based on data mashups downloaded from the Internet. Few users understand the intricacies of data cleansing, relational joins, and managing slowly changing dimensions. Proper data modeling hasn't gotten any easier.

The range of approaches—from end-user-driven on-screen mashups to centralized data-management teams with detailed master data-management plans—all have their place in a modern data warehousing strategy. A decentralized data strategy has clear quality and consistency trade-offs, but not all data is equal—sometimes faster, more flexible access to data is more important than getting it perfect. The decision should be made based on an awareness of the quality, flexibility, and cost tradeoffs. These are often linked to where the integration occurs.

Integrating at Different Layers in the Architecture

External data can be incorporated at any of the stages of the enterprise information flow: during ETL (extract-transform-load), in the core data warehouse, at departmental-level data marts, and during end-user consumption via reports, BI (business intelligence) tools, and applications. The two most common stages for data integration are in the initial ingestion or at the final consumption, but all stages merit consideration.

ETL. Enriching data on the way in is a common technique. All the major vendors provide ETL tools to cleanse, transform, and augment datasets on their way into the warehouse. Data problems (errors, omissions, and incompatibilities) are handled in an explicit manner at this layer, leading to high-quality data before it even hits the warehouse. Many of these tools now incorporate some level of Web service integration. However, ETL tools are typically geared toward large-volume batches, whereas most Web-service interfaces tend to assume single or small datasets per request. For this reason, Web services tend to play more prominently as the data gets closer to the user.

The data warehouse is often considered a central repository; the single source of truth. In reality, it can rarely keep up with the diversity of information needed by of the organization.

Some types of data lend themselves well to a purely enriching role and do not need to augment the existing data model. These are typically good candidates for ETL stage integration. This is a common role for address and geolocation-related data. During the ETL process addresses are compared against an external data source. Zip codes are corrected, street names are normalized, and latitude and longitude fields are added. The external source does not result in new tables in the warehouse with joins to the internal data. The data is just fixed up on the way in, perhaps adding a field here or there.

Core Data Warehouse. Integrating at the core model, as is commonly done in the traditional flat-files-via-FTP model, ensures the greatest level of adherence to enterprise-data standards and data-management processes. Data cleanliness issues, interruptions in service, and semantic mismatches are all dealt with in the core model of the warehouse. The currency-conversion tables are consistent across the whole system, promoting the much sought-after "one version of the truth." For financial information this is often essential.

In many cases, however, this is also the most costly approach to maintain. For rapidly changing business conditions, end users must engage in a dialog with IT and put up with data policies and procedures they may not understand or care about (even if they should). Changing your database schema is more difficult than changing your reports. If the information is important enough to the users and the system friction too great, then they will find a workaround.

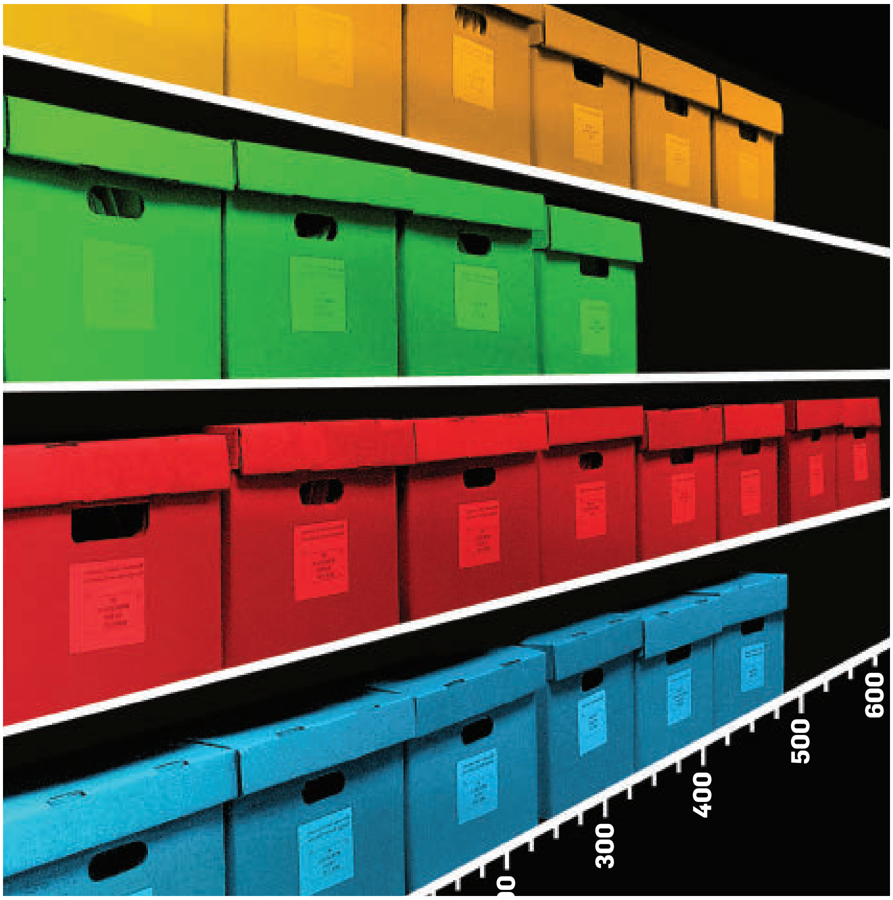
Data Mart. Integrating external data at the mart layer reduces some of the friction typical of core data-warehouse-model integration. Although data-warehouse topologies vary considerably, for this purpose we consider the data mart to be an entity at least partially in the departmental sphere of control. Each mart—whether a separate database instance from the warehouse, an OLAP (online analytical processing) cube, or a completely different database technology—can source and integrate its own feeds, creating a perspective on the data that is relevant to that department or user base. The marketing department may need an external address database for cleansing before doing mass mailings, while the customer support organization may be more interested in geo-tagging for the purpose of analyzing case distribution patterns.

As integration moves closer to the end user, the key issue to be aware of is loss of data control. If each mart integrates the same data in a slightly different way, then the chances are greater that inconsistencies will be introduced. The sales group in North America may be adjusting for refunds to take into account high rotation in its box stores, while the Asia Pacific group is not. A report that tries to aggregate the results from these two independent data marts may naively sum the numbers, resulting in an incorrect result. There may be good reasons for different rules at the data-mart level, but when values are combined or compared downstream, these differences can cause problems.

BI Tools. Most BI tool vendors have now incorporated data-mashup capabilities so end users can join external and internal data. This approach tends to be more end-user driven than doing it at the data-mart layer and often requires no administration access to the database or formal data-modeling expertise. Vendors' tools vary in their approaches, but typically there are options to do lightweight business-user modeling, as well as on-screen combining of datasets via formula languages. This provides a great deal of flexibility while maintaining the ability to audit and trace usage.

The issues faced at this layer are similar to those of the data mart. Aspects of the data management and integration are still decentralized, which can result in redundant definitions of the same business concepts, increasing the risk of incorrect interpretations. Facilities for auditing and tracing, as well as common business-user-level semantic layers, help overcome some of the issues.

Desktop. This end of the spectrum represents the most common mashup scenario. Excel, flat files, macros, and Web application integration are easy to throw together and occasionally are even accurate. While database administrators and warehouse architects may cringe, the bottom line is that when people need information they will find a way to get it. From an information architecture point of view, planning for this eventuality allows you to decide which data belongs where and what the potential impacts will be in an informed way. For some data sources, where accuracy and traceability are not critical, this is a perfectly acceptable choice. As these data sources become more heavily used, though, it may make sense to push them further down the stack to ensure the data-integration work is done by only one person rather than many.



Conclusion

There is no correct layer to integrate external data into the enterprise information flow, rather a set of trade-offs to consider. The characteristics of the data, its consumption scenarios, and the business context all must be considered. These factors also need to be re-evaluated periodically. As a data source becomes more widely used, economics may dictate centralizing and formalizing data acquisition. Choosing the right integration approach for an external data source can balance the variables of flexibility, quality, and cost while providing end users with timely answers to their business questions. C

Related articles on queue.acm.org

Why Your Data Won't Mix
Alon Halevy
http://queue.acm.org/detail.cfm?id=1103836

The Pathologies of Big Data
Adam Jacobs
http://queue.acm.org/detail.cfm?id=1563874

A Conversation with Michael Stonebraker and Margo Seltzer
http://queue.acm.org/detail.cfm?id=1255430

References

1. Boncella, R.J. Competitive intelligence and the Web. *Commun. AIS 12* (2003): 327–340; http://www.washburn.edu/faculty/boncella/COMPETITIVE-INTELLIGENCE.pdf.
2. International Securities Exchange; http://www.ise.com/.
3. Programmableweb.com; http://www.programmableweb.com/.

Stephen Petschulat is a senior product architect in the advanced analytics area of SAP Business Objects.

DOI:10.1145/1629175.1629194

Article development led by acmqueue
queue.acm.org

As hard-drive capacities continue to outpace their throughput, the time has come for a new level of RAID.

BY ADAM LEVENTHAL

Triple-Parity RAID and Beyond

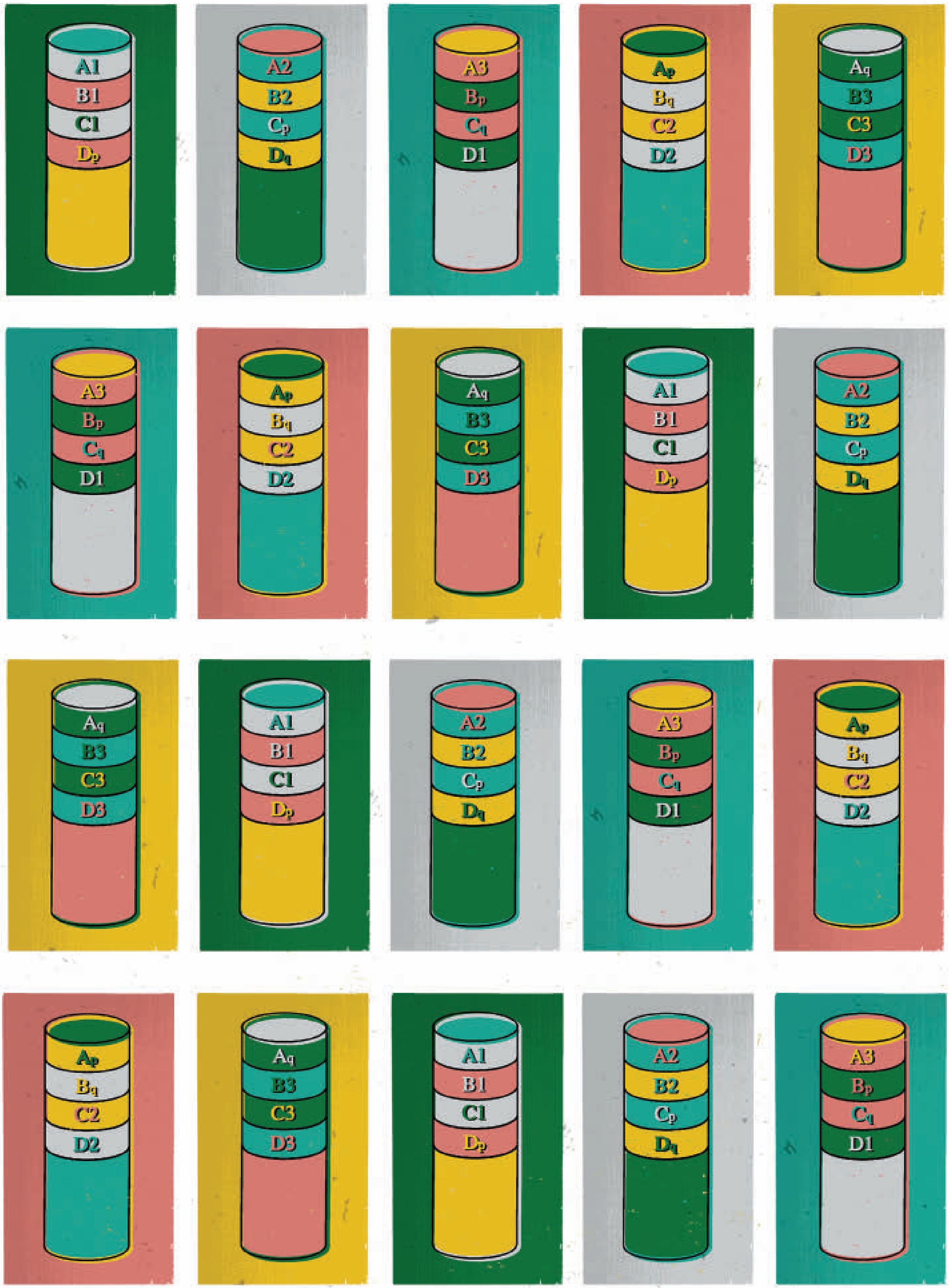
HOW MUCH LONGER will current RAID techniques persevere? The RAID levels were codified in the late 1980s; double-parity RAID, known as RAID-6, is the current standard for high-availability, space-efficient storage. The incredible growth of hard-drive capacities, however, could impose serious limitations on the reliability even of RAID-6 systems. Recent trends in hard drives show that triple-parity RAID must soon become pervasive. In 2005, *Scientific American* reported on Kryder's Law,[11] which predicts that hard-drive density will double annually. While the rate of doubling has not quite maintained that pace, it has been close.

Problematically for RAID, hard-disk throughput has failed to match that exponential rate of growth. Today repairing a high-density disk drive in a RAID group can easily take more than four hours, and the problem is getting significantly more pronounced as hard-drive capacities continue to outpace their throughput. As the time required for rebuilding a disk increases, so does the likelihood of data loss. The ability of hard-drive vendors to maintain reliability while pushing to higher capacities has already been called into question in these pages.[5] Perhaps even more ominously, in a few years, reconstruction will take so long as to effectively strip away a level of redundancy. What follows is an examination of RAID, the rate of capacity growth in the hard-drive industry, and the need for triple-parity RAID as a response to diminishing reliability.

The first systems that would come to be known as RAID were developed in the mid-1980s. David Patterson, Garth Gibson, and Randy Katz of the University of California, Berkeley, classified those systems into five distinct categories under the umbrella of RAID (redundant arrays of inexpensive disks).[9] In their 1988 paper, RAID played David to the Goliath of SLED (single large expensive disks). The two represented fundamentally different strategies for how to approach the future of computer storage. While SLED offered specialized performance and reliability—at a price—RAID sought to assemble reliable, high-performing storage from cheap parts, reflecting a broader trend in the computing industry. The economics of commodity components are unstoppable.

Patterson et al. were seemingly prescient in their conclusion: "With advantages in cost-performance, reliability, power consumption, and modular growth, we expect RAIDs to replace SLEDs in future I/O systems."[9] However, their characterization of RAID as "a disk array made from personal computer disks" was a bit too specific and a bit too hopeful. While RAID is certainly used with those inexpensive, high-volume disks, RAID in its de facto incarnation today combines its algorithmic reliability and performance improvements with disks that are themselves often designed for performance and reliability, and therefore

ILLUSTRATION BY GARY NEILL


A1 B1 C1 Dp
A2 B2 Cp Dq
A3 Bp Cq D1
Ap Bq C2 D2
Aq B3 C3 D3
A3 Bp Cq D1
Ap Bq C2 D2
Aq B3 C3 D3
A1 B1 C1 Dp
A2 B2 Cp Dq
Aq B3 C3 D3
A1 B1 C1 Dp
A2 B2 Cp Dq
A3 Bp Cq D1
Ap Bq C2 D2
Ap Bq C2 D2
Aq B3 C3 D3
A1 B1 C1 Dp
A2 B2 Cp Dq
A3 Bp Cq D1

remain expensive. This evolution is reflected in the subtle but important mutation of the meaning of the I in RAID from *inexpensive* to *independent* that took place in the mid-1990s (indeed, it was those same SLED manufacturers that instigated this shift to apply the new research to their existing products).

In 1993, Gibson, Katz, and Patterson, along with Peter Chen, Edward Lee, completed a taxonomy of RAID levels that remain unamended to date.[3]

Of the seven RAID levels described, only four are commonly used:

▸ **RAID-0.** Data is striped across devices for maximal write performance. It is an outlier among the other RAID levels as it provides no actual data protection.

▸ **RAID-1.** Disks are organized into mirrored pairs and data is duplicated on both halves of the mirror. This is typically the highest-performing RAID level, but at the expense of lower usable capacity. (The term *RAID-10* or *RAID-1+0* is used to refer to a RAID configuration in which mirrored pairs are striped, and *RAID-01* or *RAID-0+1* refer to striped configurations that are then mirrored. The terms are of decreasing relevance since striping over RAID groups is now more or less assumed.)

▸ **RAID-5.** A group of N+1 disks is maintained such that the loss of any one disk would not result in data loss. This is achieved by writing a parity block, P, for each logical row of N disk blocks. The location of this parity is distributed, rotating between disks so that all disks contribute equally to the delivered system performance. Typically P is computed simply as the bitwise XOR of the other blocks in the row.

▸ **RAID-6.** This is like RAID-5, but employs two parity blocks, P and Q, for each logical row of N+2 disk blocks. There are several RAID-6 implementations such as IBM's EVENODD,[2] NetApp's Row-Diagonal Parity,[4] or more generic Reed-Solomon encodings.[10] (Chen et al. refer to RAID-6 as P+Q redundancy, which some have taken to imply P data disks with an arbitrary number of parity disks, Q. In fact, RAID-6 refers exclusively to double-parity RAID; P and Q are the two parity blocks.) For completeness, it's worth noting the other less prevalent RAID levels:

▸ **RAID-2.** Data is protected by memory-style ECC (error correcting codes). The number of parity disks required is proportional to the log of the number of data disks; this makes RAID-2 relatively inflexible and less efficient than RAID-5 or RAID-6 while also delivering lower performance and reliability.

▸ **RAID-3.** As with RAID-5, protection is provided against the failure of any disk in a group of N+1, but blocks are carved up and spread across the disks—bitwise parity as opposed to the block parity of RAID-5. Further, parity resides on a single disk rather than being distributed between all disks. RAID-3 systems are significantly less efficient than with RAID-5 for small read requests; to read a block all disks must be accessed; thus the capacity for read operations is more readily exhausted.

▸ **RAID-4.** This is merely RAID-5, but with a dedicated parity disk rather

Figure 1. Comparison of RAID-5 and RAID-6 reliability.[1]

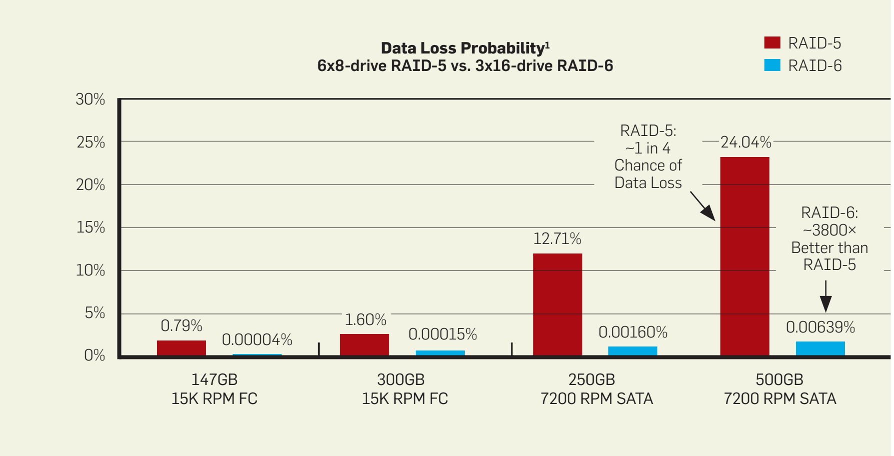


Figure 2. Historical Capacity/Throughput of 7200 RPM SATA HDDs.

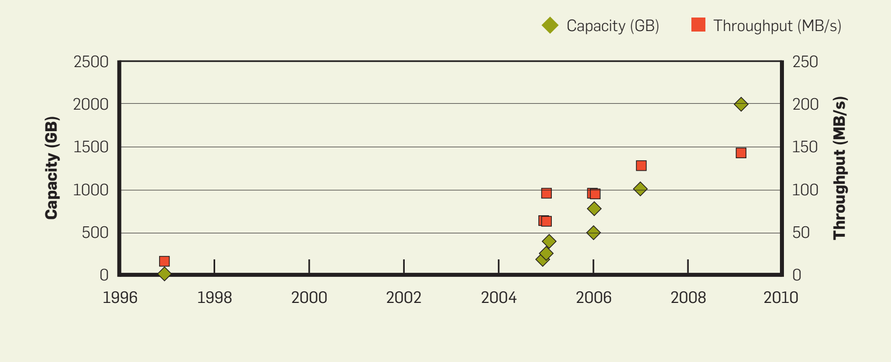
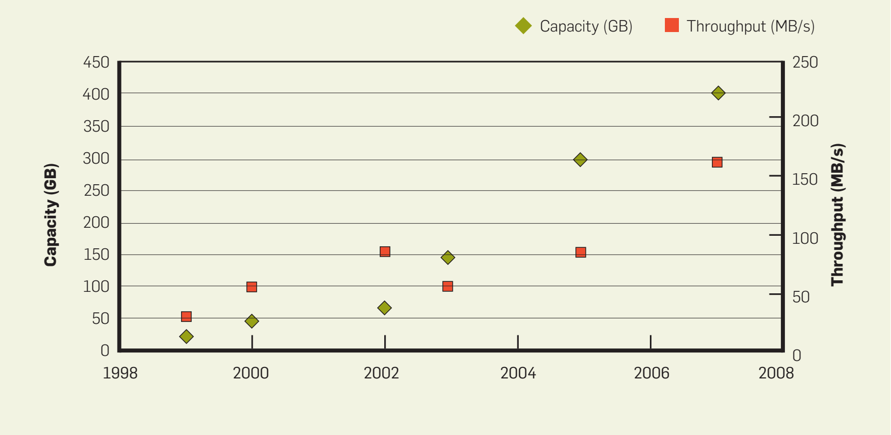


Figure 3. Historical Capacity/Throughput of 10K RPM FC HDDs.



than having parity distributed among all disks. Since fewer disks participate in reads (the dedicated parity disk is not read except in the case of a failure), RAID-4 is strictly less efficient than RAID-5.

RAID-6, double-parity RAID, was not described in Patterson, Gibson, and Katz's original 1988 paper[9] but was added in 1993 in response to the observation that as disk arrays grow, so too do the chances of a double failure. Further, in the event of a failure under any redundancy scheme, data on all drives within that redundancy group must be successfully read in order for the data that had been on the failed drive to be reconstructed. A read failure during a rebuild would result in data loss. As Chen et al. state:

"The primary ramification of an uncorrectable bit error is felt when a disk fails and the contents of the failed disk must be reconstructed by reading data from the nonfailed disks. For example, the reconstruction of a failed disk in a 100GB disk array requires the successful reading of approximately 200 million sectors of information. A bit error rate of one in 10^{14} bits implies that one 512-byte sector in 24 billion sectors cannot be correctly read. Thus, if we assume the probability of reading sectors is independent of each other, the probability of reading all 200 million sectors successfully is approximately

$$(1 - 1/(2.4 \times 10^{10}))\ \hat{}\ (2.0 \times 10^{8}) = 99.2\%.$$

This means that on average, 0.8% of disk failures would result in data loss due to an uncorrectable bit error."[3]

Since that observation, bit error rates have improved by about two orders of magnitude while disk capacity has increased by slightly more than two orders of magnitude, doubling about every two years and nearly following Kryder's law. Today, a RAID group with 10TB (nearly 20 billion sectors) is commonplace, and typical bit error rate stands at one in 10^{16} bits:

$$(1 - 1/(2.4 \times 10^{12}))\ \hat{}\ (2.0 \times 10^{10}) = 99.2\%$$

While bit error rates have nearly kept pace with the growth in disk capacity, throughput has not been given its due consideration when determining RAID reliability.

As motivation for its RAID-6 solution, NetApp published a small comparison of RAID-5 and -6 with equal capacities (7+1 for RAID-5 and 14+2 for RAID-6) and hard drives of varying quality and capacity.[1] Note that despite having an additional parity disk, RAID-6 need not reduce the total capacity of the system.[7] Typically the RAID stripe width—the number of disks within a single RAID group—for RAID-6 is double that of a RAID-5 equivalent; thus, the number of data disks remains the same. The NetApp comparison is not specific about the bit error rates of the devices tested, the reliability of the drives themselves, or the length of the period over which the probability of data loss is calculated; therefore, we did not attempt to reproduce these specific results. The important point to observe in Figure 1 is the stark measured difference in the probability of data loss between RAID-5 and RAID-6.

When examining the reliability of a RAID solution, typical considerations range from the reliability of the component drives to the time for a human administrator to replace failed drives. The throughput of drives has not been a central focus despite being critical for RAID reconstruction, because throughput has been more than adequate. While factors such as the bit error rate have kept pace with capacity, throughput has lagged behind, forcing a new examination of RAID reliability.

Capacity vs. Throughput

Capacity has increased steadily and significantly, and the bit error rate has improved at nearly the same pace. Hard-drive throughput, however, has lagged behind significantly. Using vendor-supplied hard-drive data sheets, we've been able to examine the relationship between hard-drive capacity and throughput for the past 10 years. Figures 2–4 show samples for various hard-drive protocols and rotational speeds.

This data presents a powerful con-

Figure 4. Historical Capacity/Throughput of 15K RPM FC HDDs.

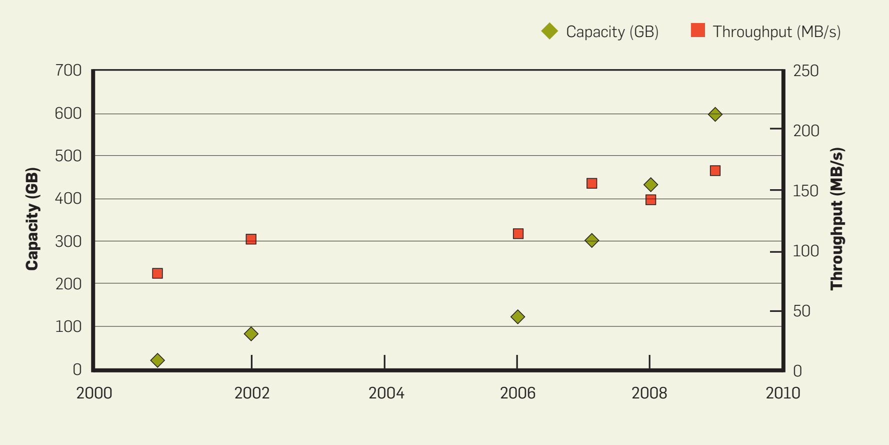
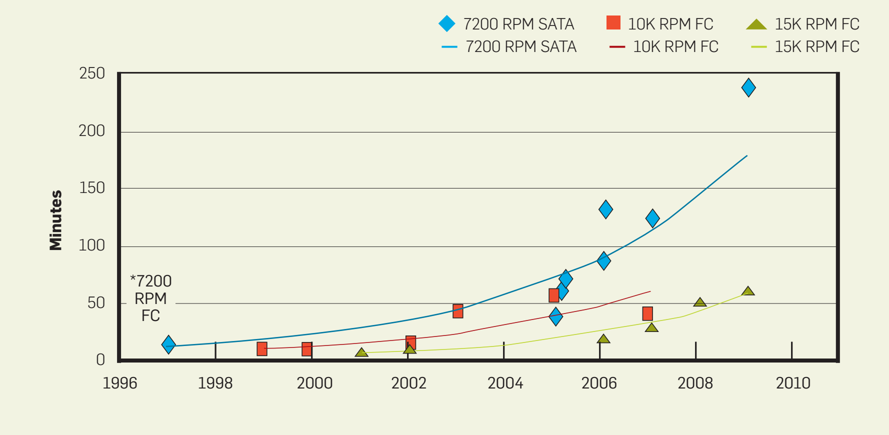


Figure 5. Minimum time required to populate HDDs through the years.



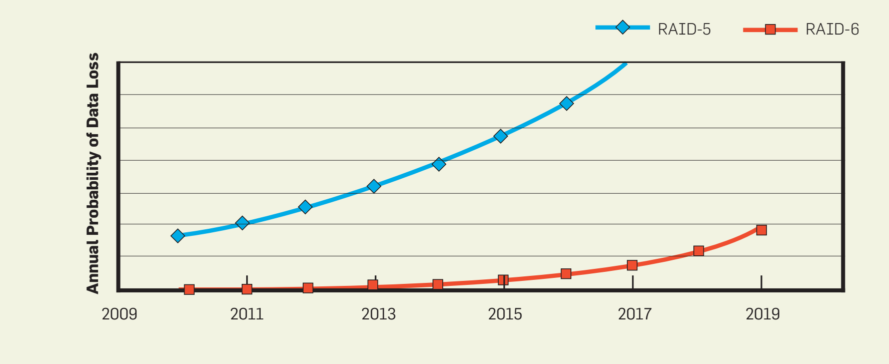


Figure 6. Projected relative reliability of single- and double-parity RAID.

clusion about the relative rates of capacity and throughput growth for hard drives of all types—there's obviously no exponential law governing hard-drive throughput. By dividing capacity by throughput, we can compute the amount of time required to fully scan or populate a drive. It is this duration that dictates how long a RAID group is operating without full parity protection. Figure 5 shows the duration such an operation would take for the various drive types over the years.

When RAID systems were developed in the 1980s and 1990s, reconstruction times were measured in minutes. The trend for the past 10 years is quite clear regardless of the drive speed or its market segment: the time to perform a RAID reconstruction is increasing exponentially as capacity far outstrips throughput. At the extreme, rebuilding a fully populated 2TB 7200-RPM SATA disk—today's capacity champ—after a failure would take four hours operating at the theoretical optimal throughput. It is rare to achieve those data rates in practice; in the context of a heavily used system the full bandwidth can't be dedicated exclusively to RAID repair without adversely affecting performance. If one assumes that only 10%–50% of the total system throughput is available for reconstruction, the minutes-long RAID rebuild times of the 1990s balloon to multiple hours or days in practice. RAID systems operate in this degraded state for far longer than they once did and as a consequence are at higher risk for data loss.

Latent data on hard drives can acquire defects over time—a process blithely referred to as bit rot. To mitigate this, RAID systems typically perform background scrubbing in which data is read, verified, and corrected as needed to eradicate correctable failures before they become uncorrectable.[5] The phenomenon of scrubbing data necessarily impacts system performance, but the time required for a full scrub is a significant component of the reliability of the total system. A natural tension results between how priorities are assigned to scrubbing versus other system activity. As throughput is dwarfed by capacity, either the percentage of resources dedicated to scrubbing must increase, or the time for a complete scrub must increase. With the trends noted previously, storage pools will easily take weeks or months for a full scrub regardless of how high a priority scrubbing is given, further reducing the reliability of the total system as it becomes more likely that RAID reconstructions will encounter latent data corruption.

Given the growing disparity between the capacity growth of hard drives and improvements to their performance, the long-term prospects of RAID-6 must be reconsidered. The time to repair a failed drive is increasing, and at the same time the lengthening duration of a scrub means that errors are more likely to be encountered during the repair. In Figure 6, we have chosen reasonable values for the bit error rate and annual failure rate, and a relatively modest rate of capacity growth (doubling every three years). This is meant to approximate the behavior of low-cost, high-density, 7200-RPM drives. Different values would change the precise position of the curves, but not their relative shapes.

RAID-5 reached a threshold 15 years ago at which it no longer provided adequate protection. The answer then was RAID-6. Today RAID-6 is quickly approaching that same threshold. In about 10 years, RAID-6 will provide only the level of protection that we get from

A Classification for Triple-Parity RAID

None of the existing RAID classifications apply for triple-parity RAID. One option would be to extend the existing RAID-6 definition, but this could be confusing, as many RAID-6 systems exist today. The next obvious choice is RAID-7, but rather than applying the designation merely to RAID with triple-parity protection, RAID-7 should be a catch-all for any RAID technique that can be extended to an arbitrary number of parity disks. Specific techniques or deployments that fix the number of parity disks at N should use the RAID-7.N nomenclature with RAID-7.3 referring to triple-parity RAID, and RAID-5 and RAID-6 effectively as the degenerate forms RAID-7.1 and RAID-7.2, respectively.

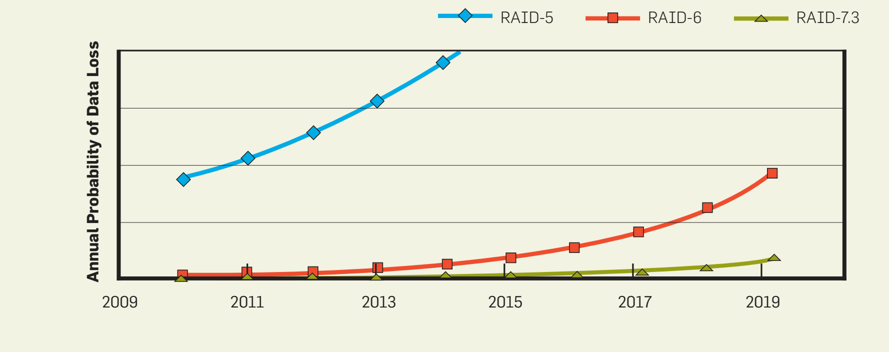


Figure 7. Projected relative reliability of single-, double-, and triple-parity RAID.

RAID-5 today. It is again time to create a new RAID level to accommodate the realities of disk reliability, capacity, and throughput merely to maintain that same level of data protection.

Triple-Parity RAID

With RAID-6 increasingly unable to meet reliability requirements, there is an impending but not yet urgent need for triple-parity RAID. The addition of another level of parity mitigates increasing RAID rebuild times and occurrences of latent data errors. As shown in Figure 7, triple-parity RAID will address the shortcomings of RAID-6 for years (see the accompanying sidebar "A Classification for Triple-Parity RAID"). The reliability is largely independent of the specific implementation of triple-parity RAID; a general Reed-Solomon method suffices for our analysis.

A recurring theme in computer science is that algorithms can be specialized for small fixed values, but are then generalized to scale to an arbitrary value. A common belief in the computer industry had been that double-parity RAID was effectively that generalization, that it provided all the data reliability that would ever be needed. RAID-6 is inadequate, leading to the need for triple-parity RAID, but that, too, if current trends persist, will become insufficient. Not only is there a need for triple-parity RAID, but there's also a need for efficient algorithms that truly address the general case of RAID with an arbitrary number of parity devices.

Beyond RAID-5 and -6, what are the implications for RAID-1, simple two-way mirroring? RAID-1 can be viewed as a degenerate form of RAID-5, so even if bit error rates improve at the same rate as hard-drive capacities, the time to repair for RAID-1 could become debilitating. How secure would an administrator be running without redundancy for a week-long scrub? For the same reasons that make triple-parity RAID necessary where RAID-6 had sufficed, three-way mirroring will displace two-way mirroring for applications that require both high performance and strong data reliability. Indeed, four-way mirroring may not be far off, since even three-way mirroring is effectively a degenerate, but more reliable, form of RAID-6, and will be susceptible to the same failings.

Implications for RAID

While triple-parity RAID will be necessary, the steady penetration of flash solid-state storage could have a significant effect on the fate of disk drives. At one extreme, some have predicted the relegation of disk to a tape-like backup role as flash becomes cheap and reliable enough to act as a replacement for disk.[6] In that scenario, RAID is still necessary as even solid-state devices suffer catastrophic and partial failures, but the specific capacities, error rates, and throughputs for such devices could mean that triple-parity RAID is not required. Unfortunately, too little is known about the properties of devices that might flourish, and that scenario is too far in the future to obviate the need for triple-parity RAID.

At another extreme, the integration of flash into the storage hierarchy[8] could address high-performance needs though solid-state caching and buffering, thus decoupling system performance from that of the component hard drives. This could hasten current trends as hard-drive manufacturers would be able to increase capacity even more quickly, unhindered by performance requirements, while likely slowing the rate of throughput increases. Further, divorced from performance, RAID stripes could grow very wide to optimize for absolute capacity; this would reduce the reliability further with the same amount of parity protecting more data. In this scenario, the need for triple-parity RAID would be made all the more urgent by accelerating current trends.

If Kryder's Law continues to hold, the burden of correctness will increasingly shift from the hard-drive manufacturers to the RAID systems that integrate them. Today, RAID reconstruction times factor more into reliability calculations than ever before, and their contribution will increasingly dominate. Triple-parity RAID will soon be critical to provide sufficient reliability even in the face of exponential growth.

Acknowledgments

Many thanks to Dominic Kay for gathering the historical hard-drive data, and to Matt Ahrens, Daniel Leventhal, and Beverly Hodgson for their helpful reviews.

Related articles on queue.acm.org

Flash Storage Today
Adam Leventhal
http://queue.acm.org/detail.cfm?id=1413262

Hard Disk Drives: The Good, the Bad and the Ugly
Jon Elerath
http://queue.acm.org/detail.cfm?id=1317403

You Don't Know Jack about Disks
Dave Anderson
http://queue.acm.org/detail.cfm?id=864058

References

1. Berriman, E., Feresten, P., and Kung, S. NetApp RAID-DP: Dual-parity Raid-6 protection without compromise; http://www.mochadata.com/download/NetApp-raid-dp.pdf (2006).
2. Blaum, M., Brady, J., Bruck, J., and Menon, J. EVENODD: An optimal scheme for tolerating double disk failures in RAID architectures. In *Proceedings of the International Symposium on Computer Architecture* (1994), 245–254; http://portal.acm.org/citation.cfm?id=191995.192033.
3. Chen, P., Lee, E., Patterson, D., Gibson, G., and Katz, R. RAID: High-performance, reliable secondary storage. Technical Report CSD 93-778 (1993); http://portal.acm.org/citation.cfm?id=893811.
4. Corbett, P., English, B., Goel, A., Grcanac, T., Kleiman, S., Leong, J., and Sankar, S. Row-diagonal parity for double disk failure correction. In *Proceedings of the 3rd Usenix Conference on File and Storage Technologies* (2004), 1–14; http://portal.acm.org/citation.cfm?id=1096673.1096677.
5. Elerath, J. Hard-disk drives: The good, the bad, and the ugly. *Commun. ACM 52*, 6 (June 2009), 38–45; http://portal.acm.org/citation.cfm?id=1516046.1516059.
6. Gray, J. and Fitzgerald, B. Flash Disk Opportunity for Server-Applications. Microsoft Research; http://research.microsoft.com/en-us/um/people/gray/papers/FlashDiskPublic.doc (2007).
7. Hitz, D. 2006. Why "Double Protecting RAID" (RAID-DP) doesn't waste extra disk space; http://blogs.netapp.com/dave/2006/05/why_double_prot.html (2006).
8. Leventhal, A. Flash storage memory. *Commun. ACM 51*, 7 (July 2008), 47–51; http://portal.acm.org/citation.cfm?id=1364782.
9. Patterson, D., Gibson, G., and Katz, R. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of ACM SIGMOD International Conference on Management of Data* (1988), 109–116; http://portal.acm.org/citation.cfm?id=50214.
10. Plank, J. A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems. Technical Report UT-CS-96-332; http://portal.acm.org/citation.cfm?id=898928 (1996).
11. Walter, C. Kryder's Law. *Scientific American* (Aug. 2005); http://www.scientificamerican.com/article.cfm?id=kryders-law.

Adam Leventhal is a senior staff engineer and flash architect for Sun's Fishworks advanced product development team responsible for the Sun Storage 7000 series. He is one of the three authors of DTrace, for which he and his colleagues were named one of *InfoWorld*'s Innovators of 2005 and won top honors from the 2006 *Wall Street Journal*'s Innovation Awards.

DOI:10.1145/1629175.1629197

MapReduce complements DBMSs since databases are not designed for extract-transform-load tasks, a MapReduce specialty.

BY MICHAEL STONEBRAKER, DANIEL ABADI, DAVID J. DEWITT, SAM MADDEN, ERIK PAULSON, ANDREW PAVLO, AND ALEXANDER RASIN

MapReduce and Parallel DBMSs: Friends or Foes?

THE MAPREDUCE[7] (MR) PARADIGM has been hailed as a revolutionary new platform for large-scale, massively parallel data access.[16] Some proponents claim the extreme scalability of MR will relegate relational database management systems (DBMS) to the status of legacy technology. At least one enterprise, Facebook, has implemented a large data warehouse system using MR technology rather than a DBMS.[14]

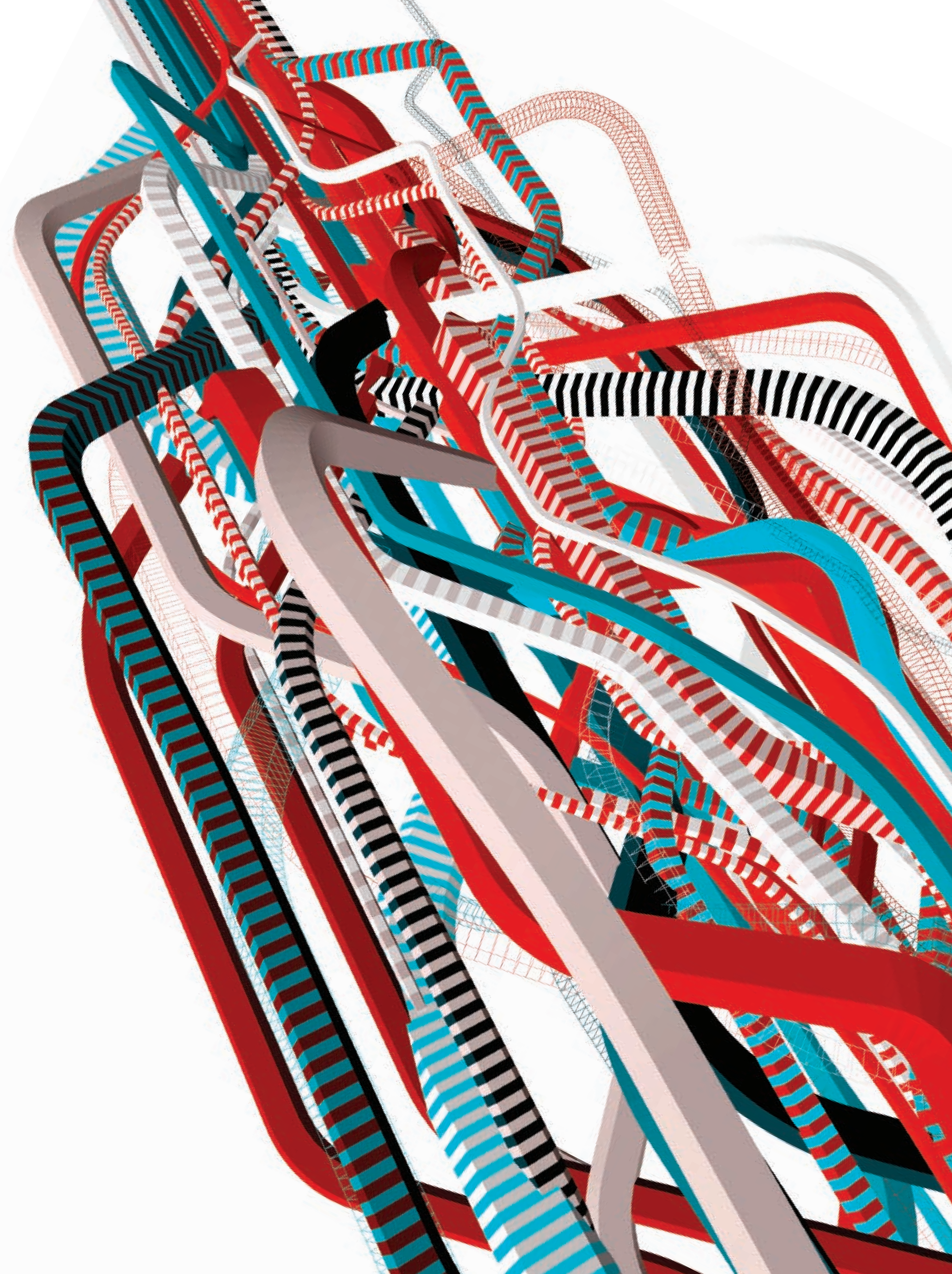
Here, we argue that using MR systems to perform tasks that are best suited for DBMSs yields less than satisfactory results,[17] concluding that MR is more like an extract-transform-load (ETL) system than a DBMS, as it quickly loads and processes large amounts of data in an ad hoc manner. As such, it complements DBMS technology rather than competes with it. We also discuss the differences in the architectural decisions of MR systems and database systems and provide insight into how the systems should complement one another.

The technology press has been focusing on the revolution of "cloud computing," a paradigm that entails the harnessing of large numbers of processors working in parallel to solve computing problems. In effect, this suggests constructing a data center by lining up a large number of low-end servers, rather than deploying a smaller set of high-end servers. Along with this interest in clusters has come a proliferation of tools for programming them. MR is one such tool, an attractive option to many because it provides a simple model through which users are able to express relatively sophisticated distributed programs.

Given the interest in the MR model both commercially and academically, it is natural to ask whether MR systems should replace parallel database systems. Parallel DBMSs were first available commercially nearly two decades ago, and, today, systems (from about a dozen vendors) are available. As robust, high-performance computing platforms, they provide a high-level programming environment that is inherently parallelizable. Although it might seem that MR and parallel DBMSs are different, it is possible to write almost any parallel-processing task as either a set of database queries or a set of MR jobs.

Our discussions with MR users lead us to conclude that the most common use case for MR is more like an ETL system. As such, it is complementary to DBMSs, not a competing technology, since databases are not designed to be good at ETL tasks. Here, we describe what we believe is the ideal use of MR technology and highlight the different MR and parallel DMBS markets.

ILLUSTRATION BY MARIUS WATZ



We recently conducted a benchmark study using a popular open-source MR implementation and two parallel DBMSs.[17] The results show that the DBMSs are substantially faster than the MR system once the data is loaded, but that loading the data takes considerably longer in the database systems. Here, we discuss the source of these performance differences, including the limiting architectural factors we perceive in the two classes of system, and conclude with lessons the MR and DBMS communities can learn from each other, along with future trends in large-scale data analysis.

Parallel Database Systems

In the mid-1980s the Teradata[20] and Gamma projects[9] pioneered a new architectural paradigm for parallel database systems based on a cluster of commodity computers called "shared-nothing nodes" (or separate CPU, memory, and disks) connected through a high-speed interconnect.[19] Every parallel database system built since then essentially uses the techniques first pioneered by these two projects: horizontal partitioning of relational tables, along with the partitioned execution of SQL queries.

The idea behind horizontal partitioning is to distribute the rows of a relational table across the nodes of the cluster so they can be processed in parallel. For example, partitioning a 10-million-row table across a cluster of 50 nodes, each with four disks, would place 50,000 rows on each of the 200 disks. Most parallel database systems offer a variety of partitioning strategies, including hash, range, and round-robin partitioning.[8] Under a hash-partitioning physical layout, as each row is loaded, a hash function is applied to one or more attributes of each row to determine the target node and disk where the row should be stored.

The use of horizontal partitioning of tables across the nodes of a cluster is critical to obtaining scalable performance of SQL queries[8] and leads naturally to the concept of partitioned execution of the SQL operators: selection, aggregation, join, projection, and update. As an example how data partitioning is used in a parallel DBMS, consider the following SQL query:

```
SELECT custId, amount FROM Sales
 WHERE date BETWEEN
 "12/1/2009" AND "12/25/2009";
```

With the Sales table horizontally partitioned across the nodes of the cluster, this query can be trivially executed in parallel by executing a SELECT operator against the Sales records with the specified date predicate on each node of the cluster. The intermediate results from each node are then sent to a single node that performs a MERGE operation in order to return the final result to the application program that issued the query.

Suppose we would like to know the total sales amount for each custId within the same date range. This is done through the following query:

```
SELECT custId, SUM(amount)
FROM Sales
 WHERE date BETWEEN
 "12/1/2009" AND "12/25/2009"
 GROUP BY custId;
```

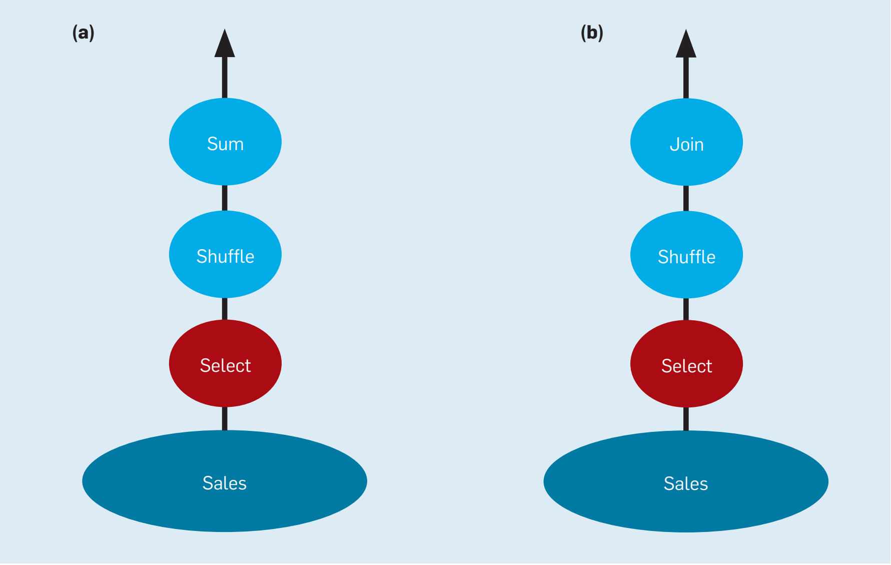
If the Sales table is round-robin partitioned across the nodes in the cluster, then the rows corresponding to any single customer will be spread across multiple nodes. The DBMS compiles this query into the three-operator pipeline in Figure(a), then executes the query plan on all the nodes in the cluster in parallel. Each SELECT operator scans the fragment of the Sales table stored at that node. Any rows satisfying the date predicate are passed to a SHUFFLE operator that dynamically repartitions the rows; this is typically done by applying a hash function on the value of the custId attribute of each row to map them to a particular node. Since the same hash function is used for the SHUFFLE operation on all nodes, rows for the same customer are routed to the single node where they are aggregated to compute the final total for each customer.

As a final example of how SQL is parallelized using data partitioning, consider the following query for finding the names and email addresses of customers who purchased an item costing more than $1,000 during the holiday shopping period:

```
SELECT C.name, C.email FROM
Customers C, Sales S
 WHERE C.custId = S.custId
 AND S.amount > 1000
  AND S.date BETWEEN
  "12/1/2009" AND
  "12/25/2009";
```

Assume again that the Sales table is round-robin partitioned, but we now hash-partition the Customers table on the Customer.custId attribute. The DBMS compiles this query into the operator pipeline in Figure(b) that is executed in parallel at all nodes in the cluster. Each SELECT operator scans

Parallel database query execution plans. (a) Example operator pipeline for calculating a single-table aggregate. (b) Example operator pipeline for performing a joining on two partitioned tables.



its fragment of the Sales table looking for rows that satisfy the predicate

```
S.amount > 1000 and S.date
BETWEEN "12/1/2009" and
"12/25/2009."
```

Qualifying rows are pipelined into a shuffle operator that repartitions its input rows by hashing on the Sales.custId attribute. By using the same hash function that was used when loading rows of the Customer table (hash partitioned on the Customer.custId attribute), the shuffle operators route each qualifying Sales row to the node where the matching Customer tuple is stored, allowing the join operator (C.custId = S.custId) to execute in parallel on all the nodes.

Another key benefit of parallel DBMSs is that the system automatically manages the various alternative partitioning strategies for the tables involved in the query. For example, if Sales and Customers are each hash-partitioned on their custId attribute, the query optimizer will recognize that the two tables are both hash-partitioned on the joining attributes and omit the shuffle operator from the compiled query plan. Likewise, if both tables are round-robin partitioned, then the optimizer will insert shuffle operators for both tables so tuples that join with one another end up on the same node. All this happens transparently to the user and to application programs.

Many commercial implementations are available, including Teradata, Netezza, DataAllegro (Microsoft), ParAccel, Greenplum, Aster, Vertica, and DB2. All run on shared-nothing clusters of nodes, with tables horizontally partitioned over them.

Mapping Parallel DBMSs onto MapReduce

An attractive quality of the MR programming model is simplicity; an MR program consists of only two functions—Map and Reduce—written by a user to process key/value data pairs.[7] The input data set is stored in a collection of partitions in a distributed file system deployed on each node in the cluster. The program is then injected into a distributed-processing framework and executed in a manner to be described. The MR model was first popularized by Google in 2004, and, today, numerous open source and commercial implementations are available. The most popular MR system is Hadoop, an open-source project under development by Yahoo! and the Apache Software Foundation (http://hadoop.apache.org/).

The semantics of the MR model are not unique, as the filtering and transformation of individual data items (tuples in tables) can be executed by a modern parallel DBMS using SQL. For Map operations not easily expressed in SQL, many DBMSs support user-defined functions[18]; UDF extensibility provides the equivalent functionality of a Map operation. SQL aggregates augmented with UDFs and user-defined aggregates provide DBMS users the same MR-style reduce functionality. Lastly, the reshuffle that occurs between the Map and Reduce tasks in MR is equivalent to a GROUP BY operation in SQL. Given this, parallel DBMSs provide the same computing model as MR, with the added benefit of using a declarative language (SQL).

The linear scalability of parallel DBMSs has been widely touted for two decades[10]; that is, as nodes are added to an installation, the database size can be increased proportionally while maintaining constant response times. Several production databases in the multi-petabyte range are run by very large customers operating on clusters of order 100 nodes.[13] The people who manage these systems do not report the need for additional parallelism. Thus, parallel DBMSs offer great scalability over the range of nodes that customers desire. There is no reason why scalability cannot be increased dramatically to the levels reported by Jeffrey Dean and Sanjay Ghemawat,[7] assuming there is customer demand.

Possible Applications

Even though parallel DBMSs are able to execute the same semantic workload as MR, several application classes are routinely mentioned as possible use cases in which the MR model might be a better choice than a DBMS. We now explore five of these scenarios, discussing the ramifications of using one class of system over another:

ETL and "read once" data sets. The canonical use of MR is characterized by the following template of five operations:

- Read logs of information from several different sources;
- Parse and clean the log data;
- Perform complex transformations (such as "sessionalization");
- Decide what attribute data to store; and
- Load the information into a DBMS or other storage engine.

These steps are analogous to the extract, transform, and load phases in ETL systems; the MR system is essentially "cooking" raw data into useful information that is consumed by another storage system. Hence, an MR system can be considered a general-purpose parallel ETL system.

For parallel DBMSs, many products perform ETL, including Ascential, Informatica, Jaspersoft, and Talend. The market is large, as almost all major enterprises use ETL systems to load large quantities of data into data warehouses. One reason for this symbiotic relationship is the clear distinction as to what each class of system provides to users: DBMSs do not try to do ETL, and ETL systems do not try to do DBMS services. An ETL system is typically upstream from a DBMS, as the load phase usually feeds data directly into a DBMS.

Complex analytics. In many data-mining and data-clustering applications, the program must make multiple passes over the data. Such applications cannot be structured as single SQL aggregate queries, requiring instead a complex dataflow program where the output of one part of the application is the input of another. MR is a good candidate for such applications.

Semi-structured data. Unlike a DBMS, MR systems do not require users to define a schema for their data. Thus, MR-style systems easily store and process what is known as "semi-structured" data. In our experience, such data often looks like key-value pairs, where the number of attributes present in any given record varies; this style of data is typical of Web traffic logs derived from disparate sources.

With a relational DBMS, one way to model such data is to use a wide table with many attributes to accommodate multiple record types. Each unrequired attribute uses NULLs for the values that are not present for a given record. Row-based DBMSs generally have trouble with the tables, often suffering poor performance. On the other hand, column-based DBMSs (such as Vertica) mitigate the problem by reading only the relevant attributes for any query and automatically suppressing the NULL values.[3] These techniques have been shown to provide good performance on RDF data sets,[2] and we expect the same would be true for simpler key-value data.

To the extent that semistructured data fits the "cooking" paradigm discussed earlier (that is, the data is prepared for loading into a back-end data-processing system), then MR-style systems are a good fit. If the semistructured data set is primarily for analytical queries, we expect a parallel column store to be a better solution.

Quick-and-dirty analyses. One disappointing aspect of many current parallel DBMSs is that they are difficult to install and configure properly, as users are often faced with a myriad of tuning parameters that must be set correctly for the system to operate effectively. From our experiences with installing two commercial parallel systems, an open-source MR implementation provides the best "out-of-the-box" experience[17]; that is, we were able to get the MR system up and running queries significantly faster than either of the DBMSs. In fact, it was not until we received expert support from one of the vendors that we were able to get one particular DBMS to run queries that completed in minutes, rather than hours or days.

Once a DBMS is up and running properly, programmers must still write a schema for their data (if one does not already exist), then load the data set into the system. This process takes considerably longer in a DBMS than in an MR system, because the DBMS must parse and verify each datum in the tuples. In contrast, the default (therefore most common) way for MR programmers to load their data is to just copy it into the MR system's underlying distributed block-based storage system.

If a programmer must perform some one-off analysis on transient data, then the MR model's quick startup time is clearly preferable. On the other hand, professional DBMS programmers and administrators are more willing to pay in terms of longer learning curves and startup times, because the performance gains from faster queries offset the upfront costs.

Limited-budget operations. Another strength of MR systems is that most are open source projects available for free. DBMSs, and in particular parallel DBMSs, are expensive; though there are good single-node open source solutions, to the best of our knowledge, there are no robust, community-supported parallel DBMSs. Though enterprise users with heavy demand and big budgets might be willing to pay for a commercial system and all the tools, support, and service agreements those systems provide, users with more modest budgets or requirements find open source systems more attractive. The database community has missed an opportunity by not providing a more complete parallel, open source solution.

Powerful tools. MR systems are fundamentally powerful tools for ETL-style applications and for complex analytics. Additionally, they are popular for "quick and dirty" analyses and for users with limited budgets. On the other hand, if the application is query-intensive, whether semistructured or rigidly structured, then a DBMS is probably the better choice. In the next section, we discuss results from use cases that demonstrate this performance superiority; the processing tasks range from those MR systems ought to be good at to those that are quite complex queries.

DBMS "Sweet Spot"

To demonstrate the performance trade-offs between parallel DBMSs and MR systems, we published a benchmark comparing two parallel DBMSs to the Hadoop MR framework on a variety of tasks.[17] We wished to discover the performance envelope of each approach when applied to areas inside and outside their target application space. We used two database systems: Vertica, a commercial column-store relational database, and DBMS-X, a row-based database from a large commercial vendor. Our benchmark study included a simple benchmark presented in the original MR paper from Google,[7] as well as four other analytical tasks of increasing complexity we think are common processing tasks that could be done using either class of systems. We ran all experiments on a 100-node shared-nothing cluster at the University of Wisconsin-Madison. The full paper[17] includes the complete results and discussion from all our experiments, including load times; here, we provide a summary of the most interesting results. (The source code for the benchmark study is available at http://database.cs.brown.edu/projects/mapreduce-vs-dbms/.)

Hadoop is by far the most popular publicly available version of the MR framework (the Google version might be faster but is not available to us), and DBMS-X and Vertica are popular row- and column-store parallel database systems, respectively.

In the time since publication of Pavlo et al.[17] we have continued to tune all three systems. Moreover, we have received many suggestions from the Hadoop community on ways to improve performance. We have tried them all, and the results here (as of August 2009) represent the best we can do with a substantial amount of expert help on all three systems. In fact, the time we've spent tuning Hadoop has now exceeded the time we spent on either of the other systems. Though Hadoop offers a good out-of-the-box experience, tuning it to obtain maximum performance was an arduous task. Obviously, performance is a moving target, as new releases of all three products occur regularly

Original MR Grep task. Our first benchmark experiment is the "Grep task'" from the original MR paper, which described it as "representative of a large subset of the real programs written by users of MapReduce."[7] For the task, each system must scan through a data set of 100B records looking for a three-character pattern. Each record consists of a unique key in the first 10B, followed by a 90B random value. The search pattern is found only in the last 90B once in every 10,000 records. We use a 1TB data set spread over the 100 nodes (10GB/node). The data set consists of 10 billion records, each 100B. Since this is essentially a sequential search of the data set looking for the pattern, it provides a simple measurement of how

Benchmark performance on a 100-node cluster.

	Hadoop	DBMS-X	Vertica	Hadoop/DBMS-X	Hadoop/Vertica
Grep	284s	194s	108x	1.5x	2.6x
Web Log	1,146s	740s	268s	1.6x	4.3x
Join	1,158s	32s	55s	36.3x	21.0x

quickly a software system can scan through a large collection of records. The task cannot take advantage of any sorting or indexing and is easy to specify in both MR and SQL. Therefore, one would expect a lower-level interface (such as Hadoop) running directly on top of the file system (HDFS) to execute faster than the more heavyweight DBMSs.

However, the execution times in the table here show a surprising result: The database systems are about two times faster than Hadoop. We explain some of the reasons for this conclusion in the section on architectural differences.

Web log task. The second task is a conventional SQL aggregation with a GROUP BY clause on a table of user visits in a Web server log. Such data is fairly typical of Web logs, and the query is commonly used in traffic analytics. For this experiment, we used a 2TB data set consisting of 155 million records spread over the 100 nodes (20GB/node). Each system must calculate the total ad revenue generated for each visited IP address from the logs. Like the previous task, the records must all be read, and thus there is no indexing opportunity for the DBMSs. One might think that Hadoop would excel at this task since it is a straightforward calculation, but the results in the table show that Hadoop is beaten by the databases by a larger margin than in the Grep task.

Join task. The final task we discuss here is a fairly complex join operation over two tables requiring an additional aggregation and filtering operation. The user-visit data set from the previous task is joined with an additional 100GB table of PageRank values for 18 million URLs (1GB/node). The join task consists of two subtasks that perform a complex calculation on the two data sets. In the first part of the task, each system must find the IP address that generated the most revenue within a particular date range in the user visits. Once these intermediate records are generated, the system must then calculate the average PageRank of all pages visited during this interval.

DBMSs ought to be good at analytical queries involving complex join operations (see the table). The DBMSs are a factor of 36 and 21 respectively faster than Hadoop. In general, query times for a typical user task fall somewhere in between these extremes. In the next section, we explore the reasons for these results.

Architectural Differences

The performance differences between Hadoop and the DBMSs can be explained by a variety of factors. Before delving into the details, we should say these differences result from implementation choices made by the two classes of system, not from any fundamental difference in the two models. For example, the MR processing model is independent of the underlying storage system, so data could theoretically be massaged, indexed, compressed, and carefully laid out on storage during a load phase, just like a DBMS. Hence, the goal of our study was to compare the real-life differences in performance of representative realizations of the two models.

Repetitive record parsing. One contributing factor for Hadoop's slower performance is that the default configuration of Hadoop stores data in the accompanying distributed file system (HDFS), in the same textual format in which the data was generated. Consequently, this default storage method places the burden of parsing the fields of each record on user code. This parsing task requires each Map and Reduce task repeatedly parse and convert string fields into the appropriate type. Hadoop provides the ability to store data as key/value pairs as serialized tuples called SequenceFiles, but despite this ability it still requires user code to parse the value portion of the record if it contains multiple attributes. Thus, we found that using SequenceFiles without compression consistently yielded slower performance on our benchmark. Note that using SequenceFiles without compression was but one of the tactics for possibly improving Hadoop's performance suggested by the MR community.

In contrast to repetitive parsing in MR, records are parsed by DBMSs when the data is initially loaded. This initial parsing step allows the DBMSs storage manager to carefully lay out records in storage such that attributes can be directly addressed at runtime in their most efficient storage representation. As such, there is no record interpretation performed during query execution in parallel DBMSs.

There is nothing fundamental about the MR model that says data cannot be parsed in advance and stored in optimized data structures (that is, trading off some load time for increased runtime performance). For example, data could be stored in the underlying file system using Protocol Buffers (http://code.google.com/p/protobuf/), Google's platform-neutral, extensible mechanism for serializing structured data; this option is not available in Hadoop. Alternatively, one could move the data outside the MR framework into a relational DBMS at each node, thereby replacing the HDFS storage layer with DBMS-style optimized storage for structured data.[4]

There may be ways to improve the Hadoop system by taking advantage of these ideas. Hence, parsing overhead is a problem, and SequenceFiles are not an effective solution. The problem should be viewed as a signpost for guiding future development.

Compression. We found that enabling data compression in the DBMSs delivered a significant performance gain. The benchmark results show that using compression in Vertica and DBMS-X on these workloads improves performance by a factor of two to four. On the other hand, Hadoop often executed slower when we used compression on its input files; at most, compression improved performance by 15%; the benchmark results in Dean and Ghemawat[7] also did not use compression.

It is unclear to us why this improvement was insignficant, as essentially all commercial SQL data warehouses use compression to improve performance. We postulate that commercial DBMSs use carefully tuned compression algorithms to ensure that the cost of decompressing tuples does not offset the performance gains from the reduced I/O cost of reading compressed data. For example, we have found that on modern processors standard Unix implementations of gzip and bzip are often too slow to provide any benefit.

Pipelining. All parallel DBMSs operate by creating a query plan that is distributed to the appropriate nodes at execution time. When one operator in this plan must send data to the next operator, regardless of whether that operator is running on the same or a different node, the qualifying data is "pushed" by the first operator to the second operator. Hence, data is streamed from producer to consumer; the intermediate data is never written to disk; the resulting "back-pressure" in the runtime system will stall the producer before it has a chance to overrun the consumer. This streaming technique differs from the approach taken in MR systems, where the producer writes the intermediate results to local data structures, and the consumer subsequently "pulls" the data. These data structures are often quite large, so the system must write them out to disk, introducing a potential bottleneck. Though writing data structures to disk gives Hadoop a convenient way to checkpoint the output of intermediate map jobs, thereby improving fault tolerance, we found from our investigation that it adds significant performance overhead.

Scheduling. In a parallel DBMS, each node knows exactly what it must do and when it must do it according to the distributed query plan. Because the operations are known in advance, the system is able to optimize the execution plan to minimize data transmission between nodes. In contrast, each task in an MR system is scheduled on processing nodes one storage block at a time. Such runtime work scheduling at a granularity of storage blocks is much more expensive than the DBMS compile-time scheduling. The former has the advantage, as some have argued,[4] of allowing the MR scheduler to adapt to workload skew and performance differences between nodes.

The commercial DBMS products must move toward one-button installs, automatic tuning that works correctly, better Web sites with example code, better query generators, and better documentation.

Column-oriented storage. In a column store-based database (such as Vertica), the system reads only the attributes necessary for solving the user query. This limited need for reading data represents a considerable performance advantage over traditional, row-stored databases, where the system reads all attributes off the disk. DBMS-X and Hadoop/HDFS are both essentially row stores, while Vertica is a column store, giving Vertica a significant advantage over the other two systems in our Web log benchmark task.

Discussion. The Hadoop community will presumably fix the compression problem in a future release. Furthermore, some of the other performance advantages of parallel databases (such as column-storage and operating directly on compressed data) can be implemented in an MR system with user code. Also, other implementations of the MR framework (such as Google's proprietary implementation) may well have a different performance envelope. The scheduling mechanism and pull model of data transmission are fundamental to the MR block-level fault-tolerance model and thus unlikely to be changed.

Meanwhile, DBMSs offer transaction-level fault tolerance. DBMS researchers often point out that as databases get bigger and the number of nodes increases, the need for finer-granularity fault tolerance increases as well. DBMSs readily adapt to this need by marking one or more operators in a query plan as "restart operators." The runtime system saves the result of these operators to disk, facilitating "operator level" restart. Any number of operators can be so marked, allowing the granularity of restart to be tuned. Such a mechanism is easily integrated into the efficient query execution framework of DBMSs while allowing variable granularity restart. We know of at least two separate research groups, one at the University of Washington, the other at the University of California, Berkeley, that are exploring the trade-off between runtime overhead and the amount of work lost when a failure occurs.

We generally expect ETL and complex analytics to be amenable to MR systems and query-intensive workloads to be run by DBMSs. Hence, we expect the best solution is to interface an MR framework to a DBMS so MR can do complex analytics, and interface to a DBMS to do embedded queries. HadoopDB,[4] Hive,[21] Aster, Greenplum, Cloudera, and Vertica all have commercially available products or prototypes in this "hybrid" category.

Learning from Each Other

What can MR learn from DBMSs? MR advocates should learn from parallel DBMS the technologies and techniques for efficient query parallel execution. Engineers should stand on the shoulders of those who went before, rather than on their toes. There are many good ideas in parallel DBMS executors that MR system developers would be wise to adopt.

We also feel that higher-level languages are invariably a good idea for any data-processing system. Relational DBMSs have been fabulously successful in pushing programmers to a higher, more-productive level of abstraction, where they simply state what they want from the system, rather than writing an algorithm for how to get what they want from the system. In our benchmark study, we found that writing the SQL code for each task was substantially easier than writing MR code.

Efforts to build higher-level interfaces on top of MR/Hadoop should be accelerated; we applaud Hive,[21] Pig,[15] Scope,[6] Dryad/Linq,[12] and other projects that point the way in this area.

What can DBMSs learn from MR? The out-of-the-box experience for most DBMSs is less than ideal for being able to quickly set up and begin running queries. The commercial DBMS products must move toward one-button installs, automatic tuning that works correctly, better Web sites with example code, better query generators, and better documentation.

Most database systems cannot deal with tables stored in the file system (in situ data). Consider the case where a DBMS is used to store a very large data set on which a user wishes to perform analysis in conjunction with a smaller, private data set. In order to access the larger data set, the user must first load the data into the DBMS. Unless the user plans to run many analyses, it is preferable to simply point the DBMS at data on the local disk without a load phase. There is no good reason DBMSs cannot deal with in situ data. Though some database systems (such as PostgreSQL, DB2, and SQL Server) have capabilities in this area, further flexibility is needed.

Conclusion

Most of the architectural differences discussed here are the result of the different focuses of the two classes of system. Parallel DBMSs excel at efficient querying of large data sets; MR-style systems excel at complex analytics and ETL tasks. Neither is good at what the other does well. Hence, the two technologies are complementary, and we expect MR-style systems performing ETL to live directly upstream from DBMSs.

Many complex analytical problems require the capabilities provided by both systems. This requirement motivates the need for interfaces between MR systems and DBMSs that allow each system to do what it is good at. The result is a much more efficient overall system than if one tries to do the entire application in either system. That is, "smart software" is always a good idea.

Acknowledgment

This work is supported in part by National Science Foundation grants CRI-0707437, CluE-0844013, and CluE-0844480.

References

1. Abadi, D.J., Madden, S.R., and Hachem, N. Column-stores vs. row-stores: How different are they really? In *Proceedings of the SIGMOD Conference on Management of Data*. ACM Press, New York, 2008.
2. Abadi, D.J., Marcus, A., Madden, S.R., and Hollenbach, K. Scalable semantic Web data management using vertical partitioning. In *Proceedings of the 33rd International Conference on Very Large Databases*, 2007.
3. Abadi, D.J. Column-stores for wide and sparse data. In *Proceedings of the Conference on Innovative Data Systems Research*, 2007.
4. Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D.J., Silberschatz, A., and Rasin, A. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads. In *Proceedings of the Conference on Very Large Databases*, 2009.
5. Boral, H. et al. Prototyping Bubba, a highly parallel database system. *IEEE Transactions on Knowledge and Data Engineering 2*, 1 (Mar. 1990), 4–24.
6. Chaiken, R., Jenkins, B., Larson, P., Ramsey, B., Shakib, D., Weaver, S., and Zhou, J. SCOPE: Easy and efficient parallel processing of massive data sets. In *Proceedings of the Conference on Very Large Databases*, 2008.
7. Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. In *Proceedings of the Sixth Conference on Operating System Design and Implementation* (Berkeley, CA, 2004).
8. DeWitt, D.J. and Gray, J. Parallel database systems: The future of high-performance database systems. *Commun. ACM 35*, 6 (June 1992), 85–98.
9. DeWitt, D.J., Gerber, R.H., Graefe, G., Heytens, M.L., Kumar, K.B., and Muralikrishna, M. GAMMA: A high-performance dataflow database machine. In *Proceedings of the 12th International Conference on Very Large Databases*. Morgan Kaufmann Publishers, Inc., 1986, 228–237.
10. Englert, S., Gray, J., Kocher, T., and Shah, P. A benchmark of NonStop SQL Release 2 demonstrating near-linear speedup and scaleup on large databases. *Sigmetrics Performance Evaluation Review 18*, 1 (1990), 1990, 245–246.
11. Fushimi, S., Kitsuregawa, M., and Tanaka, H. An overview of the system software of a parallel relational database machine. In *Proceedings of the 12th International Conference on Very Large Databases*, Morgan Kaufmann Publishers, Inc., 1986, 209–219.
12. Isard, M., Budiu, M., Yu, Y., Birrell, A., and Fetterly, D. Dryad: Distributed data-parallel programs from sequential building blocks. *SIGOPS Operating System Review 41*, 3 (2007), 59–72.
13. Monash, C. Some very, very, very large data warehouses. In NetworkWorld.com community blog, May 12, 2009; http://www.networkworld.com/community/node/41777.
14. Monash, C. Cloudera presents the MapReduce bull case. In DBMS2.com blog, Apr. 15, 2009; http://www.dbms2.com/2009/04/15/cloudera-presents-the-mapreduce-bull-case/.
15. Olston, C., Reed, B., Srivastava, U., Kumar, R., and Tomkins, A. Pig Latin: A not-so-foreign language for data processing. In *Proceedings of the SIGMOD Conference*. ACM Press, New York, 2008, 1099–1110.
16. Patterson, D.A. Technical perspective: The data center is the computer. *Commun. ACM 51*, 1 (Jan. 2008), 105.
17. Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S.R., and Stonebraker, M. A comparison of approaches to large-scale data analysis. In *Proceedings of the 35th SIGMOD International Conference on Management of Data*. ACM Press, New York, 2009, 165–178.
18. Stonebraker, M. and Rowe, L. The design of Postgres. In *Proceedings of the SIGMOD Conference*, 1986, 340–355.
19. Stonebraker, M. The case for shared nothing. *Data Engineering 9* (Mar. 1986), 4–9.
20. Teradata Corp. *Database Computer System Manual, Release 1.3*. Los Angeles, CA, Feb. 1985.
21. Thusoo, A. et al. Hive: A warehousing solution over a Map-Reduce framework. In *Proceedings of the Conference on Very Large Databases*, 2009, 1626–1629.

Michael Stonebraker (stonebraker@csail.mit.edu) is an adjunct professor in the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, Cambridge, MA.

Daniel J. Abadi (dna@cs.yale.edu) is an assistant professor in the Department of Computer Science at Yale University, New Haven, CT.

David J. DeWitt (dewitt@microsoft.com) is a technical fellow in the Jim Gray Systems Lab at Microsoft Inc., Madison, WI.

Samuel Madden (madden@csail.mit.edu) is a professor in the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, Cambridge, MA.

Erik Paulson (epaulson@cs.wisc.edu) is a Ph.D. candidate in the Department of Computer Sciences at the University of Wisconsin-Madison, Madison, WI.

Andrew Pavlo (pavlo@cs.brown.edu) is a Ph.D. candidate in the Department of Computer Science at Brown University, Providence, RI.

Alexander Rasin (alexr@cs.brown.edu) is a Ph.D. candidate in the Department of Computer Science at Brown University, Providence, RI.

DOI:10.1145/1629175.1629198

MapReduce advantages over parallel databases include storage-system independence and fine-grain fault tolerance for large jobs.

BY JEFFREY DEAN AND SANJAY GHEMAWAT

MapReduce: A Flexible Data Processing Tool

MAPREDUCE IS A programming model for processing and generating large data sets.[4] Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs and a reduce function that merges all intermediate values associated with the same intermediate key. We built a system around this programming model in 2003 to simplify construction of the inverted index for handling searches at Google.com. Since then, more than 10,000 distinct programs have been implemented using MapReduce at Google, including algorithms for large-scale graph processing, text processing, machine learning, and statistical machine translation. The Hadoop open source implementation of MapReduce has been used extensively outside of Google by a number of organizations.[10,11]

To help illustrate the MapReduce programming model, consider the problem of counting the number of occurrences of each word in a large collection of documents. The user would write code like the following pseudocode:

```
map(String key, String value):
  // key: document name
  // value: document contents
  for each word w in value:
    EmitIntermediate(w, “1”);

reduce(String key, Iterator values):
  // key: a word
  // values: a list of counts
  int result = 0;
  for each v in values:
    result += ParseInt(v);
  Emit(AsString(result));
```

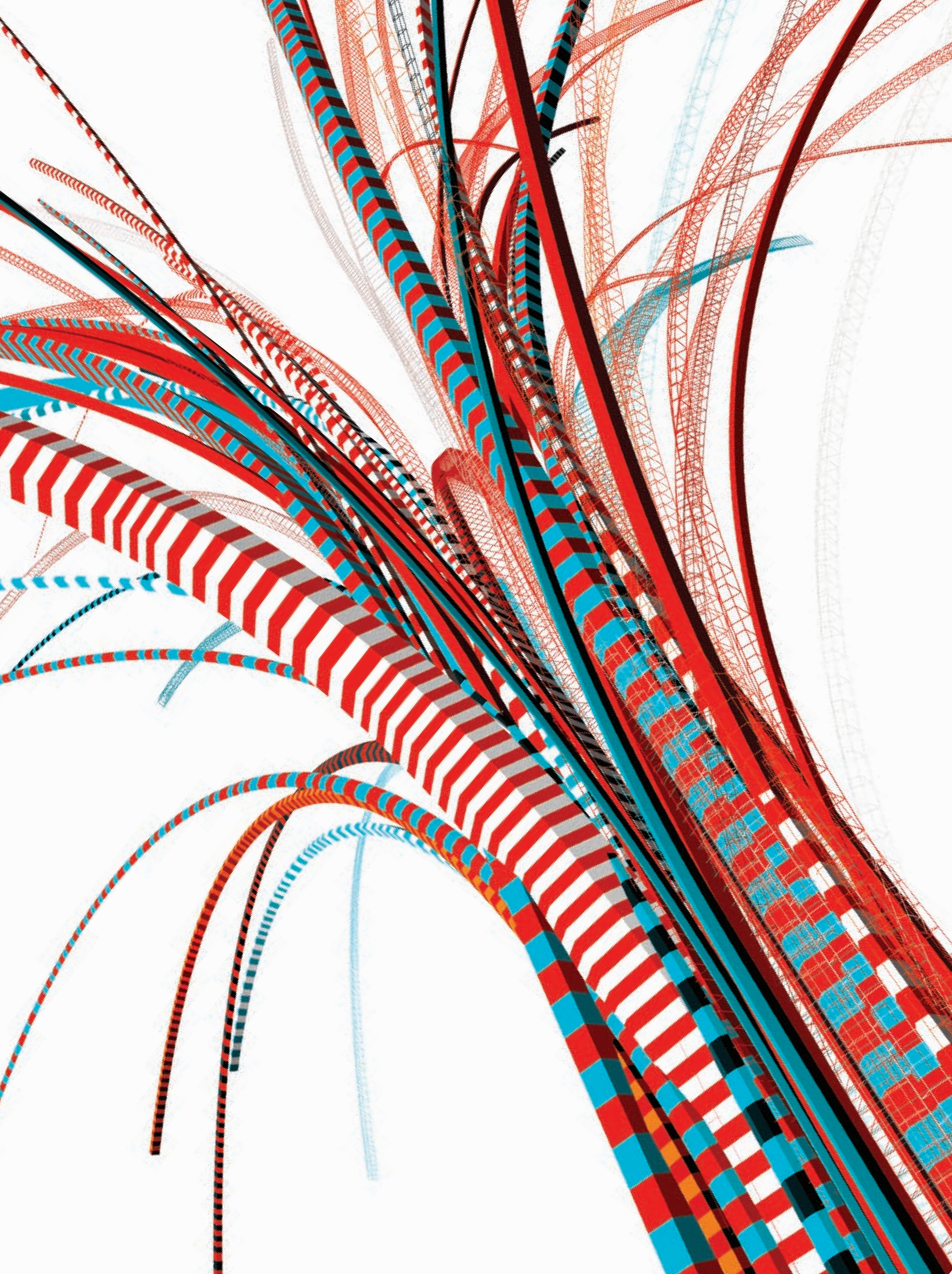
The map function emits each word plus an associated count of occurrences (just `1' in this simple example). The reduce function sums together all counts emitted for a particular word.

MapReduce automatically parallelizes and executes the program on a large cluster of commodity machines. The runtime system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing required inter-machine communication. MapReduce allows programmers with no experience with parallel and distributed systems to easily utilize the resources of a large distributed system. A typical MapReduce computation processes many terabytes of data on hundreds or thousands of machines. Programmers find the system easy to use, and more than 100,000 MapReduce jobs are executed on Google's clusters every day.

Compared to Parallel Databases

The query languages built into parallel database systems are also used to

ILLUSTRATION BY MARIUS WATZ



express the type of computations supported by MapReduce. A 2009 paper by Andrew Pavlo et al. (referred to here as the "comparison paper"[13]) compared the performance of MapReduce and parallel databases. It evaluated the open source Hadoop implementation[10] of the MapReduce programming model, DBMS-X (an unidentified commercial database system), and Vertica (a column-store database system from a company co-founded by one of the authors of the comparison paper). Earlier blog posts by some of the paper's authors characterized MapReduce as "a major step backwards."[5,6] In this article, we address several misconceptions about MapReduce in these three publications:

- MapReduce cannot use indices and implies a full scan of all input data;
- MapReduce input and outputs are always simple files in a file system; and
- MapReduce requires the use of inefficient textual data formats.

We also discuss other important issues:

- MapReduce is storage-system independent and can process data without first requiring it to be loaded into a database. In many cases, it is possible to run 50 or more separate MapReduce analyses in complete passes over the data before it is possible to load the data into a database and complete a single analysis;
- Complicated transformations are often easier to express in MapReduce than in SQL; and
- Many conclusions in the comparison paper were based on implementation and evaluation shortcomings not fundamental to the MapReduce model; we discuss these shortcomings later in this article.

We encourage readers to read the original MapReduce paper[4] and the comparison paper[13] for more context.

Heterogenous Systems

Many production environments contain a mix of storage systems. Customer data may be stored in a relational database, and user requests may be logged to a file system. Furthermore, as such environments evolve, data may migrate to new storage systems. MapReduce provides a simple model for analyzing data in such heterogenous systems. End users can extend MapReduce to support a new storage system by defining simple reader and writer implementations that operate on the storage system. Examples of supported storage systems are files stored in distributed file systems,[7] database query results,[2,9] data stored in Bigtable,[3] and structured input files (such as B-trees). A single MapReduce operation easily processes and combines data from a variety of storage systems.

Now consider a system in which a parallel DBMS is used to perform all data analysis. The input to such analysis must first be copied into the parallel DBMS. This loading phase is inconvenient. It may also be unacceptably slow, especially if the data will be analyzed only once or twice after being loaded. For example, consider a batch-oriented Web-crawling-and-indexing system that fetches a set of Web pages and generates an inverted index. It seems awkward and inefficient to load the set of fetched pages into a database just so they can be read through once to generate an inverted index. Even if the cost of loading the input into a parallel DBMS is acceptable, we still need an appropriate loading tool. Here is another place MapReduce can be used; instead of writing a custom loader with its own ad hoc parallelization and fault-tolerance support, a simple MapReduce program can be written to load the data into the parallel DBMS.

Indices

The comparison paper incorrectly said that MapReduce cannot take advantage of pregenerated indices, leading to skewed benchmark results in the paper. For example, consider a large data set partitioned into a collection of nondistributed databases, perhaps using a hash function. An index can be added to each database, and the result of running a database query using this index can be used as an input to MapReduce. If the data is stored in D database partitions, we will run D database queries that will become the D inputs to the MapReduce execution. Indeed, some of the authors of Pavlo et al. have pursued this approach in their more recent work.[11]

Another example of the use of indices is a MapReduce that reads from Bigtable. If the data needed maps to a sub-range of the Bigtable row space, we would need to read only that sub-range instead of scanning the entire Bigtable. Furthermore, like Vertica and other column-store databases, we will read data only from the columns needed for this analysis, since Bigtable can store data segregated by columns.

Yet another example is the processing of log data within a certain date range; see the Join task discussion in the comparison paper, where the Hadoop benchmark reads through 155 million records to process the 134,000 records that fall within the date range of interest. Nearly every logging system we are familiar with rolls over to a new log file periodically and embeds the rollover time in the name of each log file. Therefore, we can easily run a MapReduce operation over just the log files that may potentially overlap the specified date range, instead of reading all log files.

Complex Functions

Map and Reduce functions are often fairly simple and have straightforward SQL equivalents. However, in many cases, especially for Map functions, the function is too complicated to be expressed easily in a SQL query, as in the following examples:

- Extracting the set of outgoing links from a collection of HTML documents and aggregating by target document;
- Stitching together overlapping satellite images to remove seams and to select high-quality imagery for Google Earth;
- Generating a collection of inverted index files using a compression scheme tuned for efficient support of Google search queries;
- Processing all road segments in the world and rendering map tile images that display these segments for Google Maps; and
- Fault-tolerant parallel execution of programs written in higher-level languages (such as Sawzall[14] and Pig Latin[12]) across a collection of input data.

Conceptually, such user defined functions (UDFs) can be combined with SQL queries, but the experience reported in the comparison paper indicates that UDF support is either buggy (in DBMS-X) or missing (in Vertica). These concerns may go away over the long term, but for now, MapReduce is a better framework for doing more complicated tasks (such as those listed earlier) than the selection and aggregation that are SQL's forte.

Structured Data and Schemas

Pavlo et al. did raise a good point that schemas are helpful in allowing multiple applications to share the same data. For example, consider the following schema from the comparison paper:

```
   CREATE TABLE Rankings (
      pageURL VARCHAR(100)
PRIMARY KEY,
      pageRank INT,
      avgDuration INT );
```

The corresponding Hadoop benchmarks in the comparison paper used an inefficient and fragile textual format with different attributes separated by vertical bar characters:

```
   137|http://www.somehost.com/
   index.html|602
```

In contrast to ad hoc, inefficient formats, virtually all MapReduce operations at Google read and write data in the Protocol Buffer format.[8] A high-level language describes the input and output types, and compiler-generated code is used to hide the details of encoding/decoding from application code. The corresponding protocol buffer description for the Rankings data would be:

```
message Rankings {
   required string pageurl = 1;
   required int32 pagerank = 2;
   required int32 avgduration = 3;
}
```

The following Map function fragment processes a Rankings record:

```
Rankings r = new Rankings();
r.parseFrom(value);
if (r.getPagerank() > 10) { ... }
```

The protocol buffer framework allows types to be upgraded (in constrained ways) without requiring existing applications to be changed (or even recompiled or rebuilt). This level of schema support has proved sufficient for allowing thousands of Google engineers to share the same evolving data types.

MapReduce is a highly effective and efficient tool for large-scale fault-tolerant data analysis.

Furthermore, the implementation of protocol buffers uses an optimized binary representation that is more compact and much faster to encode and decode than the textual formats used by the Hadoop benchmarks in the comparison paper. For example, the automatically generated code to parse a Rankings protocol buffer record runs in 20 nanoseconds per record as compared to the 1,731 nanoseconds required per record to parse the textual input format used in the Hadoop benchmark mentioned earlier. These measurements were obtained on a JVM running on a 2.4GHz Intel Core-2 Duo. The Java code fragments used for the benchmark runs were:

```
// Fragment 1: protocol buf-
fer parsing
for (int i = 0; i < numItera-
tions; i++) {
   rankings.parseFrom(value);
   pagerank = rankings.get-
   Pagerank();
}

// Fragment 2: text for-
mat parsing (extracted from
Benchmark1.java
// from the source code
posted by Pavlo et al.)
for (int i = 0; i < numItera-
tions; i++) {
   String data[] = value.to-
   String().split("\\|");
   pagerank = Integer.
   valueOf(data[0]);
}
```

Given the factor of an 80-fold difference in this record-parsing benchmark, we suspect the absolute numbers for the Hadoop benchmarks in the comparison paper are inflated and cannot be used to reach conclusions about fundamental differences in the performance of MapReduce and parallel DBMS.

Fault Tolerance

The MapReduce implementation uses a pull model for moving data between mappers and reducers, as opposed to a push model where mappers write directly to reducers. Pavlo et al. correctly pointed out that the pull model can result in the creation of many small files and many disk seeks to move data between mappers and reducers. Implementation tricks like batching, sorting, and grouping of intermediate data and smart scheduling of reads are used by Google's MapReduce implementation to mitigate these costs.

MapReduce implementations tend not to use a push model due to the fault-tolerance properties required by Google's developers. Most MapReduce executions over large data sets encounter at least a few failures; apart from hardware and software problems, Google's cluster scheduling system can preempt MapReduce tasks by killing them to make room for higher-priority tasks. In a push model, failure of a reducer would force re-execution of all Map tasks.

We suspect that as data sets grow larger, analyses will require more computation, and fault tolerance will become more important. There are already more than a dozen distinct data sets at Google more than 1PB in size and dozens more hundreds of TBs in size that are processed daily using MapReduce. Outside of Google, many users listed on the Hadoop users list[11] are handling data sets of multiple hundreds of terabytes or more. Clearly, as data sets continue to grow, more users will need a fault-tolerant system like MapReduce that can be used to process these large data sets efficiently and effectively.

Performance

Pavlo et al. compared the performance of the Hadoop MapReduce implementation to two database implementations; here, we discuss the performance differences of the various systems:

Engineering considerations. Startup overhead and sequential scanning speed are indicators of maturity of implementation and engineering tradeoffs, not fundamental differences in programming models. These differences are certainly important but can be addressed in a variety of ways. For example, startup overhead can be addressed by keeping worker processes live, waiting for the next MapReduce invocation, an optimization added more than a year ago to Google's MapReduce implementation.

Google has also addressed sequential scanning performance with a variety of performance optimizations by, for example, using efficient binary-encoding format for structured data (protocol buffers) instead of inefficient textual formats.

Reading unnecessary data. The comparison paper says, "MR is always forced to start a query with a scan of the entire input file." MapReduce does not require a full scan over the data; it requires only an implementation of its input interface to yield a set of records that match some input specification. Examples of input specifications are:

- All records in a set of files;
- All records with a visit-date in the range [2000-01-15..2000-01-22]; and
- All data in Bigtable table T whose "language" column is "Turkish."

The input may require a full scan over a set of files, as Pavlo et al. suggested, but alternate implementations are often used. For example, the input may be a database with an index that provides efficient filtering or an indexed file structure (such as daily log files used for efficient date-based filtering of log data).

This mistaken assumption about MapReduce affects three of the five benchmarks in the comparison paper (the selection, aggregation, and join tasks) and invalidates the conclusions in the paper about the relative performance of MapReduce and parallel databases.

Merging results. The measurements of Hadoop in all five benchmarks in the comparison paper included the cost of a final phase to merge the results of the initial MapReduce into one file. In practice, this merging is unnecessary, since the next consumer of MapReduce output is usually another MapReduce that can easily operate over the set of files produced by the first MapReduce, instead of requiring a single merged input. Even if the consumer is not another MapReduce, the reducer processes in the initial MapReduce can write directly to a merged destination (such as a Bigtable or parallel database table).

Data loading. The DBMS measurements in the comparison paper demonstrated the high cost of loading input data into a database before it is analyzed. For many of the benchmarks in the comparison paper, the time needed to load the input data into a parallel database is five to 50 times the time needed to analyze the data via Hadoop. Put another way, for some of

ACM Transactions on Accessible Computing the benchmarks, starting with data in a collection of files on disk, it is possible to run 50 separate MapReduce analyses over the data before it is possible to load the data into a database and complete a single analysis. Long load times may not matter if many queries will be run on the data after loading, but this is often not the case; data sets are often generated, processed once or twice, and then discarded. For example, the Web-search index-building system described in the MapReduce paper[4] is a sequence of MapReduce phases where the output of most phases is consumed by one or two subsequent MapReduce phases.

Conclusion

The conclusions about performance in the comparison paper were based on flawed assumptions about MapReduce and overstated the benefit of parallel database systems. In our experience, MapReduce is a highly effective and efficient tool for large-scale fault-tolerant data analysis. However, a few useful lessons can be drawn from this discussion:

Startup latency. MapReduce implementations should strive to reduce startup latency by using techniques like worker processes that are reused across different invocations;

Data shuffling. Careful attention must be paid to the implementation of the data-shuffling phase to avoid generating `O(M*R)` seeks in a MapReduce with `M` map tasks and `R` reduce tasks;

Textual formats. MapReduce users should avoid using inefficient textual formats;

Natural indices. MapReduce users should take advantage of natural indices (such as timestamps in log file names) whenever possible; and

Unmerged output. Most MapReduce output should be left unmerged, since there is no benefit to merging if the next consumer is another MapReduce program.

MapReduce provides many significant advantages over parallel databases. First and foremost, it provides fine-grain fault tolerance for large jobs; failure in the middle of a multi-hour execution does not require restarting the job from scratch. Second, MapReduce is very useful for handling data processing and data loading in a heterogenous system with many different storage systems. Third, MapReduce provides a good framework for the execution of more complicated functions than are supported directly in SQL.

MapReduce provides fine-grain fault tolerance for large jobs; failure in the middle of a multi-hour execution does not require restarting the job from scratch.

References

1. Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D.J., Silberschatz, A., and Rasin, A. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads. In *Proceedings of the Conference on Very Large Databases* (Lyon, France, 2009); http://db.cs.yale.edu/hadoopdb/
2. Aster Data Systems, Inc. *In-Database MapReduce for Rich Analytics*; http://www.asterdata.com/product/mapreduce.php.
3. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R.E. Bigtable: A distributed storage system for structured data. In *Proceedings of the Seventh Symposium on Operating System Design and Implementation* (Seattle, WA, Nov. 6–8). Usenix Association, 2006; http://labs.google.com/papers/bigtable.html
4. Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. In *Proceedings of the Sixth Symposium on Operating System Design and Implementation* (San Francisco, CA, Dec. 6–8). Usenix Association, 2004; http://labs.google.com/papers/mapreduce.html
5. Dewitt, D. and Stonebraker, M. MapReduce: A Major Step Backwards blogpost; http://databasecolumn.vertica.com/database-innovation/mapreduce-a-major-step-backwards/
6. Dewitt, D. and Stonebraker, M. MapReduce II blogpost; http://databasecolumn.vertica.com/database-innovation/mapreduce-ii/
7. Ghemawat, S., Gobioff, H., and Leung, S.-T. The Google file system. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles* (Lake George, NY, Oct. 19–22). ACM Press, New York, 2003; http://labs.google.com/papers/gfs.html
8. Google. Protocol Buffers: Google's Data Interchange Format. Documentation and open source release; http://code.google.com/p/protobuf/
9. Greenplum. Greenplum MapReduce: Bringing Next-Generation Analytics Technology to the Enterprise; http://www.greenplum.com/resources/mapreduce/
10. Hadoop. Documentation and open source release; http://hadoop.apache.org/core/
11. Hadoop. Users list; http://wiki.apache.org/hadoop/PoweredBy
12. Olston, C., Reed, B., Srivastava, U., Kumar, R., and Tomkins, A. Pig Latin: A not-so-foreign language for data processing. In *Proceedings of the ACM SIGMOD 2008 International Conference on Management of Data* (Auckland, New Zealand, June 2008); http://hadoop.apache.org/pig/
13. Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S., and Stonebraker, M. A comparison of approaches to large-scale data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference* (Providence, RI, June 29–July 2). ACM Press, New York, 2009; http://database.cs.brown.edu/projects/mapreduce-vs-dbms/
14. Pike, R., Dorward, S., Griesemer, R., and Quinlan, S. Interpreting the data: Parallel analysis with Sawzall. *Scientific Programming Journal, Special Issue on Grids and Worldwide Computing Programming Models and Infrastructure 13*, 4, 227–298. http://labs.google.com/papers/sawzall.html

Jeffrey Dean (jeff@google.com) is a Google Fellow in the Systems Infrastructure Group of Google, Mountain View, CA.

Sanjay Ghemawat (sanjay@google.com) is a Google Fellow in the Systems Infrastructure Group of Google, Mountain View, CA.

DOI:10.1145/1629175.1629199

Exciting research in the design of automated negotiators is making great progress.

BY RAZ LIN AND SARIT KRAUS

Can Automated Agents Proficiently Negotiate with Humans?

NEGOTIATIONS SURROUND OUR everyday life, usually without us even noticing them. They can be simple and ordinary, as in haggling over a price in the market or deciding on a meeting time; or they can be complex and extraordinary, perhaps involving international disputes and nuclear disarmament[14] issues that affect the well-being of millions.

While the ability to negotiate successfully is critical for any social interaction, the act of negotiation is not an easy task. Something that might be perceived as a "simple" case of a single-issue bilateral bargaining over a price in the marketplace can demonstrate the difficulties that arise during the negotiation process. In fact, it may demonstrate the complexity of negotiation and the modeling of the environment. Each of the two sides has his or her own preferences, which might or might not be known to the other party. And if some of these preferences conflict, reaching an agreement requires a certain degree of cooperation or concession.

Keeping all this in mind, negotiation is an attractive environment for automated agents. The many benefits of such agents include alleviating some of the efforts required of humans during negotiations and assisting individuals who are less qualified in the negotiation process, or in some situations, replacing human negotiators altogether. Another possibility is for people embarking on important negotiation tasks to use these agents as a training tool, prior to actually performing the task. Thus, success in developing an automated agent with negotiation capabilities has great advantages and implications. The design of automated agents that proficiently negotiate is a challenging task, as there are different environments and constraints that should be considered.

The negotiation environment defines the specific settings of the negotiation. Based on these settings, different considerations should then be taken into account. In this article, we focus on the question of whether an automated agent can proficiently negotiate with human negotiators. To this end we define a proficient automated negotiator as one that can achieve the best possible agreement for itself. This, of course, also depends on the preferences of the other party and thus adds complexity to the design of such an agent.

The Negotiation Environment

The designer of an automated agent must take into account the environment in which the agent will operate. The environment determines several parameters that dictate the number of negotiators taking part in the negotia-

ILLUSTRATION BY DARK GEOMETRY ART STUDIO, ISTOCKPHOTO.COM

tion, the time frame of the negotiation, and the issues on which the negotiation is being conducted. The number of parties participating in the negotiation process can be two (bilateral negotiations) or more (multilateral negotiations). For example, in a market there can be one seller but many buyers, all involved in negotiating over a certain item. On the other hand, if the item is common, there may also be many sellers taking part in the negotiation process.

The negotiation environment also consists of a set of objectives and issues to be resolved. Various types of issues can be involved, including discrete enumerated value sets, integer-value sets, and real-value sets. A negotiation consists of multi-attribute issues if the parties have to negotiate an agreement that involves several attributes for each issue. Negotiations that involve multi-attribute issues allow making complex decisions while taking into account multiple factors.[18] The negotiation environment can consist of non-cooperative negotiators or cooperative negotiators. Generally speaking, cooperative agents try to maximize their combined joint utilities (see Zhang[40]) while non-cooperative agents try to maximize their own utilities regardless of the other sides' utilities.

Finally, the negotiation protocol defines the formal interaction between the negotiators—whether the negotiation is done only once (one-shot) or repeatedly—and how the exchange of offers between the agents is conducted. A common exchange of offers model is the alternating offers model.[32] In addition, the protocol states whether agreements are enforceable or not, and whether the negotiation has a finite or infinite horizon. The negotiation is said to have a finite horizon if the length of every possible history of the negotiation is finite. In this respect, time costs may also be assigned and they may increase or decrease the utility of the negotiator.

Figure 1. Variations of the negotiation settings.

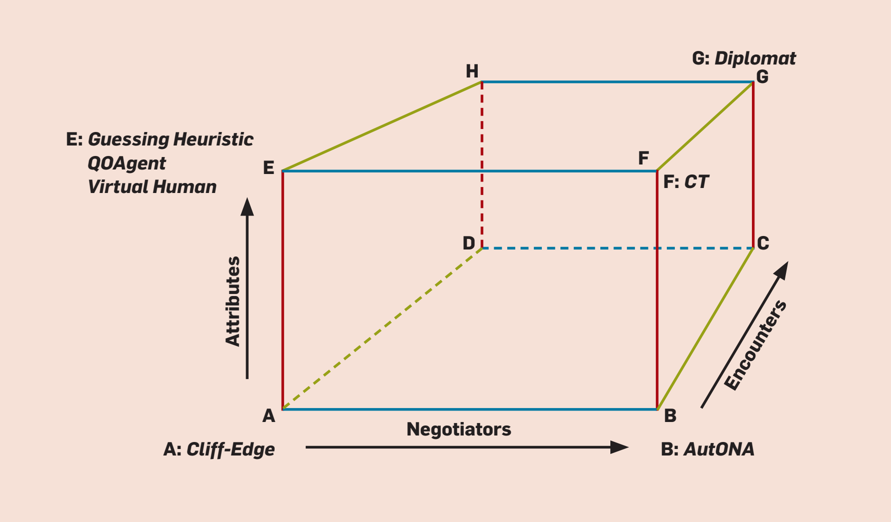


Figure 2. Example of virtual humans' negotiations.

Figure 1 depicts the different variations in the settings, along with the location of each system that is described in the section "Tackling the Challenges." For example, point D in the cube represents bilateral negotiations with multi-attribute issues and repeated interactions, while point B represents multilateral negotiations with a single attribute for negotiation and a one-shot encounter.

The negotiation domain encompasses the negotiation objectives and issues and assigns different values to each. Thus, an agent may be tailored to a given domain (for example, the *Diplomat* agent[22] described later is tailored to a specific domain of the Diplomacy game) or domain independent (for example, the *QOAgent*[24] also described later).

The Information Model

The information model dictates what is known to each agent. It can be a model of complete information, in which each agent has complete knowledge of both the state of the world and the preferences of other agents; or it can be a model of incomplete information, in which agents may have only partial knowledge of either the states of the world or the preferences of other agents (for example, bargaining games with asymmetric information), or they may be ignorant of the preferences of the opponents and the states of the world.[33] The incomplete information can be modeled in different ways with respect to the uncertainty regarding the preferences of the other party. One approach to modeling the information is to assume that there is a set of different agent types and the other party can be any one of these types.

Human-Agent Negotiations

The issue of automated negotiation is too broad to cover in a short review paper. To this end, we have decided to concentrate on adversarial *bilateral bargaining* in which the automated agent is matched with people. The challenges in this area could motivate readers to pursue this field (note that this sets the focus and leaves most auction settings outside the scope of this article, even though automated agents that bid in auctions competing with humans have been proposed and evaluated in the literature; for example, Grossklags and Schmidt[11]).

Automated Negotiator Agents. The problem of developing an automated agent for negotiations is not new for researchers in the fields of multiagent systems and game theory (for example, Kraus[20] and Muthoo[26]). However, designing an automated agent that can successfully negotiate with a human counterpart is quite different from negotiating with another automated agent. Although an automated agent that played in the Diplomacy game with other human players was introduced by Kraus and Lehmann[22] some 20 years ago, the difficulties of designing proficient automated negotiators have not been resolved.

In essence, assumptions in most research are made that do not necessarily apply in genuine negotiations with humans, such as assuming complete information or the rationality of the opponent negotiator. In this sense, both parties are assumed to be rational in their behavior (for example, the decisions made by the agents are described as rational and the agents are considered to be expected utility maximizing agents that cannot deviate from their prescribed behavior). Yet, when dealing with human counterparts, one must take into consideration the fact that humans do not necessarily maximize expected utility or behave rationally. In particular, results from social sciences suggest that people do not follow equilibrium strategies.[6,25] Moreover, when playing with humans, the theoretical equilibrium strategy is not necessarily the optimal strategy.[38] In this respect, equilibrium-based automated agents that play with people must incorporate heuristics to allow for “unknown” deviations in the behavior of the other party. Moreover, when people are the ones who design agents, they do not always design them to follow equilibrium strategies.[12] Nonetheless, some assumptions are made, mainly that the other party will not necessarily maximize its expected utility. However, if given two offers, it will prefer the one with the highest utility value. Lastly, it has been shown that whether the opponent is oblivious or has full knowledge that its counterpart is a computer agent can change the overall result. For example, Grossklags and Schmidt[11] showed that efficient market prices were achieved when human subjects knew that computer agents existed in a double auction market environment. Sanfey[34] matched humans with other humans and with computer agents in the Ultimatum Game and showed that people rejected unfair offers made by humans at significantly higher rates than those made when matched with a computer agent.

Automated Agents Negotiating with People. Researchers have tried to take some of these issues into consideration when designing agents that are capable of proficiently negotiating with people. For example, dealing only with the bounded rationality of the opponent, several researchers have suggested new notions of equilibria (for example, the *trembling hand equilibrium* described in Rasmusen[30]). Approximately 10 years ago, Kasbah, a seminal negotiation model between agents designed by humans, was presented in the virtual marketplace by Chavez and Maes.[5] Here, the agent’s behavior was fully controlled by human players. The main idea was to help users in the negotiation process between buyers and sellers by using automated negotiators. Chavez and Maes’s main innovation was not so much the sophisticated design of the automated negotiators but rather the creation of a multiagent negotiation environment. Kraus[21] describe an automated agent that negotiates proficiently with humans. Although they also deal with negotiation with humans, there is complete information in their settings. Other researchers have suggested a shift from quantitative decision theory to qualitative decision theory.[36] In using such a model it is not necessary to assume that the opponent will follow the equilibrium strategy or try to be a utility maximizer. Another approach was to develop heuristics for negotiations motivated by the behavior of people in negotiations.[22] However, the fundamental question of whether it is possible to build automated agents for negotiations with humans in open environments has not been fully addressed by these researchers.

Another direction being pursued is the development of virtual humans to train people in interpersonal skills (for example, Kenny[19]). Achieving this goal requires cognitive and emotional modeling, natural language processing, speech recognition, knowledge representation, as well as the construction and implementation of the appropriate logic for the task at hand (for example, negotiation), is in order to make the virtual human into a good trainer. An example of the researchers’ prototype, in which trainees conduct real-time negotiations with a virtual human doctor and a village elder to move a clinic to another part of the town out of harm’s way is given in Figure 2.

Commercial companies and schools have also displayed interest in automated negotiation technologies. Many courses and seminars are offered for the public and for institutions. These courses often guarantee that upon completion you will “know many strategies on which to base the negotiation,” “Discover the negotiation secrets and techniques,” “Learn common rival’s tactics and how to neutralize them” and “Be able to apply an efficient negotiation strategy.”[1,27] Yet, in many of these courses, the agents are restricted to one domain and cannot be generalized. Some of the automated agents cannot be adapted to the user and are restricted to a single attribute negotiation with no time constraints. Nonetheless, human factors and results of laboratory and field experiments reviewed in esteemed publications[9,29] provide guidelines for the design of automated negotiators. Yet, it is still a great challenge to incorporate these guidelines in the inherent design of an agent to allow it to proficiently negotiate with people.

The Main Challenges

The main difficulty in the development of automated negotiators is that in order to negotiate proficiently with a human counterpart, they must be able to work in settings with both opponents with bounded rationality and incomplete information. The difficulty can also stem from the fact the humans are also influenced by behavioral aspects and by social preferences that hold between players (such as inequity-aversion[2] and reciprocity[4]). Thus, it is difficult to predict individual choices.

Tackling the issues of bounded rationality and incomplete information is a complex task. To achieve this, an automated agent is required to have two interdependent mechanisms. The first is a decision-making component that works via modeling human factors. This mechanism is in charge of generating offers and deciding whether to accept or reject offers made by the opponent. The challenge behind this mechanism does not lie in the computational complexity of making good decisions but rather in reasoning about the psychological and social factors that characterize human behavior. The second component is learning, which allows the agent to infer the opponent's preferences and strategies, based on his actions.

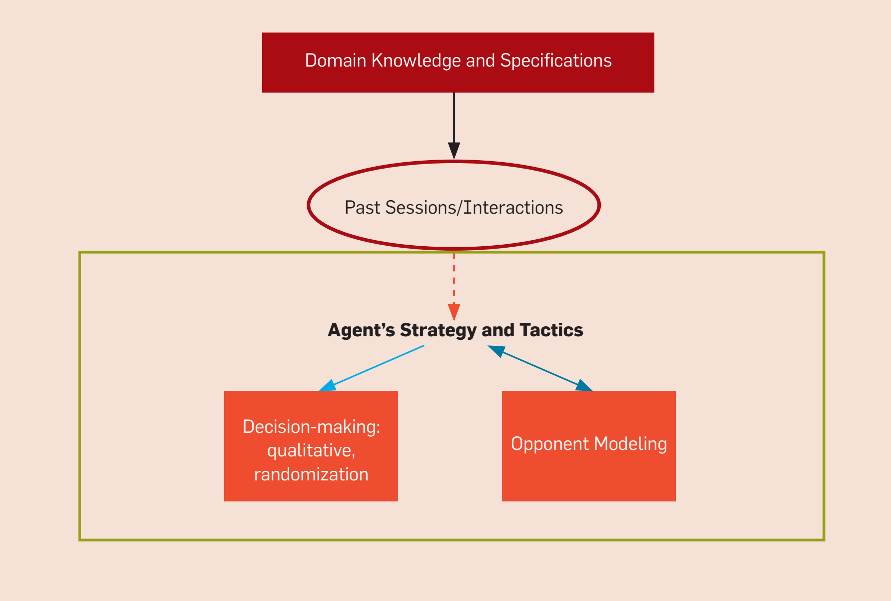
Another inherent problem in the design of the automated agent is the ability to generalize its behavior. While humans can negotiate in different settings and domains, when designing an automated agent a decision should be made whether the agent should be a general-purpose negotiator, that is, will be able to successfully negotiate in many settings and be domain-independent,[24] or the agent will only be suitable for one specific domain (for example, Ficici and Pfeffer,[8] Kraus and Lehmann[22]). Perhaps the advantage of the agent's specificity is the ability to construct better strategies that could allow it to achieve better agreements, as compared to a more general-purpose negotiator. This is due to the fact that the specificity allows the designer to debug the agent's strategy more carefully and against more test cases. By doing so, the designer can fine-tune the agent's strategy and allow for a more proficient automated negotiator. Agents that are domain independent, on the other hand, are more difficult to test against all possible cases and states.

The issue of trust also plays an important role in negotiations, especially when the other side's behavior is unpredictable. Successful negotiations depend on the trust established between all parties, which can depend on cheap-talk during negotiations (that is, unverifiable information with regard to the other party's private information[7]) and the introduction of unenforceable agreements. Based on the actions and information each party can update its reputation (for better or for worse) with regard to the other party and thus build trust between the sides. Some of the systems we review below do allow cheap-talk and unenforceable agreements. Building trust can also depend on past and future interactions with the other party (for example, one-shot interaction or repeated interactions). Due to limited space, we do not cover the issue of trust in detail. Readers are encouraged to refer to Ross[31] for a comprehensive review on this topic.

Another important issue is how automated agents can be evaluated and compared. Such an evaluation is important in order to select the most appropriate agent for the task at hand. Yet, no single criteria is defined. The answer to the questions of "what constitutes a good negotiator agent?" is multifaceted. For example, is a good agent an agent that:

- Achieves a maximal payoff when matched with human negotiators? But

Figure 3. Architecture of a general agent's design.


Domain Knowledge and Specifications
Past Sessions/Interactions
Agent's Strategy and Tactics
Decision-making: qualitative, randomization
Opponent Modeling

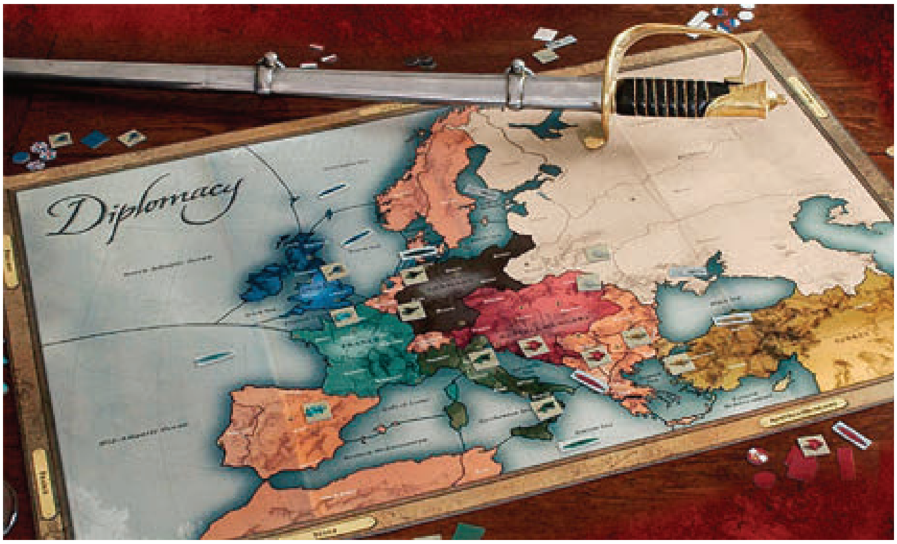


Figure 4. The Diplomacy game.

will it also generate these payoffs when matched with other automated agents, which might be more accessible than human negotiators, and which also exist in open environments?

- Generates a maximal combined payoff for both negotiators, that is, the agent is more concerned with maximizing the combined utilities than its own reward?
- Allows most negotiations to end with an agreement, rather than one of the sides opting-out or terminating the negotiations with a status-quo outcome?
- Is domain dependent and its technique suitable only for that domain or one that is domain independent and can be adapted to several domains? This might be an important factor if an agent is required to adapt to dynamic settings, for example.
- Behave in such a manner that would leave its counterpart speculating whether it is an automated negotiator or a human one?

In this article we do not define what or whether there is a best answer. We also do not claim a best answer indeed exists. Yet researchers should take these and other measures into consideration when designing their agents. Perhaps certain criteria and benchmarks are in order to allow an adequate comparison between automated agents.

Here we review automated agents that incorporate the two mechanisms of decision making via modeling human factors and learning the opponent's model. By doing so they try to tackle the aforementioned challenges in bilateral negotiations. While many automated negotiators' designs have been suggested in the literature, we only review those that have actually been evaluated and tested with human counterparts. This is mainly due to the fact that in order to test the proficiency of an automated negotiator whose purpose is to negotiate with human negotiators, one must match it with humans. It is not sufficient to test it with other automated agents, even if they were supposed to have been designed by humans as bounded rational agents, due to many of the reasons previously mentioned.

Tackling the Challenges

Here we describe several automated agents that try to tackle the challenges and proficiently negotiate in open environments. All of these agents were evaluated with human counterparts. It is worth noting that most of these agents use structured (or semi-structured) language and do not implement any natural language processing methods (with the one exception of the Virtual Human agent). In addition, the agents vary with respect to their characteristics. For example, some are domain-dependent, while others are domain-independent and are more general in nature; some use the history of past interactions to model the opponent, while others only have access to current interaction data. Figure 3 depicts a general architecture for an automated agent design. We begin by describing the oldest agent of all of them—the *Diplomat* agent.

The *Diplomat* Agent

Over 20 years ago Kraus and Lehmann developed an agent called *Diplomat*[22] that played the Diplomacy game (see Figure 4) with the goal to win. The game involves negotiations in multi-issue settings with incomplete information concerning the other agents' goals, and misleading information can be exchanged between the different agents. The negotiation protocol extends the model of alternating offers and allows simultaneous negotiations between the parties, as well as multiple interactions with the opponent agents during each time period. The issue of trust also plays an important role, as commitments might be breached. In addition, as each game consists of several sessions, it can be viewed as repeated negotiation settings.

The main innovation of the *Diplomat* agent is probably the fact that it consists of five different modules that work together to achieve a common goal. Different personality traits are implemented in the different modules. These traits affect the behavior of the agent and can be changed during each run, allowing *Diplomat* to change its 'personality' from one game to another and to act nondeterministically. In addition, the agent has a limited learning capability that allows it to try to estimate the personality traits of its rivals (for example, their risk attitude). Based on this, *Diplomat* assesses whether or not the other players will keep their promises. In addition, *Diplomat* incorporates randomization in its decision-making component. This randomization, influenced by *Diplomat*'s personality traits, determines whether some agreements will be breached or fulfilled.

The results reported by Kraus and Lehmann show that *Diplomat* played well in the games in which it participated, and most human players were not able to guess which of the players was played by the automated agent. Nonetheless, the main disadvantage of *Diplomat* is that it is a domain-dependent agent, that is, suitable only for the Diplomacy game. Since the game is quite complex and time consuming not many experiments were carried out with human players to validate the results and reach a level of significance. Yet, at the time *Diplomat* did open a new and exciting line of research, some of which we review here.

We continue with a more recent agent also constrained to a specific domain and involving single-issue negotiations. However, it takes into account the history of past interactions to model the opponents.

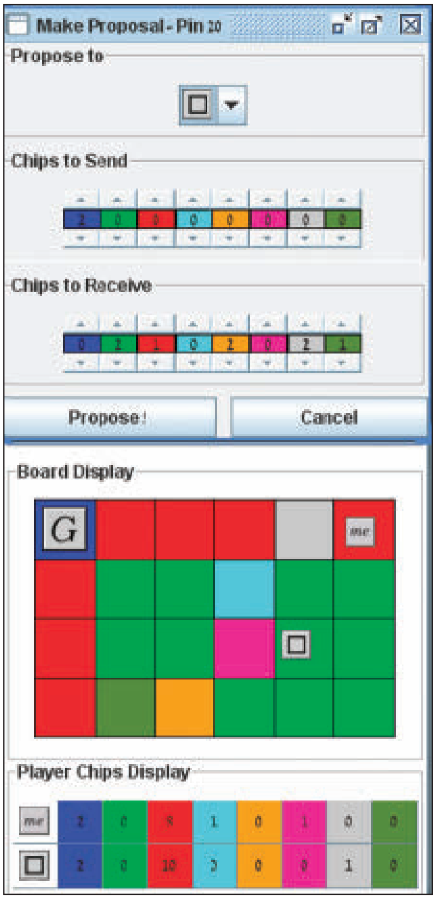


Figure 5. The Colored-Trail game screenshot.

The *AutONA* Agent

Byde[3] developed *AutONA*—an automated negotiation agent. Their problem domain involves multiple negotiations between buyers and sellers over the price and quantity of a given product. The negotiation protocol follows the alternating offers model. Each offer is directed at only one player on the other side of the market, and is private information between each pair of buyers and sellers. In each round, a player can make a new offer, accept an offer, or terminate negotiations. In addition, a time cost is used to provide incentives for timely negotiations. While the model can be viewed as one-shot negotiations, for each experiment, *AutONA* was provided with data from previous experiments.

In order to model the opponent, *AutONA* attaches a belief function to each player that tries to estimate the probability of a price for a given seller and a given quantity. This belief function is updated based on observed prices in prior negotiations. Several tactics and heuristics are implemented to form the strategy of the negotiator during the negotiation process (for example, for selecting the opponents with which it will negotiate and for determining the first offer it will suggest to the opponent). Byde also allowed cheap-talk during negotiations, that is, the proposition of offers with no commitments. The results obtained from the experiments with human negotiators revealed that the negotiators did not detect which negotiator was the software agent. In addition, Byde found that *AutONA* is not sufficiently aggressive during negotiations and thus many remained incomplete. Their experiments showed that at first *AutONA* performed worse than the human players. Thus, a modified version that fine-tuned several configuration parameters of the *AutONA* agent, improved the results that were more in line with those of human negotiators, yet not better. They conclude that different environments would most likely require changing the configurations of the *AutONA* agent.

We now proceed with agents that are applicable to a larger family of domains: The *Cliff Edge* and *Colored Trails* agents.

The *Cliff-Edge* Agent

Katz and Kraus[16] proposed an innovative model for human learning and decision making. Their agent competes repeatedly in one-shot interactions, each time against a different human opponent (for example, sealed-bid first-price auctions, ultimatum game). Katz and Kraus utilized a reinforcement learning algorithm that integrates virtual learning with reinforcement learning. That is, offers higher than an accepted offer are treated as successful (virtual) offers, notwithstanding they were not actually proposed. Similarly, offers lower than a rejected offer are treated as having been (virtually) unsuccessfully proposed. A threshold is also employed to allow for some deviations from this strict categorization. The results of previous interactions are stored in a database used for later interactions. The decision-making mechanism of Katz and Kraus's Ultimatum Game agent follows a heuristic based on the qualitative theory of Learning Direction.[35] Simply speaking, if an offer is rejected at a given interaction, then at the next interaction the proposer will offer the opponent a higher offer. In contrast, if an offer is accepted, then during the following interaction the offer will be decreased. Katz and Kraus show that their algorithm performs better than other automated agents. When compared to human behavior, there is an advantage to their automated agent over the human's average payoff.

Later, Katz and Kraus[17] improved the learning of their agent by allowing gender-sensitive learning. In this case, the information obtained from previous negotiations is stored in three databases, one is general and the other two are each associated with a specific gender. During the interaction, the agent's algorithm tries to determine when to use each database. Katz and Kraus show their gender-sensitive agent yields higher payoffs than the generic approach, which lacks gender sensitivity.

However, Katz and Kraus's agent was tested in a single-issue domain with repeated interactions that are used to improve the learning and decision-making mechanism. It is not clear whether their approach would be applicable to negotiation domains in which several rounds are made with the same opponent and multi-issue offers are made. In addition, the success of their gender-sensitive approach depends on the existence of different behavioral patterns of different gender groups.

The following agents are tailored to a rich environment of multi-issue negotiations. Similar to the agent proposed by Katz, the history of past interactions is used to fine-tune agents' behavior and modeling.

The *Colored-Trails* Agents

Ficici and Pfeffer[8] were concerned with understanding human reasoning, and using this understanding to build their automated agents. They did so by means of collecting negotiation data and then constructing a proficient automated agent. Both Byde's *AutONA* agent[3] and the *Colored-Trail* agent collect historical data and use it to model the opponent. Byde used the data to update the belief regarding the price for each player, while Ficici and Pfeffer used it to construct different models of how humans reason in the game.

The negotiation was conducted in the Colored Trails game environment[12] played on a $n \times m$ board of colored squares. Players are issued colored chips and are required to move from their initial square to a designated goal square. To move to an adjacent square, a player must turn in a chip of the same color as the square. Players must negotiate with each other to obtain chips needed to reach the goal square (see Figure 5). Their learning mechanism involved constructing different possible models for the players and using gradient descent to learn the appropriate model.

Ficici and Pfeffer trained their agents with results obtained from human-human simulations and then incorporated their models in their automated agents that were later matched against human players. They show that this method allows them to generate more successful agents in terms of the expected number of accepted offers and the expected total benefit for the agent. They also illustrate how their agent contributes to the social good by providing high utility scores for the other players. Ficici and Pfeffer were also able to show that their agent performs similarly to human players.

In order for the *Colored-Trails* Agent to model the opponent, prior knowledge regarding the behavior of humans is needed. The learning mechanism requires sufficient human data for training and is currently limited to one domain only.

Gal[10] also examines automated agent design in the domain of the Colored Trails. They present a machine-learning approach for modeling human behavior in a two-player negotiation, where one player proposes a trade to the other, who can accept or reject it. Their model tries to predict the reaction of the opponent to the different offers, and using this prediction it determines the best strategy for the agent. The domain on which Gal et al. tested their agent can also be viewed as a Cliff-Edge environment, more complex than the Ultimatum Game, upon which Katz and Kraus evaluated their agent.[16]

Gal et al. show that the proposed model successfully learns the social preferences of the opponent and achieves better results than the Nash equilibrium, Nash bargaining computer agents, and human players.

We now continue with agents that are domain-independent, and we propose an agent that has greater generality than the aforementioned agents.

The *Guessing Heuristic* Agent

Jonker et al.[15] deal with bilateral multi-issue and multi-attribute negotiations that involve incomplete information. The negotiation follows the alternating offer protocol and is conducted once with each opponent. Jonker designed a generic agent that uses a "guessing heuristic" in the buyer-seller domain.[a] This heuristic tries to predict the opponent's preferences based on its offers' history. This is under the assumption the opponent's utility has a linear function structure. Jonker et al. assert that this heuristic allows their agent to improve the outcome of the negotiations. Regarding the offer generation mechanism, they use a concession mechanism to obtain the next offer. In their experiments, the automated agent acts as a proxy for the human user. The user is involved only in the beginning when he inputs the preference parameters. Then the agent generates the offers and the counteroffers. When comparing negotiations involving only automated agents with negotiations involving only humans, the agents usually outperformed the humans (in the buyer's role). Yet, in an additional experiment they matched humans versus agent negotiators. In this experiment, humans only played the role of the buyer. When comparing the human vs. agent negotiations to that of only automated agents, the humans attained somewhat better results than the agents (in the buyer's role), based on the average utilities. The authors believe this should be accounted to the fact that humans forced the automated negotiators to make more concessions then they themselves did.

If we look into the design elements of all the agents mentioned in this article, we cannot find one specific feature that connects them or can account for their good negotiation skills.

The next agent also deals with bilateral multi-issue negotiations that involve incomplete information. Nonetheless the negotiation protocol is richer than that of the *Guessing Heuristic* agent.

The *QOAgent*

The *QOAgent*[24] is a domain-independent agent that can negotiate with people in environments of finite horizon bilateral negotiations with incomplete information. The negotiations consider a finite set of multi-attribute issues and time constraints. Costs are assigned to each negotiator, such that during the negotiation process, the negotiator might gain or lose utility over time. If no agreement is reached by a given deadline a status quo outcome is enforced. A negotiator can also opt-out of the negotiation if it decides that the negotiation is not proceeding in a favorable manner. Similar to the negotiation protocol in the *Diplomat* agent's domain, the negotiation protocol in the *QOAgent*'s domain extends the model of alternating offers such that each agent can perform up to $M > 0$ interactions with the opponent agent during each time period. In addition, queries and promises are allowed that add unenforceable agreements to the environment.

With respect to incomplete information, each negotiator keeps his preferences private, though the preferenc-

[a] Although Jonker et al. discuss and present results on one domain only, they state their model is generic and has also been applied in other domains.

es might be inferred from the actions of each side (for example, offers made or responses to offers proposed). Incomplete information is expressed as uncertainty regarding the utility preferences of the opponent, and it is assumed there is a finite set of different negotiator types. These types are associated with different additive utility functions (for example, one type might have a long-term orientation regarding the final agreement, while the other type might have a more constrained orientation). Lastly, the negotiation is conducted once with each opponent.

As for incomplete information, the *QOAgent* tackles the problem by applying a simple Bayesian update mechanism, which, after each action tries to infer which utility best suits the opponent (when receiving an offer or when receiving a response to an offer). For the decision-making process, the approach used by the *QOAgent* is more of a qualitative approach.[36] While the *QOAgent*'s model applies utility functions, it is based on a non-classical decision-making method, rather than focusing on maximizing the expected utility. The *QOAgent* uses the maximin function and the qualitative valuation of offers. Using these methods the *QOAgent* generates offers and decides whether to accept or reject proposals it has received.

Lin et al.[24] tested the *QOAgent* in several distinct domains and their results show that the *QOAgent* reaches more agreements and plays more effectively than its human counterparts, when the effectiveness is measured by the score of the individual utility. They also show that the sum of utilities is higher in negotiations when the *QOAgent* is involved, as compared to human-human negotiations. Thus, they assert, it is indeed possible to build an automated agent that can negotiate successfully with humans. However, it is also important to state that their agent has certain limitations. They assume there is a finite set of different agent types and thus their agent cannot generate a dynamic model (and perhaps a more accurate one) of the opponent. In addition, they have not shown whether their agent can also maintain high scores when matched with other automated agents, which is an important characteristic of open environment negotiations. Moreover, the *QOAgent* does not scale well when numerous offers are proposed, which can cause its performance to deteriorate.

Finally, we conclude with a description of a more complex type of agent that incorporates many features, far beyond the negotiation strategy itself.

The *Virtual Human* Agent

Kenny et al.[19] describe work on virtual humans used for interpersonal training for skills, such as: negotiation, leadership, interviewing, and cultural training. To achieve this they require a large amount of research in many fields (such as, knowledge representation, cognitive and emotional modeling, natural language processing, among others). Their intelligent agent is based on the Soar Cognitive Architecture, which is a symbolic reasoning system used to make decisions.

Main contributions of each agent.

Agent	Main Contribution
Diplomat	Changing the agent's personality heuristics Non-deterministic behavior / randomization
AutONA	Tactics and heuristics Incorporating data from past interactions Concession mechanism
Cliff-Edge	Virtual learning Incorporating data from past interactions Gender-sensitive approach Non-deterministic behavior / randomization (implicitly)
Colored-Trails	Incorporating data from past interactions Machine learning
Guessing Heuristic	Generic agent / domain independent Concession mechanism
QOAgent	Generic agent / domain independent Qualitative decision making Non-deterministic behavior / randomization
Virtual Human	Tactics and heuristics Cognitive architecture

Traum et al. discuss the negotiation strategies of the virtual human agent in more detail.[37] In their paper they describe a set of strategies implemented by the agent (for example, when to act aggressively if it seems that the current outcome will incur a negative utility, or when to find the appropriate issue on which to currently negotiate). The strategy chosen each time is influenced by several factors: the control the agent has over the negotiations, the estimated utility of an outcome and the estimated best utility of an outcome, the trust the agent bestows the opponent and the commitment of all agents to the given issues. The virtual agent also tries to model the opponent by reasoning about its mental state.

Traum et al. tested their agents in several negotiation scenarios. One of these scenarios is a simulation for soldiers that practice and conduct bilateral engagements with virtual humans, and in situations in which culture plays an important role. In this case, the different actions can be selected from a menu that includes appropriate questions based on the history of the simulation thus far. The second domain requires trainees to communicate with an embodied virtual human doctor to negotiate and convince him to move a clinic, located in a middle of a war zone, out of harm's way (see Figure 2). Their prototypes are continuously tested with cadets and civilians. Traum et al. are more concerned with the system as a whole and thus they do not provide insights with respect to the proficiency of their automated negotiator. Regarding the environment, they state that the subjects enjoy using the system for negotiations and that it also allows them to learn from their mistakes.

Traum also report some of the existing limitations of their system. Currently, the virtual agent cannot consider arbitrary offers made by a human negotiator. In addition, more strategies are required to better cover the environment's rich settings. They also state that the negotiation problem can be addressed more in depth (following other researchers who have focused mainly on the negotiation field), rather than in breadth (as presently conducted in their system).

The Rule of Thumb for Designing Automated Agents

We should probably begin with the conclusion. Despite the title of this section, there may not be a good rule of thumb for designing automated negotiators with human negotiators. The accompanying table summarizes the main contributions made by each of the reviewed agents. If we look into the design elements of all the agents mentioned in this article, we cannot find one specific feature that connects them or can account for their good negotiation skills. Nonetheless, we can note several features that have been used in several agents. Agent designers might take these features into consideration when designing their automated agent, while also taking into account the settings and the environment in which their agent will operate.

The first feature is *randomization*, which was used in *Diplomat*, *QOAgent,* and also (though not explicitly) in the *Cliff-Edge* agents. The randomization factor allows these agents to be more resilient (or robust) to adversaries that try to manipulate them to gain better results on their part. In addition, it allows them to be more flexible, rather than strict, in accepting agreements and ending negotiations.

The second feature can be viewed as a *concession strategy*. Both the *AutONA* agent and the *Guessing Heuristic* agent implemented this strategy, which influenced the offer-generation mechanism of their agent. A concession strategy might also have a psychological effect on the opponent that would make it more comfortable for the opponent to accept agreements or to make concessions on his own as well.

The last feature common in several agents is the use of a *database*. The database can be built on previous interactions with the same human opponent or for all opponents. The agent consults the database to better model the opponent, to learn about possible behaviors and actions and to adjust its behavior to the specific opponent. A database of the history can also be used to obtain information about the behavior of the opponents, if such information is not known, or cannot be characterized, in advance.

Lastly, though not exactly a feature, but worth mentioning, is that none of the agents we reviewed implemented *equilibrium strategies*. This is an interesting observation and most likely is due to the fact that these strategies have been shown to behave poorly when implemented in automated negotiators matched with human negotiators, mainly due to the complex environment and the bounded rationality of people. In some cases,[21] experiments have shown that when the automated agent follows its equilibrium strategy the human negotiators who negotiate with it become frustrated, mainly since the automated agent repeatedly proposes the same offer, and the negotiation often ends with no agreement. This has been shown in cases in which the complexity of finding the equilibrium is low and the players have full information.

Conclusion

In this article we presented the challenges and current state-of-the-art automated solutions for proficient negotiations with humans. Nonetheless we do not claim that all existing solutions have been summarized in this article. We briefly state the importance of automated negotiators and propose suggestions for future work in this field.

The importance of designing an automated negotiator that can negotiate efficiently with humans cannot be understated and we have shown that indeed it is possible to design such negotiators. By pursuing non-classical methods of decision making and a learning mechanism for modeling the opponent it could be possible to achieve greater flexibility and effective outcomes. As we have shown, this can also be accomplished without constraining the model to the domain.

Many of the automated negotiation agents are not intended to replace humans in negotiations, but rather as an efficient decision support tool or as a training tool for negotiations with people. Thus, such agents can be used to support training in real-life negotiations, such as: e-commerce and electronic negotiations (e-negotiations), and they can also be used as the main tool in conventional lectures or online courses, aimed at turning the trainee into a better negotiator.

To date, it seems that research in AI has neglected the issue of proficiently negotiating with people, at the expense of designing automated agents aimed to negotiate with rational agents or other automated agents.[39] Others have focused on improving different heuristics and strategies and the analysis of game theory aspects (for example, Kraus[20] and Muthoo[26]). Nonetheless, it is noteworthy that these are important aspects in which the AI community has certainly made an impact. Unfortunately, not much progress has been made with regard to automated negotiators with people, leaving many unfaced challenges.

Suggestions for Future Research

The work is far from complete and the challenges remain exciting. To entice the reader, we list a few of these challenges here:

The first challenge is to enrich the negotiation language. Many researchers restrict themselves to the basic model of alternating offers whereby the language consists of offers and counteroffers alone. Rich and realistic negotiations, however, consist of other types of actions (for example, threats, comments, promises, and queries), as well as simultaneous actions (that is, each agent can perform up to $M > 0$ interactions with the other party each time period). It is essential these actions and behaviors are modeled in the automated negotiators to allow better negotiations with human negotiators.

Another challenge, also discussed previously, is the need for a general-purpose automated negotiator. With the vast amount of applications and domains, automated agents cannot be restricted to one single domain and must be adaptable to different settings. The trade-off between the performance of a general-purpose automated negotiator and a domain-dependent negotiator should be considered and methods for improving the efficacy of a general-purpose negotiator should be sought. Achieving this will also contribute to the feasibility of comparing between different automated agents when matched with people. Preliminary work on this facet is already under way by Hindriks et al.[13] and Oshrat et al.,[28] however, we believe the aspect of generality should be addressed more by researchers. In this respect, metrics should be designed to allow a comparison between agents. To achieve this, some of the questions described earlier regarding "what constitutes a good negotiator agent?" should be answered as well.

In addition, argumentation, though dealt with in the past, still poses a challenge for researchers in this field. For example, about 10 years ago Kraus[23] presented argumentation as an iterative process emerging from exchanges among agents to persuade each other and bring about a change in intentions. They developed a formal logic that forms a basis for the development of a formal axiomatization system for argumentation. In particular, Kraus identified argumentation categories in human negotiations and demonstrated how the logic can be used to specify argument formulations and evaluations. Finally, they developed an agent that was implemented, based on the logical model.

However, this agent was not matched with human negotiators. Moreover, there are several open research questions associated with how to integrate the argumentation model into automated negotiators. Since the argumentation module is based on logic and thus is time consuming, a more efficient approach should be used. In addition, the current model is built on a very complex model of the opponent and therefore should be incorporated in the automated negotiator's model of the opponent. In order to facilitate the design, a mapping between the logical model and the utility-based model is required.

To conclude, in recent years the field of automated negotiators that can proficiently negotiate with human players has received much needed focus and the results are encouraging. We presented several of these automated negotiators and showed it is indeed possible to design such proficient agents. Nonetheless, there are still challenges that pose interesting research questions that must be pursued and exciting work is still very much in progress.

Acknowledgments

We thank David Sarne, Ya'akov (Kobi) Gal, and the anonymous referees for their helpful remarks and suggestions.

References

1. Bargaining negotiations course; https://www.irwaonline.org/eweb/dynamicpage.aspx?webcode=205 (2008).
2. Bolton, G. A comparative model of bargaining: Theory and evidence. *American Economic Review 81*, 5 (1989), 1096–1136.
3. Byde, A., Yearworth, M., Chen, Y.-K., and Bartolini, C. *AutONA*: A system for automated multiple 1-1 negotiation. In *Proceedings of the 2003 IEEE International Conference on Electronic Commerce* (2003), 59–67.
4. Charness, G. and Rabin, M. Understanding social preferences with simple tests. *The Quarterly Journal of Economics 117*, 3 (2002), 817–869.
5. Chavez, A. and Maes, P. Kasbah: An agent marketplace for buying and selling goods. In *Proceedings of the first international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology* (1996), 75–90.
6. Erev, I. and Roth, A. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibrium. *American Economic Review 88*, 4 (1998), 848–881.
7. Farrell, J. and Rabin, M. Cheap talk. *Journal of Economic Perspectives 10*, 3 (1996), 103–118.
8. Ficici, S. and Pfeffer, A. Modeling how humans reason about others with partial information. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems* (2008), 315–322.
9. Fisher, R. and Ury, W. *Getting to Yes: Negotiating Agreement without Giving In.* Penguin Books, 1991.
10. Gal, Y., Pfeffer, A., Marzo, F. and Grosz, B.J. Learning social preferences in games. In *Proceedings of the National Conference on Artificial Intelligence* (2004), 226–231.
11. Grossklags, J. and Schmidt, C. Software agents and market (in) efficiency: a human trader experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 36*, 1 (2006), 56–67.
12. Grosz, B., Kraus, S., Talman, S. and Stossel, B. The influence of social dependencies on decision-making: Initial investigations with a new game. In *Proceedings of 3rd International Joint Conference on Multiagent Systems* (2004), 782–789.
13. Hindriks, K., Jonker, C. and Tykhonov, D. Towards an open negotiation architecture for heterogeneous agents. In *Proceedings for the 12th International Workshop on Cooperative Information Agents.* LNAI, 5180 (2008), Springer, NY, 264–279.
14. Hoppman, P.T. *The Negotiation Process and the Resolution of International Conflicts.* University of South Carolina Press, Columbia, SC, May 1996.
15. Jonker, C.M., Robu, V., and Treur, J. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems 15*, 2 (2007), 221–252.
16. Katz, R. and Kraus,S. Efficient agents for cliff-edge environments with a large set of decision options. In *Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems* (2006), 697–704.
17. Katz, R. and Kraus, S. Gender-sensitive automated negotiators. In *Proceedings of the 22nd National Conference on Artificial Intelligence* (2007), 821–826.
18. Keeney, R. and Raiffa, H. *Decisions with Multiple Objective: Preferences and Value Tradeoffs.* John Wiley, NY, 1976.
19. Kenny, P., Hartholt, A., Gratch, J., Swartout, W., Traum, D., Marsella, S. and Piepol, D. Building interactive virtual humans for training environments. In *Proceedings of Interservice/Industry Training, Simulation and Education Conference* (2007).
20. Kraus, S. *Strategic Negotiation in Multiagent Environments.* MIT Press, Cambridge MA, 2001.
21. Kraus, S., Hoz-Weiss, P., Wilkenfeld, S., Andersen, D.R., and Pate, A. Resolving crises through automated bilateral negotiations. *Artificial Intelligence 172*, 1 (2008), 1–18.
22. Kraus, S. and Lehmann, D. Designing and building a negotiating automated agent. *Computational Intelligence 11*, 1 (1995), 132–171.
23. Kraus, S., Sycara, K., and Evenchik, A. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence 104*, 1–2 (1998), 1–68.
24. Lin, R., Kraus, S., Wilkenfeld, J. and Barry, J. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence 172*, 6–7 (2008), 823–851.
25. McKelvey, R.D. and Palfrey, T.R. An experimental study of the centipede game. *Econometrica 60*, 4 (1992), 803–836.
26. Muthoo, A. *Bargaining Theory with Applications.* Cambridge University Press, MA, 1999.
27. Online negotiation course; http://www.negotiate.tv/ (2008).
28. Oshrat, Y., Lin, R., and Kraus, S. Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems* (2009).
29. Raiffa, H. *The Art and Science of Negotiation.* Harvard University Press, Cambridge, MA, 1982.
30. Rasmusen, E. *Games and Information: An Introduction to Game Theory.* Blackwell Publishers, 2001.
31. Ross, W. and LaCroix, J. Multiple meanings of trust in negotiation theory and research: A literature review and integrative model. *International Journal of Conflict Management 7*, 4 (1996), 314–360.
32. Rubinstein, A. Perfect equilibrium in a bargaining model. *Econometrica 1* (1982), 97–109.
33. Rubinstein, A. A bargaining model with incomplete information about preferences. *Econometrica 53*, 5 (1985), 1151–1172.
34. Sanfey, A., Rilling, J., Aronson, J., Nystrom, L., and Cohen, J. The neural basis of economic decision-making in the ultimatum game. *Science 300* (2003), 1755–1758.
35. Selten, R. and Stoecker, R. End behavior in sequences of finite prisoner's dilemma supergames: A learning theory approach. *Economic Behavior and Organization 7*, 1 (1986), 47–70.
36. Tennenholtz, M. On stable social laws and qualitative equilibrium for risk-averse agents. In Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (1996), 553-561.
37. Traum, D., Marsella, S., Gratch, J., Lee, J., and Hartholt, A. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Proceedings of the 8th International Conference on Intelligent Virtual Agents*, 2008.
38. Tversky, A. and Kahneman, D. The framing of decisions and the psychology of choice. *Science 211* (1981), 453–458.
39. Wellman, M.P., Greenwald, A., and Stone, P. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition.* MIT Press, Cambridge, MA, 2007.
40. Zhang, X., Lesser, V., and Podorozhny, R. Multi-dimensional, multistep negotiation for task allocation in a cooperative system. *Autonomous Agents and MultiAgent Systems 10*, 1 (2005), 5–40.

This research is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-08-1-0144 and under NSF grant 0705587.

Raz Lin is a Postdoctoral Fellow in the computer science department at Bar-Ilan University, Ramat-Gan, Israel.

Sarit Kraus is a professor of computer science department at Bar-Ilan University, Ramat-Gan, Israel, and adjunct professor in the Institute for Advanced Computer Studies at the University of Maryland, College Park, MD.

research highlights

DOI:10.1145/1629175.1629202

Technical Perspective
Native Client: A Clever Alternative

By Dan Wallach

GOOGLE'S NATIVE CLIENT (typically abbreviated "NaCl" and pronounced NAH-cull) is an intriguing new system that allows untrusted x86 binaries to run safely on bare metal. Untrusted code is already essential to the Web, whether shipping JavaScript source code, Java byte code, Flash applications, or ActiveX controls. Java, JavaScript, and Flash all use an intermediate representation that is quite abstracted from the hardware, using increasingly sophisticated analysis and compilation techniques to achieve good performance on modern computers.

ActiveX (or Netscape/Firefox plugins), on the other hand, allows the direct transmission of Windows x86 binary objects, digitally signed and manually approved by the user to run natively.

ActiveX has never been particularly desirable. It is not portable to non-Windows platforms, and every user is one mistaken click away from installing malware. Meanwhile, Flash has become a standard install, largely due to its powerful graphics and video libraries. (When you watch a YouTube video in your browser, you're looking at the Flash plugin.) Indeed, Flash has sufficient access to the local system that it has, itself, been the target of a variety of security attacks. Sure, you can uninstall Flash on your system (and mobile phones don't support it at all), but far too many Web sites assume you've got Flash installed, and will be unusable without it.

Into this gap, NaCl offers a clever alternative. A plugin like Flash, compiled and optimized in native x86 code, could be downloaded, installed, and run by any Web page without bothering the user for permission. If the plugin turned out to have security flaws, those would be contained by the walls that NaCl builds around the code.

Plugins could just run as distinct, unprivileged users in the system, leveraging the multi-user isolation mechanisms already present in any modern operating system, but this ignores several unpleasant realities. First, a substantial portion of the world's computers are running old Windows variants with unacceptable security holes. We must build stronger walls than those platforms' native mechanisms can support. Second, we have to worry about CPU bugs. While possibly the most famous CPU implementation error was Intel's Pentium floating-point division flaw (where arithmetic could yield errors in the low-order bits of the mantissa), other bugs have happened from time to time that result in more serious security ramifications. We need the ability to filter out instructions that might tickle CPU bugs or otherwise have undesirable behavior.

If all the world were running classic RISC machines, where every instruction was 32-bits long, this process would be simple. Variable length x86 instructions, however, allow any given array of bytes to correspond to multiple different instruction streams, depending on the exact byte offset to which you jump. Consequently, NaCl introduces a simple static verifier to ensure that all jump instructions can only target instructions on 32-byte-aligned boundaries, and to ensure that code blocks, starting at those offsets, have no known unsafe instructions.

The NaCl system hides the native system call interface and uses its own inter-process communication mechanism, while also building an "outer sandbox" using more traditional operating system process privilege limits. In principle, NaCl could be built into a browser and ActiveX and Flash could be kicked out. Adobe could recompile Flash to pass NaCl's verifier, and end users would have one less source of security holes to worry about. Also, if Web designers wanted to use different video codecs, they would no longer be limited to whatever Flash supports. Even better, as NaCl doesn't necessarily expose the native operating system's system call interface, we can even imagine NaCl apps running portably across Linux, OS X, and Windows (NaCl is even being extended to support ARM and x86-64).

How secure is the open-source NaCl implementation? I was one of the judges for a contest that Google held earlier this year to find out. In the end, only five teams had entries, together identifying what the Google development team considered to be 24 valid security issues. These can be roughly categorized into bugs in NaCl's support infrastructure (unhandled exceptions, buffer overflow vulnerabilities, and a few "type confusion" attacks that exploit the ability to pass one type where another was expected), and obscure instruction sequences that the static verifier missed (for example, the verifier missed a class of "prefix" bits on jump instructions that change their behavior). One vulnerability relied on NaCl's support for memory-mapping to unmap and remap a code segment, allowing unverified code to be executed. Clever attacks, but all straightforward to remediate.

In summary, the NaCl design as detailed in the following paper is pragmatic and attractive, with its known implementation flaws no worse than what we might see in any fledgling operating system's security boundaries. The NaCl codebase is small and simple enough that these sorts of bugs can and will be fixed if and when NaCl leaves the lab and gains market share in the field. C

Dan Wallach (dwallach@cs.rice.edu) is an associate professor in the Department of Computer Science at Rice University in Houston, TX.

DOI:10.1145/1629175.1629203

Native Client: A Sandbox for Portable, Untrusted x86 Native Code

By Bennet Yee, David Sehr, Gregory Dardyk, J. Bradley Chen, Robert Muth, Tavis Ormandy, Shiki Okasaka, Neha Narula, and Nicholas Fullagar

Abstract

Native Client is a sandbox for untrusted x86 native code. It aims to give browser-based applications the computational performance of native applications without compromising safety. Native Client uses software fault isolation and a secure runtime to direct system interaction and side effects through interfaces it controls. It further provides operating system portability for binary code while supporting performance-oriented features generally absent from Web application programming environments, such as thread support, instruction set extensions such as SSE, and use of compiler intrinsics and hand-coded assembler. We combine these properties in an open architecture that encourages community review and third-party tools.

1. INTRODUCTION

As an application platform, the modern Web browser brings together a remarkable combination of resources, including seamless access to Internet resources, high-productivity programming languages such as JavaScript, and the richness of the Document Object Model (DOM) for graphics presentation and user interaction. While these strengths put the browser in the forefront as a target for new application development, it remains handicapped in a critical dimension: computational performance. Thanks to Moore's Law and the zeal with which it is observed by the hardware community, many interesting applications get adequate performance in a browser despite this handicap. But there remains a set of computations that are generally infeasible for browser-based applications due to performance constraints, for example, simulation of Newtonian physics, computational fluid-dynamics, and high-resolution scene rendering. The current environment also tends to preclude the use of large bodies of high-quality code developed in languages other than JavaScript.

Modern Web browsers provide extension mechanisms such as ActiveX[7] and Netscape Plugin Application Programming Interface (NPAPI)[19] allowing native code to be loaded and run as part of a Web application. Such architectures allow plug-ins to circumvent the security mechanisms otherwise applied to Web content, while giving them access to full native performance, perhaps as a secondary consideration. Given this organization, and the absence of effective technical measures to constrain these plug-ins, browser applications that wish to use native code must rely on nontechnical measures for security, for example, manual establishment of trust relationships through pop-up dialog boxes or manual installation of a console application. Historically, these nontechnical measures have been inadequate to prevent execution of malicious native code, leading to inconvenience and economic harm.[3, 22] As a consequence we believe there is a prejudice against native code extensions for browser-based applications among experts and distrust among the larger population of computer users.

While acknowledging the insecurity of the current systems for incorporating native code into Web applications, we also observe that there is no fundamental reason why native code should be unsafe. In Native Client, we separate the problem of safe native execution from that of extending trust, allowing each to be managed independently. Conceptually, Native Client is organized in two parts: a constrained execution environment for native code to prevent unintended side effects and a runtime for hosting these native code extensions through which allowable side effects may occur safely.

The main contributions of this work are

- An infrastructure for OS- and browser-portable sandboxed x86 binary modules
- Support for advanced performance capabilities such as threads, SSE instructions, compiler intrinsics, and hand-coded assembler
- An open system designed for easy retargeting of new compilers and languages
- Refinements to CISC software fault isolation, using x86 segments for improved simplicity and reduced overhead

We combine these features in an infrastructure that supports safe side effects and local communication, while preventing arbitrary file system and network access. Overall, Native Client provides sandboxed execution of native code and portability across operating systems, delivering native code performance for the browser.

The remainder of the paper is organized as follows.

A previous version of this paper was published in *Proceedings of the 2009 IEEE Symposium on Security and Privacy*, May 2009.

Section 1.1 describes our threat model. Section 2 develops some essential concepts for the NaCl[a] system architecture and programming model. Section 3 gives additional implementation details, organized around major system components. Section 4 provides a quantitative evaluation of the system using more realistic applications and application components. In Section 5, we discuss some implications of this work. Section 6 discusses relevant prior and contemporary systems. Section 7 offers our conclusion.

1.1. Threat model

Native Client should run untrusted modules from any Web site with safety comparable to systems such as JavaScript. When presented to the system, an untrusted NaCl module may contain *arbitrary* code and data. A consequence is that the NaCl runtime must be able to confirm that the module conforms to our validity rules (detailed below). Modules that do not conform to these rules are rejected by the system.

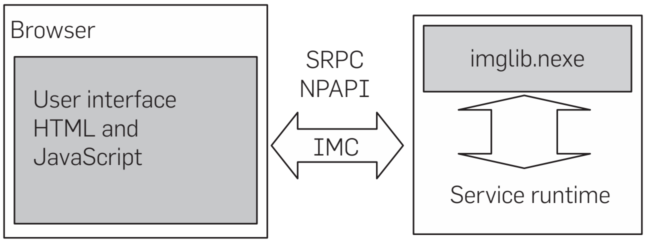
Once a conforming NaCl module is accepted for execution, the NaCl runtime must constrain its activity to prevent unintended side effects, such as might be achieved via unmoderated access to the native operating system's system call interface. The NaCl module may arbitrarily combine the entire variety of behaviors permitted by the NaCl execution environment in attempting to compromise the system. It may execute any reachable instruction block in the validated text segment. It may exercise the NaCl application binary interface to access runtime services in any way: passing invalid arguments, etc. It may also send arbitrary data via our intermodule communication interface, with the communicating peer responsible for validating input. The NaCl module may allocate memory and spawn threads up to resource limits. It may attempt to exploit race conditions in subverting the system.

The next sections detail how our architecture and code validity rules create a sandbox that effectively contains NaCl modules.

2. SYSTEM ARCHITECTURE

A NaCl application is composed of a collection of trusted and untrusted components. Figure 1 shows the structure of a hypothetical NaCl-based application for managing and sharing photos. It consists of two components: a user interface, implemented in JavaScript and executing in the Web browser, and an image processing library (imglib.nexe), implemented as a NaCl module. In this hypothetical scenario, the user interface and image processing library are part of the application and therefore untrusted. The browser component is constrained by the browser execution environment and the image library is constrained by the NaCl container. Both components are portable across operating systems and browsers, with native code portability enabled by Native Client. Prior to running the photo application, the user has installed Native Client as a browser plug-in. Note that the NaCl browser plug-in itself is OS and browser specific. Also note it is trusted, that is, it has full access to the OS system call interface and the user trusts it to not be abusive.

Figure 1. Hypothetical NaCl-based application. Untrusted modules have a gray background.



When the user navigates to the Web site that hosts the photo application, the browser loads and executes the application JavaScript components. The JavaScript in turn invokes the NaCl browser plug-in to load the image processing library into a NaCl container. Observe that the native code module is loaded silently—no pop-up window asks for permission. Native Client is responsible for constraining the behavior of the untrusted module.

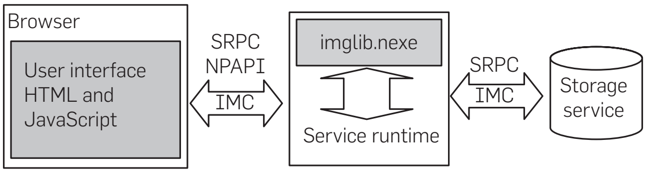
Each component runs in its own private address space. Inter-component communication is based on Native Client's reliable datagram service, the IMC (Inter-Module Communications). For communications between the browser and a NaCl module, Native Client provides two options: a Simple Remote Procedure Call (SRPC) facility, and NPAPI, both implemented on top of the IMC. The IMC also provides shared memory segments and shared synchronization objects, intended to avoid messaging overhead for high-volume or high-frequency communications.

The NaCl module also has access to a "service runtime" interface, providing for memory management operations, thread creation, and other system services. This interface is analogous to the system call interface of a conventional operating system.

In this paper we use "NaCl module" to refer to untrusted native code. Note however that applications can use multiple NaCl modules, and that both trusted and untrusted components may use the IMC. For example, the user of the photo application might optionally be able to use a (hypothetical) trusted NaCl service for local storage of images, illustrated in Figure 2. Because it has access to local disk, the storage service must be installed as a native browser plug-in; it cannot be implemented as a NaCl module. Suppose the photo application has been designed to optionally use the stable storage service; the user interface would check for the stable storage plug-in during initialization. If it detected the storage service plug-in, the user interface would establish an IMC communications channel to it, and pass a descriptor for the channel to the image library, enabling the image library and the storage service to communicate directly via IMC-based services (SRPC, shared memory, etc.). In this case the NaCl module will typically be statically linked against a library that provides a procedural interface for accessing the storage service, hiding details of the IMC-level communications such as whether it uses

[a] We use "NaCl" as an adjective reference to the Native Client system.

Figure 2. The hypothetical application of Figure 1 with a trusted storage service.



SRPC or whether it uses shared memory. Note that the storage service must assume that the image library is untrusted. The service is responsible for ensuring that it only services requests consistent with the implied contract with the user. For example, it might enforce a limit on total disk used by the photo application and might further restrict operations to only reference a particular directory.

The Native Client architecture was designed to support pure computation. It is not appropriate for modules requiring process creation, direct file system access, or unrestricted access to the network. Trusted facilities such as storage should generally be implemented outside of Native Client, encouraging simplicity and robustness of the individual components and enforcing stricter isolation and scrutiny of all components. This design choice echoes microkernel operating system design.[1, 4, 12]

With this example in mind we will now describe the design of key NaCl system components in more detail.

2.1. The inner sandbox

Native Client is built around an x86-specific intra-process "inner sandbox." We believe that the inner sandbox is robust; regardless, to provide defense in depth,[5, 8] we have also developed a second "outer sandbox" that mediates system calls at the process boundary. The outer sandbox is substantially similar to prior structures[11, 20] and we will not discuss it in detail here.

The inner sandbox uses static analysis to detect security defects in untrusted x86 code. Previously, such analysis has been challenging for arbitrary x86 code due to such practices as self-modifying code and overlapping instructions. In Native Client we disallow such practices through a set of alignment and structural rules that, when observed, ensure that the native code module can be disassembled reliably, such that all reachable instructions are identified during disassembly. With reliable disassembly as a tool, our validator can then ensure that the executable includes only the subset of legal instructions, disallowing unsafe machine instructions.

The inner sandbox further uses x86 segmented memory to constrain both data and instruction memory references. Leveraging existing hardware to implement these range checks greatly simplifies the runtime checks required to constrain memory references, in turn reducing the performance impact of safety mechanisms.

This inner sandbox is used to create a security subdomain within a native operating system process. With this organization we can place a trusted service runtime subsystem within the same process as the untrusted application module, with a secure trampoline/springboard mechanism to allow safe transfer of control from trusted to untrusted code and vice versa. Although in some cases a process boundary could effectively contain memory and system-call side effects, we believe the inner sandbox can provide better security, as it effectively isolates the native system call interface from untrusted code, thereby removing it from the attack surface. We generally assume that the operating system is not defect free, such that this interface might have exploitable defects. The inner sandbox further isolates any resources that the native operating system might deliberately map into all processes, as commonly occurs in Microsoft Windows. In effect, our inner sandbox not only isolates the system from the native module, but also helps to isolate the native module from the operating system.

2.2. Runtime facilities

The sandboxes prevent unwanted side effects, but some side effects are often necessary to make a native module useful. For interprocess communications, Native Client provides a reliable datagram abstraction, the "Inter-Module Communications" service or IMC. The IMC allows trusted and untrusted modules to send/receive datagrams consisting of untyped byte arrays along with optional "NaCl Resource Descriptors" to facilitate sharing of files, shared memory objects, communication channels, etc., across process boundaries. The IMC can be used by trusted or untrusted modules, and is the basis for two higher-level abstractions. The first of these, the SRPC facility, provides convenient syntax for defining and using subroutines across NaCl module boundaries, including calls to NaCl code from JavaScript in the browser. The second, NPAPI, provides a familiar interface to interact with browser state, including opening URLs and accessing the DOM, that conforms to existing constraints for content safety. Either of these mechanisms can be used for general interaction with conventional browser content, including content modifications, handling mouse and keyboard activity, and fetching additional site content, substantially all the resources commonly available to JavaScript.

As indicated above, the service runtime is responsible for providing the container through which NaCl modules interact with each other and the browser. The service runtime provides a set of system services commonly associated with an application programming environment. It provides `sysbrk()` and `mmap()` system calls, primitives to support a `malloc()/free()` interface or other memory allocation abstractions. It provides a subset of the POSIX threads interface, with some NaCl extensions, for thread creation and destruction, condition variables, mutexes, semaphores, and thread-local storage. Our thread support is sufficiently complete to allow a port of Intel's Thread Building Blocks[21] to Native Client. The service runtime also provides the common POSIX file I/O interface, used for operations on communications channels as well as Web-based read-only content. As the name space of the local file system is not accessible to these interfaces, local side effects are not possible.

To prevent unintended network access, network system calls such as `connect()` and `accept()` are simply omitted. NaCl modules can access the network via JavaScript in the browser. This access is subject to the same constraints that apply to other JavaScript access, with no net effect on network security.

The NaCl development environment is largely based on Linux open source systems and will be familiar to most Linux and Unix developers. We have found that porting existing Linux libraries is generally straightforward, with large libraries often requiring no source changes.

2.3. Attack surface

Overall, we recognize the following as the system components that a would-be attacker might attempt to exploit:

- Browser integration interface
- Inner sandbox: binary validation
- Outer sandbox: OS system-call interception
- Service runtime binary module loader
- Service runtime trampoline interfaces
- IMC communications interface
- NPAPI interface

In addition to the inner and outer sandbox, the system design also incorporates CPU and content blacklists. These mechanisms will allow us to incorporate layers of protection based on our confidence in the robustness of the various components and our understanding of how to achieve the best balance between performance, flexibility, and security.

In the next section we argue that secure implementations of these facilities are possible and that the specific choices made in our own implementation are sound.

3. NATIVE CLIENT IMPLEMENTATION

3.1. Inner sandbox

In this section, we explain how NaCl implements software fault isolation. The design is limited to explicit control flow, expressed with calls and jumps in machine code. Other types of control flow (e.g. exceptions) are managed in the NaCl service runtime, external to the untrusted code, as described with the NaCl runtime implementation below.

Our inner sandbox uses a set of rules for reliable disassembly, a modified compilation tool chain that observes these rules, and a static analyzer that confirms that the rules have been followed. This design allows for a small trusted code base (TCB),[26] with the compilation tools outside the TCB, and a validator that is small enough to permit thorough review and testing. Our validator implementation requires less than 600 C statements (semicolons), including an x86 decoder and `cpuid` decoding. This compiles into about 6000 bytes of executable code (Linux optimized build) of which about 900 bytes are the `cpuid` implementation, 1700 bytes the decoder, and 3400 bytes the validator logic.

To eliminate side effects the validator must address four subproblems:

- Data integrity: no loads or stores outside of data sandbox
- Reliable disassembly
- No unsafe instructions
- Control flow integrity

To solve these problems, NaCl builds on previous work on CISC fault isolation. Our system combines 80386 segmented memory[6] with previous techniques for CISC software fault isolation.[15] We use 80386 segments to constrain data references to a contiguous subrange of the virtual 32-bit address space. This allows us to effectively implement a data sandbox without requiring sandboxing of load and store instructions. VX32[10] implements its data sandbox in a similar fashion. Note that NaCl modules are 32-bit x86 executables. Support for the more recent 64-bit executable model is an area of our ongoing development.

Table 1 lists the constraints Native Client requires of untrusted binaries. Together, constraints C1 and C6 make disassembly reliable. With reliable disassembly as a tool, detection of unsafe instructions is straightforward. A partial list of opcodes disallowed by Native Client includes:

- `syscall` and `int`. Untrusted code cannot invoke the operating system directly.
- All instructions that modify x86 segment state, including `lds`, far calls, etc.
- `ret`. Returns are implemented with a sandboxing sequence that ends with a register-indirect jump.

Apart from facilitating control sandboxing, excluding `ret` also prevents a vulnerability due to a race condition if the return address were checked on the stack. A similar argument requires that we disallow memory addressing modes on indirect `jmp` and `call` instructions. Native Client does allow the `hlt` instruction. It should never be executed by a correct instruction stream and will cause the module to be terminated immediately. As a matter of hygiene, we disallow all other privileged/ring-0 instructions, as they are never required in a correct user-mode instruction stream. We also constrain x86 prefix usage to only allow known useful instructions. Empirically we have found that this

Table 1. Constraints for NaCl binaries.

C1	Once loaded into the memory, the text segment is not writable, enforced by OS-level protection mechanisms during execution.
C2	The binary is statically linked at a start address of zero, with the first byte of text at 128KB.
C3	All indirect control transfers use a `nacljmp` pseudo-instruction (defined below).
C4	The text segment is padded up to the nearest page with at least one `hlt` instruction (0xf4).
C5	The text segment contains no instructions or pseudo-instructions overlapping a 32-byte boundary.
C6	All *valid* instruction addresses are reachable by a fall-through disassembly that starts at the load (base) address.
C7	All direct control transfers target valid instructions.

eliminates certain denial-of-service vulnerabilities related to CPU errata.

The fourth problem is control flow integrity, ensuring that all control transfers in the program text target an instruction identified during disassembly. For each direct branch, we statically compute the target and confirm it is a valid instruction as per constraint C6. Our technique for indirect branches combines 80386 segmented memory with a simplified sandboxing sequence. As per constraints C2 and C4, we use the CS segment to constrain executable text to a zero-based address range, sized to a multiple of 4KB. With the text range constrained by segmented memory, a simple constant mask is adequate to ensure that the target of an indirect branch is aligned mod 32, as per constraints C3 and C5:

```
and     %eax, 0xffffffe0
jmp     *%eax
```

We will refer to this special two instruction sequence as a nacljmp. Encoded as a 3-byte and and a 2-byte jmp it compares favorably to previous implementations of CISC sandboxing.[16, 23] Without segmented memory or zero-based text, sandboxed control flow typically requires two six-byte instructions (an and and an or) for a total of 14 bytes.

Note that this analysis covers explicit, synchronous control flow only. Exceptions are discussed in Section 3.2.

If the validator were excessively slow it might discourage people from using the system. We find our validator can check code at approximately 30 MB/second (35.7 MB in 1.2 seconds, measured on a MacBook Pro with MacOS 10.5, 2.4 GHz Core 2 Duo CPU, warm file-system cache). At this speed, the compute time for validation will typically be small compared to download time, and so is not a performance issue.

We believe this inner sandbox needs to be extremely robust. We have tested it for decoding defects using random instruction generation as well as exhaustive enumeration of valid x86 instructions. We also have used "fuzzing" tests to randomly modify test executables. Initially these tests exposed critical implementation defects, although as testing continues no defects have been found in the recent past. We have also tested on various x86 microprocessor implementations, concerned that processor errata might lead to exploitable defects.[14] We did find evidence of CPU defects that lead to a system "hang" requiring a power-cycle to revive the machine. This occurred with an earlier version of the validator that allowed relatively unconstrained use of x86 prefix bytes, and since constraining it to only allow known useful prefixes, we have not been able to reproduce such problems.

3.2. Exceptions

Hardware exceptions (segmentation faults, floating point exceptions) and external interrupts are not allowed, due in part to distinct and incompatible exception models in Linux, MacOS, and Windows. Both Linux and Windows rely on the x86 stack via %esp for delivery of these events. Regrettably, since NaCl modifies the %ss segment register, the stack appears to be invalid to the operating system, such that it cannot deliver the event and the corresponding process is immediately terminated. The use of x86 segmentation for data sandboxing effectively precludes recovery from these types of exceptions. As a consequence, NaCl untrusted modules apply a failsafe policy to exceptions. Each NaCl module runs in its own OS process, for the purpose of exception isolation. NaCl modules cannot use exception handling to recover from hardware exceptions and must be correct with respect to such error conditions or risk abrupt termination. In a way, this is convenient, as there are very challenging security issues in delivering these events safely to untrusted code.

Although we cannot currently support hardware exceptions, Native Client does support C++ exceptions.[24] As these are synchronous and can be implemented entirely at user level there are no implementation issues. Windows Structured Exception Handling[18] requires nonportable operating system support and is therefore not supported.

3.3. Service runtime

Conceptually, the service runtime is a container for hosting Native Client modules. In our research system, the service runtime is implemented as an NPAPI plugin, together with a native executable that corresponds to the process container for the untrusted module. It supports a variety of Web browsers on Windows, MacOS, and Linux. It implements the dynamic enforcement that maintains the integrity of the inner sandbox and provides resource abstractions to isolate the NaCl application from host resources and operating system interface. It contains trusted code and data that, while sharing a process with the contained NaCl module, are accessible only through a controlled interface. The service runtime prevents untrusted code from inappropriate memory accesses through a combination of x86 memory segment and page protection.

When a NaCl module is loaded, it is placed in a segment-isolated 256MB region within the service runtime's address space. The first 128KB of the NaCl module's address space (NaCl "user" address space) is reserved for initialization by the service runtime. The first 64KB of this 128KB region is read and write protected to detect NULL pointers and to provide for defense-in-depth against unintended 16-bit address calculations. The remaining 64KB contains trusted code that implements our "trampoline" call gate and "springboard" return gate. Untrusted NaCl module text is loaded immediately after this reserved 128KB region. The %cs segment is set to constrain control transfers from the zero base to the end of the NaCl module text. The other segment registers are set to constrain data accesses to the 256MB NaCl module address space.

Because it originates from and is installed by the trusted service runtime, trampoline and springboard code is allowed to contain instructions that are forbidden elsewhere in untrusted executable text. This code, patched at runtime as part of the NaCl module loading process, uses segment register manipulation instructions and the far call instruction to enable control transfers between the untrusted user code and the trusted service runtime code. Since every *0 mod 32* address in the second 64KB of the NaCl user space is a potential computed control flow target, these are our entry points to a table of system-call trampolines. One of these entry points is blocked with a `hlt` instruction, so that the remaining space may be used for code that can only be invoked from the service runtime. This provides space for the springboard return gate.

Invocation of a trampoline transfers control from untrusted code to trusted code. The trampoline sequence resets `%ds` and then uses a `far call` to reset the `%cs` segment register and transfer control to trusted service handlers, reestablishing the conventional flat addressing model expected by the code in the service runtime. Once outside the NaCl user address space, it resets other segment registers such as `%fs`, `%gs`, and `%ss` to reestablish the native code threading environment, fully disabling the inner sandbox for this thread, and loads the stack register `%esp` with the location of a trusted stack for use by the service runtime. Note that the per-thread trusted stack resides outside the untrusted address space, to protect it from attack by other threads in the untrusted NaCl module.

Just as trampolines permit crossing from untrusted to trusted code, the springboard enables crossing in the other direction. The springboard is used by the trusted runtime:

- To transfer control to an arbitrary untrusted address.
- To start a new POSIX-style thread.
- To start the main thread.

Alignment ensures that the springboard cannot be invoked directly by untrusted code. The ability to jump to an arbitrary untrusted address is used in returning from a service call. The return from a trampoline call requires popping an unused trampoline return address from the top of the stack, restoring the segment registers, and finally aligning and jumping to the return address in the NaCl module.

As a point of comparison, we measured the overhead of a “null” system call. The Linux overhead of 156 ns is slightly higher than that of the Linux 2.6 getpid syscall time, on the same hardware, of 138 ns (implemented via the vsyscall table and using the `sysenter` instruction). We note that the user/kernel transfer has evolved continuously over the life of the x86 architecture. By comparison, the segment register operations and far calls used by the NaCl trampoline are somewhat less common, and may have received less consideration over the history of the x86 architecture.

3.4. Communications

The IMC is the basis of communications into and out of NaCl modules. The implementation is built around a *NaCl socket*, providing a bidirectional, reliable, in-order datagram service similar to Unix domain sockets.[13] An untrusted NaCl module receives its first NaCl socket when it is created, accessible from JavaScript via the DOM object used to create it. The JavaScript uses the socket to send messages to the NaCl module, and can also share it with other NaCl modules. The JavaScript can also choose to connect the module to other services available to it by opening and sharing NaCl sockets as NaCl descriptors. NaCl descriptors can also be used to create shared memory regions.

Using NaCl messages, Native Client’s SRPC abstraction is implemented entirely in untrusted code. SRPC provides a convenient syntax for declaring procedural interfaces between JavaScript and NaCl modules, or between two NaCl modules, supporting a few simple types (e.g. int, float, char), arrays of simple types, and NaCl descriptors. Pointers are not supported. Higher-level data representations can easily be layered on top of IMC messages or SRPC.

Our NPAPI implementation is also layered on top of the IMC and supports a subset of the common NPAPI interface. Specific requirements that shaped the current implementation are the ability to read, modify, and invoke properties and methods on the script objects in the browser, support for simple raster graphics, provide the `createArray()` method and the ability to open and use a URL like a file descriptor. We are currently studying some additional refinements to NPAPI for improved portability, performance and safety.[b]

3.5. Developer tools

Building NaCl Modules: We have modified the standard GNU tool chain, using version 4.2.2 of the gcc collection of compilers[c] and version 2.18 of binutils[d] to generate NaCl-compliant binaries. We have built a reference binary from newlib[e] using the resulting tool chain, rehosted to use the NaCl trampolines to implement system services (e.g., `read()`, `brk()`, `gettimeofday()`, `imc_sendmsg()`). Native Client supports an insecure “debug” mode that allows additional file-system interaction not otherwise allowed for secure code.

We modified gcc for Native Client by changing the alignment of function entries (`-falign-functions`) to 32 bytes and by changing the alignment of the targets branches (`-falign-jumps`) to 32 bytes. We also changed gcc to use `nacljmp` for indirect control transfers, including indirect calls and all returns. We made more significant changes to the assembler, to implement Native Client’s block alignment requirements. To implement returns, the assembler ensures that call instructions always appear in the final bytes of a 32-byte block. We also modified the assembler to implement indirect control transfer sequences by expanding the `nacljmp` pseudo-instruction as a properly aligned consecutive block of bytes. To facilitate testing we added support to use a longer `nacljmp` sequence, align the text base, and use an `and` and `or` that uses relocations as masks. This permits testing applications by running them on the command line, and has been used to run the entire gcc C/C++ test suite. We also changed the linker to set the base address of the image as required by the NaCl loader (128KB today).

Apart from their direct use the tool chain also serves to document by example how to modify an existing tools chain to generate NaCl modules. These changes were achieved with less than 1000 lines total to be patched in

[b] See https://wiki.mozilla.org/Plugins: PlatformIndependentNPAPI

[c] See http://gcc.gnu.org

[d] See http://www.gnu.org/software/binutils/

[e] See http://sourceware.org/newlib/

gcc and binutils, demonstrating the simplicity of porting a compiler to Native Client.

Profiling and Debugging: Native Client's open source release includes a simple profiling framework to capture a complete call trace with minimal performance overhead. This support is based on gcc's `-finstrument-functions` code generation option combined with the `rdtsc` timing instruction. This profiler is portable, implemented entirely as untrusted code. In our experience, optimized builds profiled in this framework have performance somewhere between `-O0` and `-O2` builds. Optionally, the application programmer can annotate the profiler output with methods similar to `printf`, with output appearing in the trace rather than stdout.

Our release also includes a modified version of gdb on Linux for Native Client debugging. The debugger recognizes the different addressing domains used by trusted and untrusted code, and independent symbol tables for both domains. Even with this support, the additional complexities of Native Client can interfere with debugging. As such we maintain a set of libraries to facilitate building both standalone and Native Client versions of a project, and commonly debug the standalone version first.

4. EXPERIENCE

Performance measurements in this section are made without the Native Client outer sandbox. The outer sandbox implementations are platform-dependent, and generally use standard kernel facilities (e.g. system call ACLs on Windows, user IDs on Linux) with inherently small incremental overhead.

4.1. SPEC2000

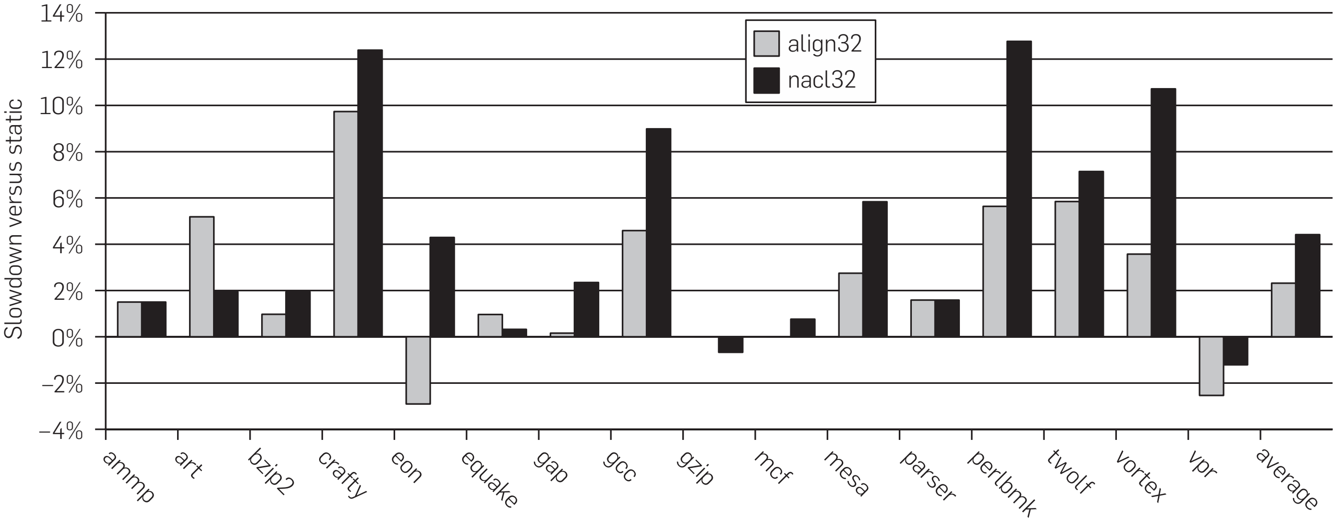
A primary goal of Native Client is to deliver substantially all of the performance of native code execution. NaCl module performance is impacted by alignment constraints, extra instructions for indirect control flow transfers, and the incremental cost of NaCl communication abstractions.

We first consider the overhead of making native code side effect free. To isolate the impact of the NaCl binary constraints (Table 1), we built the SPEC2000 CPU benchmarks using the NaCl compiler, and linked to run as a standard Linux binary. The worst case for NaCl overhead is CPU bound applications, as they have the highest density of alignment and sandboxing overhead. Figure 3 shows the overhead of NaCl compilation for a set of benchmarks from SPEC2000. The worst case performance overhead is crafty at about 12%, with an average of about 5% across all benchmarks. Hardware performance counter measurements indicate that the largest slowdowns are due to instruction cache misses. For crafty, the instruction fetch unit is stalled during 83% of cycles for the NaCl build, compared to 49% for the default build. Gcc and vortex are also significantly impacted by instruction cache misses.

As our current alignment implementation is conservative, aligning some instructions that are not indirect control flow targets, we hope to make incremental code size improvement as we refine our implementation. "NaCl32" measurements use statically linked binaries, 32-byte alignment, and the `nacljmp` pseudo-instruction for indirect control flow transfers. To isolate the impact of the indirect control flow sequence, Figure 3 also shows "align32" results for static linking and 32-byte alignment only. These comparisons make it clear that alignment is a factor in some cases where overhead is significant. Impact from static linking and sandboxing instruction overhead is small by comparison.

The impact of alignment is not consistent across the benchmark suite. In some cases, alignment appears to improve performance, and in others it seems to make things worse. We hypothesize that alignment of branch targets to 32-byte boundaries sometimes interacts favorably with caches, instruction prefetch buffers, and other facets of processor microarchitecture. These effects are curious but not large enough to justify further investigation. In cases where alignment makes performance

Figure 3. SPEC2000 performance. "Align32" results are for binaries with aligned 32-byte instruction blocks. "Nacl32" results are for NaCl binaries. Performance for both is presented relative to standard compilation with static linking.



worse, one possible factor is code size, as mentioned above. Increases in NaCl code size due to alignment can be as much as 50%, especially in programs like the gcc SPEC2000 benchmark with a large number of static call sites. Similarly, benchmarks with a large amount of control flow branching (e.g., crafty, vortex) have a higher code size growth due to branch target alignment. The incremental code size increase of sandboxing with `nacljmp` is consistently small.

Overall, the performance impact of Native Client on these benchmarks is on average less than 5%. At this level, overhead compares favorably to untrusted native execution.

4.2. H.264 decoder

We ported an internal implementation of H.264 video decoding to evaluate the difficulty of the porting effort. The original application converted H.264 video into a raw file format, implemented in about 11K lines of C for the standard GCC environment on Linux. We modified it to play video. The port required about 20 lines of additional C code, more than half of which was for error checking. Apart from rewriting the Makefile, no other modifications were required. This experience is consistent with our general experience with Native Client; legacy Linux libraries that do not inherently require network and file access generally port with minimal effort. Performance of the original and NaCl versions were comparable and limited by video frame-rate.

4.3. Quake

We profiled sdlquake-1.0.9[f] using the built-in "timedemo demo1" command. Quake was run at 640 × 480 resolution on a Ubuntu Dapper Drake Linux box with a 2.4 GHz Intel Q6600 quad core CPU. The video system's vertical sync (VSYNC) was disabled. The Linux executable was built using gcc version 4.0.3, and the Native Client version with nacl-gcc version 4.2.2, both with `-O2` optimization.

With Quake, the differences between Native Client and the normal executable are, for practical purposes, indistinguishable. See Table 2 for the comparison. We observed very little nondeterminism between runs. The test plays the same sequence of events regardless of frame rate. Slight variances in frame rate can still occur due to the OS thread scheduler and pressure applied to the shared caches from other processes. Although Quake uses software rendering, the performance of the final bitmap transfer to the user's desktop may depend on how busy the video device is.

5. DISCUSSION

As described above, Native Client has inner and outer sandboxes, redundant barriers to protect native operating system interfaces. Additional measures such as a CPU blacklist and NaCl module blacklist will also be deployed.

We have developed and tested Native Client on Ubuntu Linux, MacOS, and Microsoft Windows XP. Overall we are satisfied with the interaction of Native Client with these operating systems. That being said, there are areas where better operating system support would help. As an example, popular operating systems require all threads to use a flat addressing model in order to deliver exceptions correctly. Use of segmented memory prevents these systems from interpreting the stack pointer and other essential thread state. Through better operating system segment support we could resolve this problem and provide hardware exception support in untrusted code. However, note that due to our portability requirement we could not enable exception support for untrusted modules unless all native OSes support it. This least-common-denominator effect also arises in other parts of the system, such as the 256MB address space limit for NaCl modules.

With respect to programming languages and language implementations, we are encouraged by our initial experience with Native Client and the GNU tool chain, and are looking at porting other compilers. We have also ported two interpreters, Lua and awk. While it would be challenging to support JITted languages such as Java, we are hopeful that Native Client might someday allow developers to use their language of choice in the browser rather than being restricted to JavaScript.

6. RELATED WORK

Techniques for safely executing third-party code generally fall into four categories: system request moderation, virtualization, fault isolation, and trust with authentication.

Kernel-based mechanisms such as user-id-based access controls, systrace[20] and ptrace[25] are familiar facilities on Unix-like systems. Many projects have applied such mechanisms to containing untrusted code, most recently Android[2] from Google and Xax[9] from Microsoft Research. While they can be very effective, these approaches require dependencies on the native operating system. These dependencies in turn can interfere with portability, and expose more of the native operating system in the attack surface. Our inner sandbox design was heavily influenced by goals of portability and operating system independence.

Many research and practical systems apply abstract virtual machines to constrain untrusted code. While they commonly support ISA portability, they also tend to create a performance obstacle that we avoid by working directly with machine code. A further advantage of expressing sandboxing directly in machine code is that it does not rely on a trusted compiler or interpreter. This greatly reduces the size of the trusted computing base,[26] and has a further

Table 2. Quake performance comparison. Numbers are in frames per second.

Run #	Native Client	Linux Executable
1	143.2	142.9
2	143.6	143.4
3	144.2	143.5
Average	143.7	143.3

[f] From http://www.libsdl.org

benefit in Native Client of opening the system to third-party tool chains.

Native Client applies concepts of software fault isolation that have been extensively discussed in the research literature. Our data integrity scheme is a straightforward application of segmented memory as implemented in the Intel 80386.[6] Our control flow integrity technique builds on the seminal work by Wahbe, Lucco, Anderson, and Graham,[27] also applying techniques described by McCamant and Morrisett.[16]

Perhaps the most prevalent use of native code in Web content is via Microsoft's ActiveX.[7] ActiveX controls rely on a trust model to provide security, with controls cryptographically signed using Microsoft's proprietary Authenticode system,[17] and only permitted to run once a user has indicated they trust the publisher. This dependency on the user making prudent trust decisions is commonly exploited. ActiveX provides no guarantee that a trusted control is safe. Even when the control itself is not inherently malicious, defects in the control can be exploited, often permitting execution of arbitrary code. In contrast, Native Client is designed to prevent such exploitation, even for flawed NaCl modules.

7. CONCLUSION

This paper has described Native Client, a system for incorporating untrusted x86 native code into an application that runs in a Web browser. In addition to creating a barrier against undesirable side effects, Native Client enables modules that are portable both across operating systems and across Web browsers, and it supports performance-oriented features such as threading and vectorization instructions. We believe the NaCl inner sandbox is extremely robust; regardless, we provide additional redundant mechanisms to provide defense-in-depth.

In our experience we have found porting existing Linux/gcc code to Native Client is straightforward, and that the performance penalty for the sandbox is small, particularly in the compute-bound scenarios for which the system is designed.

By describing Native Client here and making it available as open source, we hope to encourage community scrutiny and contributions. We believe this feedback together with our continued diligence will enable us to create a system that achieves improved safety over previous native code Web technologies.

Acknowledgments

Many people have contributed to the direction and the development of Native Client; we acknowledge a few of them here. The project was conceived based on an idea from Matt Papakipos. Jeremy Lau, Brad Nelson, John Grabowski, Kathy Walrath, and Geoff Pike have made valuable contributions to the implementation and evaluation of the system. Thanks also to Danny Berlin, Chris DiBona, and Rebecca Ward. Doug Evans is responsible for our GDB implementation. We thank Sundar Pichai and Henry Bridge for their role in shaping the project direction. We would also like to thank Dick Sites for his thoughtful feedback on an earlier version of this paper.

References

1. Accetta, M., Baron, R., Bolosky, W., Golub, D., Rashid, R., Tevanian, A., Young, M. *Mach: A New Kernel Foundation for UNIX Development*. 1986, 93–112.
2. Burns, J. Developing secure mobile applications for android. http://isecpartners.com/files/iSEC_Securing_Android_Apps.pdf, 2008.
3. Campbell, K., Gordon, L., Loeb, M., Zhou, L. The economic cost of publicly announced information security breaches: empirical evidence from the stock market. *J. Comp. Secur. 11*, 3 (2003), 431–448.
4. Cheriton, D.R. The V distributed system. *Commun. ACM 31* (1988), 314–333.
5. Cohen, F.B. Defense-in-depth against computer viruses. *Comp. Secur. 11*, 6 (1993), 565–584.
6. Crawford, J. Gelsinger, P. *Programming 80386*. Sybex Inc. (1991).
7. Denning, A. *ActiveX Controls Inside Out*. Microsoft Press (May 1997).
8. Directorate for Command, Control, Communications and Computer Systems, U.S. Department of Defense Joint Staff. Information assurance through defense-in-depth. Technical report, Directorate for Command, Control, Communications and Computer Systems, U.S. Department of Defense Joint Staff, Feb. 2000.
9. Douceur, J.R., Elson, J., Howell, J., Lorch, J.R. Leveraging legacy code to deploy desktop applications on the web. In *Proceedings of the 2008 Symposium on Operating System Design and Implementation* (December 2008).
10. Ford, B., Cox, R. Vx32: Lightweight user-level sandboxing on the x86. In *2008 USENIX Annual Technical Conference* (June 2008).
11. Goldberg, I., Wagner, D., Thomas, R., Brewer, E.A. A secure enviroment for untrusted helper applications. In *Proceedings of the 6th USENIX Security Symposium* (1996).
12. Golub, D., Dean, A., Forin, R., Rashid, R. UNIX as an application program. In *Proceedings of the Summer 1990 USENIX Conference* (1990), 87–95.
13. Joy, W., Cooper, E., Fabry, R., Leffler, S., McKusick, K., Mosher, D. 4.2 BSD system manual. Technical report, Computer Systems Research Group, University of California, Berkeley, 1983.
14. Kaspersky, K., Chang, A. Remote code execution through Intel CPU bugs. In *Hack In The Box (HITB) 2008 Malaysia Conference*.
15. McCamant, S., Morrisett, G. Efficient, verifiable binary sandboxing for a CISC architecture. Technical Report MIT-CSAIL-TR-2005-030, 2005.
16. McCamant, S., Morrisett, G. Evaluating SFI for a CISC architecture. In *15th USENIX Security Symposium* (Aug. 2006).
17. Microsoft Corporation. Signing and checking code with Authenticode. http://msdn.microsoft.com/en-us/library/ms537364(VS.85).aspx.
18. Microsoft Corporation. Structured exception handling. http://msdn.microsoft.com/en-us/library/ms680657(VS.85).aspx, 2008.
19. Netscape Corporation. Gecko plugin API reference. http://developer.mozilla.org/en/docs/Gecko_Plugin_API_Reference.
20. Provos, N. Improving host security with system call policies. In *USENIX Security Symposium* (Aug. 2003).
21. Reinders, J. *Intel Thread Building Blocks*. O'Reilly & Associates, 2007.
22. Savage, M. Cost of computer viruses top $10 billion already this year. *ChannelWeb*, Aug. 2001.
23. Small, C. MiSFIT: A tool for constructing safe extensible C++ systems. In *Proceedings of the Third USENIX Conference on Object-Oriented Technologies* (June 1997).
24. Stroustrup, B. *The C++ Programming Language: Second Edition*. Addison-Wesley, 1997.
25. Tarreau, W. ptrace documentation. http://www.linuxhq.com/kernel/v2.4/36-rc1/Documentation/ptrace.txt, 2007.
26. U. S. Department of Defense, Computer Security Center. Trusted computer system evaluation criteria, Dec. 1985.
27. Wahbe, R., Lucco, S., Anderson, T.E., Graham, S.L. Efficient software-based fault isolation. *ACM SIGOPS Oper. Sys. Rev. 27*, 5 (Dec. 1993), 203–216.

Bennet Yee, David Sehr, Gregory Dardyk, J. Bradley Chen, Robert Muth, Tavis Ormandy, Shiki Okasaka, Neha Narula, and Nicholas Fullagar, Google, Inc., Mountainview, CA.

DOI:10.1145/1629175.1629200

Technical Perspective
Schema Mappings: Rules for Mixing Data

By Alon Halevy

WHEN YOU SEARCH for flight tickets on you favorite Web site, your query is often dispatched to tens of databases to produce an answer. When you search for products on Amazon.com, you are seeing results from thousands of vendor databases that were developed before Amazon existed. Did you ever wonder how that happens? What is the theory behind it all? At the core, these systems are powered by *schema mappings* that provide the glue to tie all these databases together. The following paper by ten Cate and Kolaitis will give you a glimpse into the theoretical foundations underlying schema mappings and might even inspire you to work in the area.

The scenarios I've noted here are examples of data management applications that require access to multiple heterogeneous data sets. Data integration is the field that develops architectures, systems, formalisms, and algorithms for combining data from multiple sources, be they relational databases, XML repositories, or data from the Web. The goal of a data integration system is to offer uniform access to a collection of sources, and free the user from having to locate individual sources, learn their specific interaction details, and to manually combine the data. The work on data integration spans multiple fields of computer science, including data management, artificial intelligence systems, and human-computer interaction. The field has been nicknamed the "AI-complete" problem of data management due to the challenges that arise from reconciling multiple models of data created by humans, and the realization that we never expect to solve data integration completely automatically.

Data integration challenges are pervasive in practice. Large enterprises often must combine data from hundreds of repositories, and scientists constantly face an explosion in the number of sources being created in their domain. The Web provides an extreme case of data integration with tens of millions of independently developed data sources. Fortunately, data integration is also a pervasive problem in government organizations, enabling a steady stream of research on the topic. In a nutshell, data integration is difficult because the data sets were developed independently and for different purposes. Therefore different developers model varying aspects of the data, use inconsistent terminology, and make different assumptions on the data.

There are several architectures for data integration systems, and the appropriate choice depends on the need of the application. In some cases it is possible to collect all the data in one physical repository; in other cases data must be exchanged from a source database to a target. In other scenarios, organizational boundaries or other factors dictate that data must be left at the original sources and combination of the relevant data can only occur in response to a query. Regardless of the architecture used, the core of data integration relies on schema mappings that specify how to translate terms (for example, table names and attribute names) between different sources and relate differing database organizations. Much of the effort in building a data integration application is to construct schema mappings and maintain them over time. The main reason building the mappings is difficult is that it requires understanding the semantics of the source and target databases (that may require more than one person), and the ability to express the semantic relationship formally (that may require a database specialist in addition to the domain experts). There has been a large body of research on providing assistance in creating and debugging schema mappings.

A schema mapping must be written in some logical formalism. In the earliest data integration systems, schema mappings were written like ordinary view definitions (now known as GAV mappings), where an integrated view is defined over tables from multiple sources. With time, it became evident that this approach did not scale to a large number of sources, thus LAV mappings were developed. In LAV, the focus is on describing the contents of an individual source irrespective of the other sources. LAV mappings are complemented by a general reasoning engine that infers how to combine data from multiple sources, given a particular query. As this study of mappings progressed, researchers discovered close relationships between mapping formalisms and constraint languages such as tuple-generating dependencies.

Though some properties of these languages in isolation are well understood, this paper sheds significant light, for the first time, on the relationships between these languages. The authors identify general properties of mappings (that are not tied to the formalism in which they are written), and show how these properties can be used to characterize the language that can express a mapping. Except for providing several insightful results, I believe their paper merits careful study because it opens up a new and exciting field of research involving the expressive power of data integration systems. C

Alon Halevy is a research scientist at Google, where he manages a team looking into how structured data can be used in Web search,

DOI:10.1145/1629175.1629201

Structural Characterizations of Schema-Mapping Languages

By Balder ten Cate and Phokion G. Kolaitis

Abstract

Information integration is a key challenge faced by all major organizations, business and governmental ones alike. Two research facets of this challenge that have received considerable attention in recent years are data exchange and data integration. The study of data exchange and data integration has been facilitated by the systematic use of schema mappings, which are high-level specifications that describe the relationship between two database schemas. Schema mappings are typically expressed in declarative languages based on logical formalisms and are chosen with two criteria in mind: (a) expressive power sufficient to specify interesting data interoperability tasks and (b) desirable structural properties, such as query rewritability and existence of universal solutions, that, in turn, imply good algorithmic behavior.

Here, we examine these and other fundamental structural properties of schema mappings from a new perspective by asking: How widely applicable are these properties? Which schema mappings possess these properties and which do not? We settle these questions by establishing structural characterizations to the effect that a schema mapping possesses certain structural properties if and only if it can be specified in a particular schema-mapping language. More concretely, we obtain structural characterizations of schema-mapping languages such as global-as-view (GAV) dependencies and local-as-view (LAV) dependencies. These results delineate the tools available in the study of schema mappings and pinpoint the properties of schema mappings that one stands to gain or lose by switching from one schema-mapping language to another.

1. INTRODUCTION

The aim of information integration is to synthesize information distributed over multiple heterogeneous sources into a single unified format. Information integration has been recognized as a key (and costly) challenge faced by large organizations today (see Bernstein and Haas[3, 12]). It is also well understood[12] that information integration is not a single problem but, rather, a collection of interrelated problems that include extracting and cleaning data from the sources, deriving a unified format for the integrated data, transforming data from the sources into data conforming with the unified format, and answering queries over the unified format. In this article, we focus on *relational* information integration, this is to say, we assume that the sources are databases over (different) relational schemas, called *source* or *local* schemas, and also that the unified format is some other relational schema, called the *target* or the *global* schema. A relational schema or simply a schema consists of names of relations and names of the columns of each relation. A database instance or simply an instance for a given schema is a collection containing, for each relation name in the schema, a finite relation (i.e., a table of records). An example of a source schema and a target schema is given in Figure 1. The source schema consists of three relation names that contain information about direct orders from a manufacturer together with information about retail sales; the target schema consists of a single relation name intended to summarize the sales records. Figure 1 also depicts a source instance and three target instances that will be used later on to illustrate the main concepts.

Two important facets of information integration are data exchange and data integration. Both these facets deal with the attainment of information integration, but they adopt distinctly different approaches. Data exchange is the problem of transforming data residing in different sources into data structured under a target schema; in particular, data exchange entails the materialization of data, after the data have been extracted from the sources and restructured into the unified format. In contrast, data integration can be described as symbolic or virtual integration: users are provided with the capability to pose queries and obtain answers via the unified format interface, while the data remain in the sources and no materialization of the restructured data takes place. Figure 2 depicts the data integration and data-exchange tasks.

In both data exchange and data integration, the relationship between the local schemas and the global schema must be spelled out. One way to accomplish this is via programs or SQL scripts written by human experts; this, however, can be an expensive and error-prone undertaking due to the complexity of the transformations involved. Instead, the research community has introduced *schema mappings*, a higher level

A previous version of this article appeared in the *Proceedings of the 12th International Conference on Database Theory*, 2009.

Figure 1. An example of a schema mapping.

Source database schema S:

DirectCustomer(cust-id, name, address)
DirectOrder(cust-id, date, prod, quant)
Retail(store-id, date, prod, quant)

Target database schema T:

Sales(date, cust, prod, quant)

Source database instance I:

DirectCustomer

cust-id	name	address
c1	UCSC	1156 High St, Santa Cruz, CA 95060

DirectOrder

cust-id	date	prod	quant
c1	05-01-2009	Quadcore-9950-PC	100
c1	05-01-2009	TFT-933SN-Wide	100

Retail

store-id	date	prod	quant
s1	05-03-2009	Quadcore-9950-PC	1

A target database instance J_1

Sales

date	cust	prod	quant
05-01-2009	UCSC	Quadcore-9950-PC	100
05-01-2009	UCSC	TFT-933SN-Wide	100
05-03-2009	N_1	Quadcore-9950-PC	1

A second target database instance J_2

Sales

date	cust	prod	quant
05-01-2009	UCSC	Quadcore-9950-PC	100
05-01-2009	UCSC	TFT-933SN-Wide	100
05-03-2009	UCLA	Quadcore-9950-PC	1

A third target database instance J_3

Sales

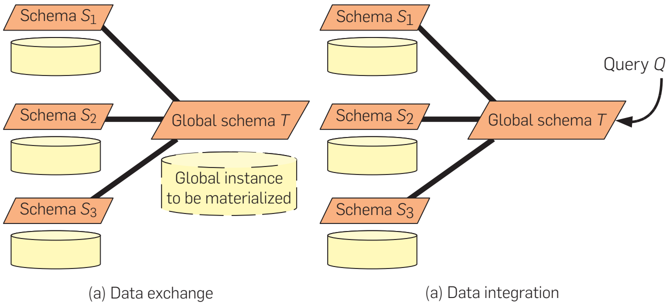
date	cust	prod	quant
05-01-2009	UCSC	Quadcore-9950-PC	100
05-03-2009	N_1	Quadcore-9950-PC	1

Schema mapping

$$\forall x,y,z,u,v,w\,(\text{DirectCustomer}(x,y,z) \wedge \text{DirectOrder}(x,u,v,w) \rightarrow \text{Sales}(u, y, v, w))$$

$$\forall x,y,z,v,w\,(\text{Retail}(x, y, v, w) \rightarrow \exists N\,\text{Sales}(y, N, v, w))$$

Figure 2. Data exchange and data integration.



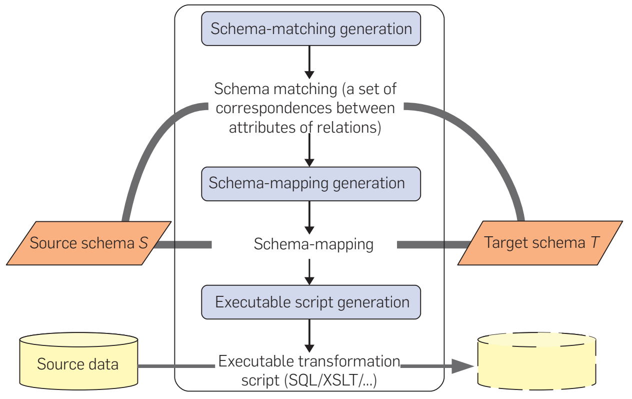
(a) Data exchange (a) Data integration

of abstraction that makes it possible to separate the specification of the relationship between the schemas from the actual implementation of the transformations. Schema mappings are declarative specifications that describe the relationship between two database schemas. In recent years, schema mappings have been used extensively in specifying data interoperability tasks and are regarded as the essential building blocks in data exchange and data integration (see, e.g., the surveys[14, 15]). The use of schema mappings helps the user understand and reason about the relationship between the source schemas and the target schema; furthermore, schema mappings can be automatically compiled into executable scripts in various languages.

A concrete example of a schema mapping is given in Figure 1. The schema mapping is specified by two sentences of first-order logic. The first sentence asserts that whenever the source relation DirectCustomer contains a triple (v_1, v_2, v_3) of values and the source relation DirectOrder contains a quadruple (v_1, v_4, v_5, v_6) of values so that the values for cust-id in these two tuples coincide, then the target relation Sales must contain the quadruple (v_1, v_2, v_4, v_5). The second sentence asserts that whenever the source relation Retail contains a quadruple (w_1, w_2, w_3, w_4) of values, then there must exist some value V so that the target relation Sales contains the quadruple (w_2, V, w_3, w_4). Clearly, the pair (I, J_1) consisting of the source relation I and the target relation J_1 satisfies both these formulas; the same holds true for the pair (I, J_2). In contrast, the pair (I, J_3) fails to satisfy the first formula because the Sales relation does not contain the required quadruple (05-01-2009, UCSC, TFT-933SN-Wide, 100).

This example also unveils some of the main conceptual and algorithmic issues arising in data exchange and data integration. On the data exchange side, suppose that we wish to transform the above source instance I to a target instance J according to the schema mapping in Figure 1. There are at least two distinct target instances that, together the source instance I, satisfy the specification of the schema mapping; in fact, it is easy to see that there are *infinitely* many such target instances. So, which one should we choose to materialize? What makes one target instance a "better" candidate to materialize than another, and how can one be computed? As we shall see, *universal solutions* turn out to be the preferred target instances to materialize. On the data integration side, suppose that a user poses a query over the target schema. Different answers may be obtained by evaluating the query on different target instances that (together with the given source instance) satisfy the schema-mapping specification. So, what is the "right" semantics of target queries in data integration? Is it possible to rewrite target queries into queries over the source schema so that they can be evaluated directly against the given source instance? This will lead us to the notions of the *certain answers* and of *allowing for query rewriting*. In the next section, we will introduce some of the most commonly used schema-mapping languages, including global-as-view (GAV) dependencies and local-as-view (LAV) dependencies, and we will find out that schema mappings specified in these formalisms indeed admit universal solutions and allow for rewriting of the most frequently asked

Figure 3. Architecture of the Clio data-exchange system.


Schema-matching generation
Schema matching (a set of correspondences between attributes of relations)
Schema-mapping generation
Source schema *S*
Schema-mapping
Target schema *T*
Executable script generation
Source data
Executable transformation script (SQL/XSLT/...)

queries in relational databases (see Figure 2).

A system that makes systematic use of schema mappings is Clio, a data-exchange system that started as a research prototype at the IBM Almaden Research Center and is now part of IBM's suite of information integration tools.[13] The architecture of Clio is depicted in Figure 3. The system has a *schema-matching* component, a *schema-mapping generation* component, and an *executable-script generation* component. The schema-matching component produces a set of correspondences between attributes of relations in a source schema and a target schema; these correspondences are derived automatically or semi-automatically through an interaction with the user and via a graphical user interface that allows the user to intervene and make changes. The schema-mapping generation component takes as input these attribute correspondences and returns a schema mapping. In general, there are more than one schema mapping that are consistent with a set of attribute correspondences. Clio produces just one of these possible schema mappings but the user can again intervene and edit the schema mapping returned by the system. Finally, the executable-script generation component automatically transforms this schema mapping into a set of scripts in some language, such as SQL or XSLT.

2. SCHEMA MAPPINGS AND LANGUAGES

In this section, we define the basic notions about schema mappings and present some of the main schema-mapping languages studied by the research community.

2.1. Basic notions

A *(relational) schema* is a tuple $\mathbf{R} = (R_1, \ldots, R_n)$ of relation symbols each of which has a fixed arity (number of attributes). An $\mathbf{R}$-*instance* is a tuple $I = (R_1^I, \ldots, R_n^I)$ of finite relations, whose arities match those of the relation symbols of $\mathbf{R}$. A *fact* of I is an expression $R_i\mathbf{a}$, where $i \le n$ and $\mathbf{a}$ is a tuple of values belonging to the relation R_i^I. The *active domain* of I, denoted by *adom* (I), is the set of all values occurring in the relation R_i^I, for $1 \le i \le n$. We will be usually working with two disjoint schemas, called the *source schema* $\mathbf{S} = (S_1, \ldots, S_n)$ and the *target schema* $\mathbf{T} = (T_1, \ldots, T_m)$. An $\mathbf{S}$-instance is called a *source instance* and a $\mathbf{T}$-instance is called a *target instance*. Whenever we consider a pair of instances (I, J), it is understood that I is a source instance and J is a target instance.

As stated earlier, a schema mapping is a declarative specification that describes the relationship between two schemas. More precisely, a *schema mapping* is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema disjoint from $\mathbf{S}$, and Σ is a set of sentences (i.e., formulas with no free variables) in some logical formalism. This is the *syntactic* view of schema mappings. There is also a complementary *semantic* view of schema mappings that we present next. Let I be a source instance and J a target instance. We say that J is a *solution for I w.r.t. a schema mapping* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ if $(I, J) \models \Sigma$, which means that (I, J) satisfies every sentence in Σ. Consequently, from a semantic standpoint, a schema mapping $\mathcal{M}$ can be thought of as a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$, where $\mathcal{W}$ is the set of all pairs (I, J) such that J is a solution for I w.r.t. $\mathcal{M}$. So, as a semantic object, a schema mapping $\mathcal{M}$ is triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$, where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema disjoint from $\mathbf{S}$, and $\mathcal{W}$ is a collection of pairs (I, J) with I a source instance and J a target instance.

Let L be a logical language, let Σ be a set of L-sentences, and let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$ be a schema mapping given as a semantic object. We say that $\mathcal{M}$ is *L-definable by* Σ if for every source instance I and every target instance J, we have that $(I, J) \in \mathcal{W}$ if and only if $(I, J) \models \Sigma$. When we work with schema mappings in the sequel, it will be clear from the context if the schema mapping at hand is viewed as a syntactic object or as a semantic one.

Example 2.1. To illustrate these notions, consider the schema mapping in Figure 1. In this example, the source schema $\mathbf{S}$ consists of the relations DirectCustomer, DirectOrder, and Retail, while the target schema $\mathbf{T}$ consists of the single relation Sales. The relationship between source and target is then described by the schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ consists of the two first-order sentences listed in Figure 1. As a semantic object, $\mathcal{M}$ is the triple $(\mathbf{S}, \mathbf{T}, \mathcal{W})$, where $\mathcal{W}$ consists of all pairs (I, J) satisfies the two sentences in Σ. In particular, the pairs (I, J_1) and (I, J_2) belong to $\mathcal{W}$, but the pair (I, J_3) does not. In other words, J_1 and J_2 are solutions for I w.r.t. to $\mathcal{M}$, but J_3 is not.

2.2. Schema-mapping languages

What is a "good" language for specifying schema mappings? To address this question, let us reflect on the relational database model, introduced by E.F. Codd 40 years ago.[4] One of the reasons for the success of this model is that it supports powerful, high-level database query languages, such as *relational algebra* and *relational calculus*, that have formed the foundation for SQL. Relational algebra and relational calculus have the same expressive power as first-order logic[5]; in fact, relational calculus is a syntactic variant of first-order logic tailored for databases. Thus, at first sight, it is natural to ask: can first-order logic also be used as a schema-mapping language? It is not hard to show, however, that basic algorithmic problems in data integration and data exchange, such as existence-of-solutions and query answering, become *undecidable* if unrestricted use of first-order logic is allowed in specifying the relationship between database schemas (intuitively, this is so because such tasks involve testing first-order sentences for satisfiability, since they quantify over all solutions of a source instance). Consequently, we have to identify proper sublanguages of first-order logic that strike a good balance between expressive power for data interoperability purposes and algorithmic properties. Towards this goal, let us consider the following basic tasks that every schema-mapping language ought to support.

- **Copy (Nicknaming):** Copy a source relation into a target relation and rename it.
- **Projection (Column Deletion):** Form a target relation by deleting one or more columns of a source relation.
- **Augmentation (Column Addition):** Form a target relation by adding one or more columns to a source relation.
- **Decomposition:** Decompose a source relation into two or more target relations.
- **Join:** Form a target relation by joining two or more source relations.
- **Combinations of the above:** For example, combine join with column augmentation.

These tasks can be easily specified in first-order logic, as shown next. For concreteness, we use relations of arities two or three.

Copy	$\forall x, y, z(P(x, y, z) \rightarrow T(x, y, z))$
Projection	$\forall x, y, z(P(x, y, z) \rightarrow U(x, y))$
Augmentation	$\forall x, y(R(x, y) \rightarrow \exists z T(x, y, z))$
Decomposition	$\forall x, y, z(P(x, y, z) \rightarrow U(x, y) \wedge V(y, z))$
Joint	$\forall x, y, z(R(x, z) \wedge S(z, y) \rightarrow T(x, y, z))$
Combinations of the above	$\forall x, y, z(R(x, z) \wedge S(z, y) \rightarrow \exists t(U(x, t) \wedge W(x, y, z, t)))$

Observe that the above formulas have a striking syntactic similarity. As a matter of fact, they all belong to a class of first-order formulas called *source-to-target tuple generating dependencies* that we define in what follows.

- If $\mathbf{R} = (R_1, \ldots, R_n)$ is a relational schema, then an *atomic formula over* **R** is an expression of the form $R_i(\mathbf{x})$, where $i \leq n$ and **x** is a tuple of variables of length equal to the arity of R_i.
- A *tuple-generating dependency* (*tgd*) is a first-order formula of the form

$$\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})),$$

where **x** and **y** are tuples of variables, $\phi(\mathbf{x})$ is a conjunction of atomic formulas with variables in **x**, each variable in **x** occurs in at least one conjunct of $\phi(\mathbf{x})$, and $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas with variables among those in **x** and **y**.

A *full tgd* is a tgd with no existential quantifiers in the right-hand side, i.e., it is of the form $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$.

- A *source-to-target tuple-generating dependency* (*s-t tgd*) is a tgd such that $\phi(\mathbf{x})$ is a conjunction of atomic formulas over a source schema **S** and $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas over a target schema **T**.

Informally, s-t tgds assert that if a pattern of facts appears in the source, then another pattern of facts must appear in the target. They are also known as *global-and-local-as-view* (*GLAV*) dependencies. In recent years, s-t tgds have been studied extensively in the context of data exchange and data integration[14, 15] because, in spite of their syntactic simplicity, they can express many data interoperability tasks arising in applications. Furthermore, s-t tgds have desirable structural properties that we will discuss in the next section. The following two types of dependencies are important special cases of s-t tgds.

- A *GAV* dependency is an s-t tgd in which the right-hand side of the implication consists of a single atomic formula. Thus, a GAV dependency is of the form

$$\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow U(\mathbf{x}'))$$

with $\phi(\mathbf{x})$ a conjunction of atomic formulas over a source schema and $U(\mathbf{x}')$ an atomic formula over a target schema such that the variables in $\mathbf{x}'$ are among those in **x**.

- A *LAV* dependency is an s-t tgd in which the left-hand side of the implication is a single atomic formula. Thus, a LAV dependency is of the form

$$\forall \mathbf{x}(R(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$$

with $R(\mathbf{x})$ an atomic formula over a source schema and $\psi(\mathbf{x}, \mathbf{y})$ a conjunction of atomic formulas over a target schema.

Consider again the schema mapping in Figure 1. The first sentence used to specify that schema mapping is a GAV dependency, while the second one is a LAV dependency. Note also that the expressions for the Copy and the Projection tasks are both GAV and LAV dependencies, the expressions for the Augmentation and the Decomposition tasks are LAV dependencies, and the expression for the Join task is a GAV dependency. The expression for the Decomposition task is a full tgd. It is easy to see that every full tgd is logically equivalent to finitely many GAV dependencies. For example, the full tgd $\forall x, y, z(P(x, y, z) \rightarrow U(x, y) \wedge V(y, z))$ for the Decomposition task is logically equivalent to the set consisting of the GAV dependencies $\forall x, y, z(P(x, y, z) \rightarrow U(x, y))$ and $\forall x, y, z(P(x, y, z) \rightarrow V(y, z))$.

Before being used in data exchange and data integration, tuple-generating dependencies had been investigated in depth in the context of *dependency theory* during the 1970s and the 1980s. Dependency theory is the study of integrity constraints in databases; in this context, tuple-generating dependencies possess desirable algorithmic properties and contain as special cases many well-known classes of integrity constraints in databases, such as inclusion dependencies, join dependencies, and multi-valued dependencies.[10] It is also interesting to note that tuple-generating dependencies seemed to have first appeared (at least in implicit form) a very long time ago. Indeed, in a recent article[2] presenting a formal system for Euclid's Elements, the authors argue convincingly that the theorems in the Elements can be expressed using tuple-generating dependencies! Intuitively, this is so because the typical theorem of Euclidean Geometry states that if a certain pattern between geometric objects (points, lines, triangles, or circles) exists, then another pattern between geometric objects must also exist.

3. PROPERTIES OF SCHEMA MAPPINGS

In this section, we present several structural properties of schema mappings, which have been widely used in the literature on data exchange and data integration. They will turn out to play a key role in our characterizations of schema-mapping languages. Before presenting these properties, however, we need to introduce some important notions in data exchange and data integration.

Homomorphisms. A central notion in the study of schema mappings is that of a *homomorphism*. Let K and K' be two instances over the same schema $\mathbf{R} = (R_1, \ldots, R_n)$. A homomorphism for K to K' is a function from the active domain of K to the active domain of K' such that if $(a_1, \ldots, a_m) \in R_i^k$, then $(h(a_1, \ldots, a_m)) \in R_i^{k'}$, for $i = 1, \ldots, n$.

A homomorphism h is said to be *constant* on a set X if h restricted to $X \cap dom\ (h)$ is the *identity* function, i.e., $h(x) = x$, if $x \in X$ and $h(x)$ is defined. Here, we will consider homomorphisms between target instances and we will require that they are constant on the active domain of some source instance. The rationale behind this requirement is as follows. Typically, when data is exchanged from source to target, the elements in the active domain of a given source instance are "known" values. The schema mapping at hand, however, may under-specify the relationship between source and target. In turn, this may force using "unknown" values, called *labeled nulls*, to materialize solutions for a given source instance. Thus, target instances may have both "known" values, originating from the source instance, and "unknown" values, chosen freshly and acting as placeholders. An example of labeled null is the value N_1 in the target instance J_1 in Figure 1. Homomorphisms are required to leave "known" values untouched, but are free to replace an "unknown" value by another, "known" or "unknown", value.

Universal Solutions. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. Recall that if I is a source instance, then a solution for I w.r.t. $\mathcal{M}$ is a target instance J such that $(I, J) \models \Sigma$. In general, a source instance I may have multiple solutions and, in fact, infinitely many; in particular, this holds true for schema mappings $\mathcal{M}$ specified by s-t tgds because if J is a solution for I w.r.t. $\mathcal{M}$, then every instance J' containing J (as a set of facts) is also a solution for I. Which among those solutions should one materialize if I is to be transformed into a target instance? To address this question, the following concept of a *universal* solution was introduced in Fagin et al.[7]

Let $\mathcal{M} = (\mathrm{S}, \mathrm{T}, \mathcal{W})$ be a schema mapping and let I be a source instance. A target instance J is a *universal solution for I* if

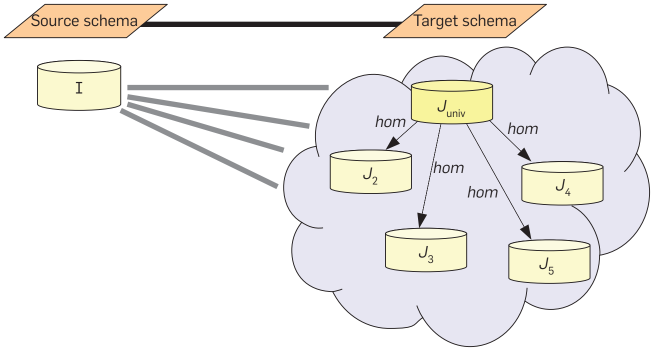
1. J is a solution for I.
2. For each target instance J' that is a solution for I, there is a homomorphism h: $J \rightarrow J'$ that is constant on adom (I).

The intuition behind this concept, which is illustrated in Figure 4, is that a universal solutions is a "most general" solution in the sense that it contains no more and no less information than that specified by the given schema mapping.

EXAMPLE 3.1. Consider the source instance I in Figure 1. One solution for I with respect to the given schema mapping is the target instance J_1. Note that N_1 is a value that does not occur in I, and therefore is interpreted as a labeled null. Another solution for I is the target instance J_2, which is the same as J_1 except that labeled null value N_1 has been replaced by UCLA. However, J_2 contains information that is not implied by the schema mapping, namely, that the customer of the order on May 3, 2009 is UCLA, and, consequently, it is not a universal solution. Indeed, there is a homomorphism from J_1 to J_2 constant on the active domain of I (N_1 is mapped to UCLA), but not vice versa.

Queries and Certain Answers. Informally, a *query* is a question that a user poses against a database. More formally, a *query* q takes as input an instance K and returns as output a relation $q(K)$ of fixed arity with values from the active domain of K. Suppose now that we have a schema mapping between a source schema and target schema. Suppose also that a query q over the target schema is posed and that a source instance I is given. What does answering q using the source instance I mean? As seen earlier, there may be infinitely many solutions J for a given source instance I; furthermore, if q is evaluated on different solutions J for I, it is

Figure 4. A universal solution.



possible that different answers are obtained each time. This ambiguity raises the conceptual problem of giving precise semantics to query answering in data exchange and data integration. The approach taken by the research community is to adopt the *certain answers* semantics, a semantics that originated in the study of incomplete databases (see van der Meyden[19]).

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping and q a query over the target schema $\mathbf{T}$. If I is source instance, then *the certain answers of q on I with respect to $\mathcal{M}$*, denoted $\mathsf{certain}_{\mathcal{M}}(q)(I)$, is the set

$$\mathsf{certain}_{\mathcal{M}}(q)(I) = \cap\{q(J) : J \text{ is a solution for } I \text{ w.r.t. } \mathcal{M}\}.$$

The certain answers semantics provides the guarantee that if a tuple $\mathbf{t}$ belongs to $\mathsf{certain}_{\mathcal{M}}(q)(I)$, then $\mathbf{t}$ belongs to the result $q(J)$ of q on *every* solution J for I. It is easy to see that every tuple in $\mathsf{certain}_{\mathcal{M}}(q)(I)$ is a tuple of values from I. Thus, every schema mapping $\mathcal{M}$ induces a (semantic) transformation $\mathsf{certain}_{\mathcal{M}}$ from queries over the target schema to queries over the source schema, so that if q is query over the target schema, then this transformation associates to it the query $\mathsf{certain}_{\mathcal{M}}(q)$ over the source schema that has the same arity as q and is defined by $\mathsf{certain}_{\mathcal{M}}(q)(I) = \cap\{q(J) : J \text{ is a solution for } I \text{ w.r.t. } \mathcal{M}\}$.

On the face of it, the certain answers semantics is non-effective, since evaluating $\mathsf{certain}_{\mathcal{M}}(q)(I)$ may entail computing the intersection of infinitely many sets. For many frequently asked queries, however, efficient algorithms for evaluating their certain answers exist.

A conjunctive query is a first-order formula of the form $\exists \mathbf{w} \chi(\mathbf{x}, \mathbf{w})$, where $\chi(\mathbf{x}, \mathbf{w})$ is a conjunction of atoms and/or equalities. Conjunctive queries are the most frequently asked queries in relational databases. They are also known as *project select-join* (SPJ) queries because they are precisely the queries that are expressible in relational algebra using the selection, projection, and join operations; in particular, conjunctive queries are easily expressible in SQL using the SELECT FROM WHERE construct. A *union of conjunctive queries* is a finite disjunction of conjunctive queries; equivalently, they are the queries expressible in relational algebra using the selection, projection, join, and union operations.

In information integration, the main approach to computing the certain answers is to try to rewrite queries over the target to queries over the source. In data exchange, one would like to take advantage of the materialized solution and use it to obtain the certain answers of target queries. Both approaches give rise to polynomial-time algorithms for computing the certain answers of unions of conjunctive queries. In particular, as shown in Fagin et al.,[7] the certain answers of unions of conjunctive queries can be obtained using universal solutions. Specifically, let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping such that Σ is a finite set of s-t tgds and let q be a union of conjunctive queries over $\mathbf{T}$. If I is a source instance and J is a universal solution for I, then $\mathsf{certain}_{\mathcal{M}}(q)(I) = q(J)_{\downarrow}$, where $q(J)_{\downarrow}$ is the result obtained by first computing $q(J)$ and then keeping only those tuples that contain values from the active domain of I only.

Example 3.2. Returning to the example schema mapping $\mathcal{M}$ and source instance I from Figure 1, consider the conjunctive query q over the target schema given by

$$q(x, y) = \exists name, date, n, m(Sales(name, date, x, n) \wedge Sales(name, date, y, m))$$

It asks for all pairs of products (x, y), such that some customer bought x and y (in some quantities) on the same date. It is not hard to see that, for every solution J of I with respect to $\mathcal{M}$, the pair (Quadcore-9950-PC,TFT-933SN-Wide) belongs to $q(J)$. In other words, this tuple belongs to the certain answers of q in I with respect to $\mathcal{M}$. It turns out that the certain answers of q on a source instance I are precisely the answers to $q'(I)$, where q' is the following union of conjunctive queries over the source schema:

$$\begin{gathered} q'(x,y) = (\exists cid_1, cid_2, name, addr_1, addr_2, date, n, m \\ (DirectOrder(cid_1, date, x, n) \wedge DirectOrder(cid_2, date, y, m) \wedge \\ DirectCustomer(cid_1, name, addr_1) \wedge \\ DirectCustomer(cid_2, name, addr_2))) \\ \vee (x = y \wedge \exists sid, date, n\, Retail(sid, date, x, n)) \end{gathered}$$

Note that the *Retail* relation does not provide information about the name of the buyer, and therefore, can only contribute identity pairs to the certain answers of q.

Alternatively, the certain answers of q on I can be computed by evaluating q on the universal solution J_1 of I that we discussed in Example 3.1, and keeping only those tuples that contain only values from the active domain of I.

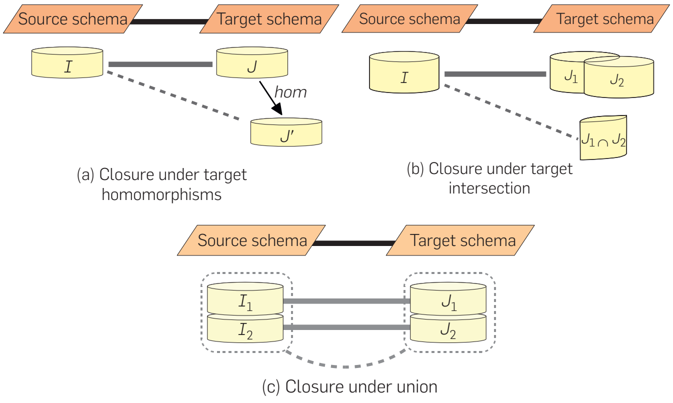
Structural Properties of Schema Mappings. Recall that a schema mapping $\mathcal{M}$ can be viewed as a syntactic object or as a semantic one. As a syntactic object, $\mathcal{M}$ is given by a triple $(\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a set of sentences in some logical formalism; as a semantic object, $\mathcal{M}$ is given by a triple $(\mathbf{S}, \mathbf{T}, \mathcal{W})$, where $\mathcal{W}$ is a set of pairs (I, J) with I a source instance and J a target instance. In what follows, whenever we write that $(I, J) \in \mathcal{M}$, we mean that $(I, J) \models \Sigma$ or that $(I, J) \in \mathcal{W}$ depending on whether $\mathcal{M}$ is given as a syntactic object or a semantic one.

We are now ready to present the structural properties of schema mappings that will play a key role in our characterization results. We begin with three such properties that have been widely used in both data exchange and data integration.

Definition 3.3. *Let $\mathcal{M}$ be a schema mapping.*

- **Closure under target homomorphisms**: *We say that* $\mathcal{M}$ is closed under target homomorphisms *if for all $(I, J) \in \mathcal{M}$ and for all homomorphisms $h: J \to J'$ that are constant on $adom(I)$, we have that $(I, J') \in \mathcal{M}$. (see Figure 5a).*
- **Admitting universal solutions**: *We say that $\mathcal{M}$* admits universal solutions *if for each source instance I there is a universal solution J for I w.r.t. $\mathcal{M}$.*
- **Allowing for conjunctive query rewriting**: *We say that* $\mathcal{M}$ allows for conjunctive query rewriting *if for each union q of conjunctive queries over the target schema, the*

Figure 5. Closure properties of schema mappings.



(a) Closure under target homomorphisms

(b) Closure under target intersection

(c) Closure under union

certain answers query $\text{certain}_{\mathcal{M}}(q)$ *is definable by a union of conjunctive queries over the source schema. In other words, there is a union q' of conjunctive queries over the source schema such that* $\text{certain}_{\mathcal{M}}(q)(I) = q'(I)$, *for every source instance I.*

The first two conditions of closure under target homomorphisms and admitting universal solutions go very well together. As was observed in Fagin et al.,[7] if a schema mapping is closed under target homomorphisms and admits universal solutions, then, for every source instance I, the (typically infinite) space of all solutions of I w.r.t. $\mathcal{M}$ can be completely described by just a single target instance J, namely, by any universal solution J for I. This is so because if J is universal for I and $\mathcal{M}$ is closed under target homomorphisms, then for every target instance J', we have that J' is a solution for I if and only if there is a homomorphism $h: J \rightarrow J'$ that is constant on $adom(I)$. We mention in passing that these two conditions together also imply the existence of *core* universal solutions, which are the smallest universal solutions (see Fagin et al.[8]). Thus, these two conditions lie at the foundation of *data exchange*. The third condition of allowing for conjunctive query rewriting is important in the context of *data integration*, since it implies that the certain answers of unions of conjunctive queries over the target are computable in polynomial time (in the sense of data complexity).

It is well known that all three conditions of closure under target homomorphisms, admitting universal solutions, and allowing for conjunctive query rewriting are possessed by every schema mapping $\mathcal{M}$ definable by a finite set of s-t tgds. Closure under homomorphisms follows easily from the definitions; admitting universal solutions was shown in Fagin et al.[7] using the *chase procedure*. In the case of GAV dependencies, a union of conjunctive queries over the target is easily transformed to a union of conjunctive queries over the source by simply replacing each target relation symbol P by a union of conjunctive queries over the source that defines P. In the case of arbitrary s-t tgds, allowing for conjunctive query rewriting is proved by first "decomposing" the given s-t tgds into GAV dependencies and to LAV dependencies, and then applying results from Abiteboul and Duschka[1] and Duschka and Genesereth.[6] We collect these results into one theorem.

THEOREM 3.4. *Every schema mapping definable by a finite set of s-t tgds is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting.*

Next, we define three additional properties of schema mappings.

DEFINITION 3.5. *Let $\mathcal{M}$ be a schema mapping.*

- **Closure Under Target Intersection**: *We say that $\mathcal{M}$* is closed under target intersection *if for all source instances I and all target instances J_1, J_2 if $(I, J_1) \in \mathcal{M}$ and $(I, J_2) \in \mathcal{M}$, then also $(I, J_1 \cap J_2) \in \mathcal{M}$. (See Figure 5b.)*
- **Closure Under Union**: *We say that $\mathcal{M}$ is* closed under union *if $(\emptyset, \emptyset) \in \mathcal{M}$, and for all $(I_1, J_1) \in \mathcal{M}$, and $(I_2, J_2) \in \mathcal{M}$, we have that also $(I_1 \cup I_2, J_1 \cup J_2) \in \mathcal{M}$. (See Figure 5c.)*
- ***n*-Modularity**: *Let n be a positive integer. We say that $\mathcal{M}$ is* n-modular *if whenever a pair (I, J) does not belong to $\mathcal{M}$, there is a sub-instance $I' \subseteq I$ such that $|adom(I')| \leq n$ and (I', J) does not belong to $\mathcal{M}$.*

Intuitively, a schema mapping is closed under union if solutions can be constructed in a "modular" fashion, i.e., on a tuple-by-tuple basis. Similarly, n-modularity asserts that if $(I, J) \notin \mathcal{M}$, then there is a concise explanation for this fact; this property can also be viewed as a relaxation of closure under union.

We now give several useful propositions about the properties we just introduced.

PROPOSITION 3.6. *Let $\mathcal{M}$ be a schema mapping.*

- *If $\mathcal{M}$ is definable by a finite set of GAV dependencies, then $\mathcal{M}$ is closed under target intersection.*
- *If $\mathcal{M}$ is definable by a finite set of LAV dependencies, then $\mathcal{M}$ is closed under union.*
- *If $\mathcal{M}$ is definable by a finite set of s-t tgds, then $\mathcal{M}$ is n-modular for some $n \geq 1$.*

PROOF. The first two parts follow easily from the definitions. For the third part, assume that $\mathcal{M}$ is a schema mapping definable by a finite set Σ of s-t tgds. Let n be the maximum number of variables occurring in the left-hand side of the s-t tgds in Σ. We claim that $\mathcal{M}$ is n-modular. Assume that $(I, J) \notin \mathcal{M}$. Then there is some s-t tgd $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$ from Σ and a tuple a of values from $adom(I)$ such that $(I, J) \models \phi(\mathbf{a}) \wedge \neg \exists \mathbf{y} \psi(\mathbf{a}, \mathbf{y})$. Now, let I' be the sub-instance of I containing only the values $\mathbf{a}$. Then, it is still the case that $(I', J) \models \phi(\mathbf{a}) \wedge \neg \exists \mathbf{y} \psi(\mathbf{a}, \mathbf{y})$, and hence $(I', J) \notin \mathcal{M}$. ❑

4. LANGUAGE CHARACTERIZATIONS

This section contains the main technical results of the paper, which yield structural characterizations of the various schema-mapping languages considered in earlier sections. We begin with schema mappings specified by LAV dependencies.

Theorem 4.1. *For all schema mappings $\mathcal{M}$, the following are equivalent:*

1. *$\mathcal{M}$ is definable by a finite set of LAV dependencies.*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and is closed under union.*

Proof. The implication (1) ⇒ (2) follows from Theorem 3.4 and Proposition 3.6. We now prove the implication (2) ⇒ (1). The idea behind the proof is that, since $\mathcal{M}$ is closed under union, universal solutions for source instances I can be constructed out of universal solutions for parts of I. This implies that, in defining our schema mapping, we only need to take into account of finite number of source instances up to isomorphism, namely, those that contain precisely one tuple. In what follows, we will make this idea precise.

Suppose that $\mathcal{M}$ satisfies the listed conditions. Let $R_1, \ldots, R_n$ be the relations of the source schema, and let D be a set consisting of k distinct values, with $k = \max_{i \le n} \mathsf{arity}(R_i)$. Let facts be the set of all possible facts, of the form $R_i(d_1, \ldots, d_l)$ with $i \le n$, $l = \mathsf{arity}(R_i)$, and $d_1, \ldots, d_l \in D$. For each $\alpha \in \mathsf{facts}$, let I_α be a the source instance containing only the fact α, and let J_α be a universal solution for I_α. Let $PosDiag_{I_\alpha}(\mathbf{x})$ be the *positive diagram* of I_α, i.e., the conjunction of all facts true in I (which consists of precisely one fact) and let $PosDiag_{J_\alpha}(\mathbf{x}, \mathbf{y})$ be the positive diagram of J_α where $\mathbf{x}$ are as many variables as there are elements of $adom(I_\alpha)$ and $\mathbf{y}$ as many variables as there are elements of $adom(J_\alpha) \setminus adom(I_\alpha)$. Let ϕ_α be the following LAV dependency: $\forall\mathbf{x}(PosDiag_{I_\alpha}(\mathbf{x}) \rightarrow \exists\mathbf{y}PosDiag_{J_\alpha}(\mathbf{x}, \mathbf{y}))$. Finally, let $\Sigma = \{\phi_\alpha | \alpha \in \mathsf{facts}\}$. We claim that Σ defines $\mathcal{M}$.

First, we prove *soundness*: every $(I, J) \in \mathcal{M}$ satisfies Σ. Suppose $(I, J) \in \mathcal{M}$, and take any $\phi_\alpha \in \Sigma$. Furthermore, suppose that the antecedent of ϕ_α is satisfied in (I, J) under some variable assignment h. In other words, h is a homomorphism from I_α to I. Let q be the certain answer query of the conjunctive query $\exists\mathbf{y}PosDiag_{J_\alpha}(\mathbf{x}, \mathbf{y})$. Since $\mathcal{M}$ allows for conjunctive query rewriting, q is definable by a union of conjunctive queries. Moreover, since $\exists\mathbf{y}PosDiag_{J_\alpha}(\mathbf{x}, \mathbf{y})$ is satisfied in J_α, which is a universal solution of I_α, it is satisfied in all solutions of I_α. In other words, q is satisfied in I_α, and hence, in I under the assignment h. Hence, (I, J) satisfies ϕ_α.

Next, we prove *completeness*: every pair (I, J) satisfying Σ belongs to $\mathcal{M}$. Suppose (I, J) satisfies Σ. Write I as $I = I_1 \cup \cdots \cup I_n$ where each I_i contains only a single fact. Then each (I_i, J) still satisfies Σ. Since I_i contains a single fact, it must be isomorphic to I_α for some $\alpha \in \mathsf{facts}$. Using the fact that (I_i, J) satisfies ϕ_α, we can show that there is a homomorphism from a universal solution of I_i to J, constant on $adom(I_i)$, hence, by closure under target homomorphisms, $(I_i, J) \in \mathcal{M}$. It follows by closure under union that $(I, J) \in \mathcal{M}$. ❑

Our next result characterizes schema mappings specified by GAV dependencies.

Theorem 4.2. *For all schema mappings $\mathcal{M}$, the following are equivalent*:

1. *$\mathcal{M}$ is definable by a finite set of GAV dependencies.*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and is closed under target intersection.*

Proof. (Hint) The implication (1) ⇒ (2) follows from Theorem 3.4 and Proposition 3.6. For the implication (2) ⇒ (1), we first show that every schema mapping $\mathcal{M}$ satisfying (2) is n-modular for some $n > 0$. For each target relation R, let $q_R = \mathsf{certain}_{\mathcal{M}}(R\mathbf{y})$, where $\mathbf{y}$ is a sequence of distinct fresh variables of appropriate length. Note that, since $\mathcal{M}$ allows for conjunctive query rewriting, q_R can be written as a union of conjunctive queries. Now, let n be the maximum of the number of variables occurring in each q_R. Using the hypothesis that $\mathcal{M}$ admits universal solutions, is closed under target homomorphisms, and is closed under target intersection, it can be shown that $\mathcal{M}$ is n-modular.

After this, the implication (2) ⇒ (1) is established along the same lines as the proof of Theorem 4.1. Instead of considering all source instances consisting of one tuple, we consider all source instances I with $|adom(I)| \le n$. There are only finitely many such source instances up to isomorphism. Moreover, it can be shown, using closure under intersection, that each has a null-free universal solution, and hence only *full* s-t tgds are needed to describe them. ❑

We now focus on schema mappings specified by arbitrary s-t tgds. As seen in Theorem 3.4, every schema mapping defined by a *finite* set of s-t tgds is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting. The next result asserts that any schema mapping satisfying these conditions is definable by an *infinite* set of s-t tgds.

Proposition 4.3. *If a schema mapping $\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting, then $\mathcal{M}$ is definable by an infinite set of s-t tgds.*

Proof. (Hint) Assume that $\mathcal{M}$ satisfies the listed properties. Consider a source instance I and a target instance J such that J is a universal solution for I with respect to $\mathcal{M}$. For each element of $adom(I)$, introduce a distinct variable x_i, and for each element of $adom(J) \setminus adom(I)$, introduce a distinct variable y_j. Define $PosDiag_I(\mathbf{x})$ to be the conjunction of all atomic formulas in $\mathbf{x}$ true in I (under the chosen assignment) and define $PosDiag_J(\mathbf{x}, \mathbf{y})$ likewise. Finally, let Σ be the set of all s-t tgds $\phi_{I,J}$ of the form $\forall\mathbf{x}(PosDiag_I(\mathbf{x}) \rightarrow \exists\mathbf{y}\ PosDiag_J(\mathbf{x},\mathbf{y}))$, where I is a source instance and J is a universal solution for I w.r.t. $\mathcal{M}$. Using an argument analogous to the one used in the proof of Theorem 4.1, it can be shown that Σ defines $\mathcal{M}$. ❑

Can Proposition 4.3 be strengthened to a characterization of schema mappings specified by a *finite* set of s-t tgds? The next result, which was proved in Fagin et al.,[9] shows that this is not possible.

PROPOSITION 4.4. *The schema mapping defined by the first-order sentence* $\forall x \exists y \forall z(Rxz \rightarrow Syz)$ *is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting, but is not definable by any finite set of s-t tgds.*

Can Proposition 4.3 be turned to a characterization of schema mappings specified by an *infinite* set of s-t tgds? The next observation shows that this is not possible.

PROPOSITION 4.5. *The schema mapping defined by the following infinite set of s-t tgds does not admit universal solutions:*

$$\{\forall x(Px \rightarrow \exists y_1 \cdots y_n(Rxy_1 \wedge Ry_1y_2 \wedge \cdots \wedge Ry_{n-1}y_n)) \mid n \geq 0\}$$

PROOF. (Hint) It is easy to see that no solution for $I = \{Pa\}$ can be universal. Here, the assumption that all instances (hence all solutions) are finite is of the essence. ❑

Proposition 4.4 implies that additional properties must be considered in order to characterize the schema mappings that are definable by a finite set of s-t tgds. It turns out that the addition of n-modularity, for some $n > 0$, yields such a characterization.

THEOREM 4.6. *For all schema mappings* $\mathcal{M}$, *the following are equivalent:*

1. $\mathcal{M}$ *is definable by a finite set of s-t tgds.*
2. $\mathcal{M}$ *is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and is n-modular for some* $n > 0$.

We conclude this section by commenting briefly on additional characterizations presented in the previous version of this paper.[17] Observe that the condition of *allowing for conjunctive query rewriting* differs in nature from the other structural conditions: while the latter are model-theoretic conditions, the former refers to a certain syntactically defined class of queries. Thus, it is natural to ask: can the condition of allowing for conjunctive query rewriting be replaced by a condition of model-theoretic character? To this effect, we consider the notion of *reflecting source homomorphisms*, which states, roughly, that every homomorphism between source instances I, I' extends to a homomorphism from any universal solution of I to any universal solution of I'. We show that the structural characterizations of LAV dependencies in Theorem 4.1 and of s-t tgds in Theorem 4.6 hold with the condition of allowing for conjunctive query rewriting replaced by that of reflecting source homomorphisms. Further, we pursue characterizations where one assumes that the schema mapping is first-order definable to start with. We establish that, for all schema mappings $\mathcal{M}$ definable by a first order sentence, $\mathcal{M}$ is definable by a finite set of GAV dependencies if and only if $\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, reflects source homomorphisms, and is closed under target intersection. The proof makes essential use of the sophisticated machinery developed by Rossman[16] for proving the preservation-under-homomorphisms theorem in the finite.

5. COMPLEXITY OF DEFINABILITY

Our characterizations provide tools for testing whether a schema mapping defined in one language can also be defined in another language. For example, our results imply that a schema mapping defined by a finite set of s-t tgds is definable by a finite set of GAV dependencies if and only if it is closed under target intersection: furthermore, it is definable by a finite set of LAV dependencies if and only if it is closed under union. Here, we pinpoint the *computational complexity* of testing definability in the different languages.

First, assume that the input to the problem is a finite set of s-t tgds. The results are summarized in the following table.

Input schema mapping	Desired schema mapping	Complexity of definability
s-t tgds	GAV dependencies	NP-complete
s-t tgds	LAV dependencies	NP-complete
GAV dependencies	LAV dependencies	PTIME
LAV dependencies	GAV dependencies	NP-complete

The proofs also yield effective methods for constructing an equivalent schema mapping in the smaller language whenever it exists. Our proofs are based on reductions from definability problems to the *entailment problem for s-t tgds*; given two schema mappings $\mathcal{M}_1$, $\mathcal{M}_2$, specified by a finite set of s-t tgds, is it the case that whenever $(I, J) \in \mathcal{M}_1$ also $(I, J) \in \mathcal{M}_2$? We show that the latter problem in NP-complete; moreover, it is in PTIME if $\mathcal{M}_1$ is specified by LAV dependencies and $\mathcal{M}_2$ by GAV dependencies.

It is also natural to consider the problem of testing whether a schema mapping specified by a first-order sentence is definable in one of the schema-mapping languages studied here. This problem, however, turns out to be undecidable no matter what schema-mapping language we consider (s-t tgds, GAV dependencies, or LAV dependencies). This can be easily proved using the undecidability of satisfiability for first-order sentences in the finite.[18]

6. DISCUSSION AND OPEN PROBLEMS

The work presented here has methodological implications for the study of schema mappings. Concretely, our structural characterizations delineate the exact set of tools available in the study of schema mappings specified in particular languages. For example, consider the language of LAV dependencies. A perusal of the literature reveals that earlier results about schema mappings specified by a finite set of LAV dependencies made systematic use of the fact that such schema mappings are closed under target homomorphisms, admit universal solutions, allow for conjunctive query rewriting, and are closed under union. The structural characterization given in Theorem 4.1, in effect, turns the tables around and asserts that these four properties are the *only* properties one needs to use in reasoning about schema mappings specified by LAV dependencies. On the computational side, the complexity-theoretic results in Section 5 quantify in precise terms the difficulty of determining whether a schema mapping specified in one language can also be specified in a different language.

There has also been an extensive study of schema mappings specified using languages richer than the language of s-t tgds. Consider, for instance, schema mappings specified by s-t tgds and target tgds, which were studied in Fagin et al.,[7] or schema mappings specified by second-order tgds (SO tgds), which arise when composing schema mappings specified by s-t tgds.[9] These languages are known to be strictly more expressive than the language of s-t tgds. Our results then predict that these languages lack at least one of the structural properties considered here. Indeed, schema mappings specified by s-t tgds and target tgds are closed under target homomorphisms and admit universal solutions, but need *not* allow for conjunctive query rewriting. Likewise, schema mappings specified by SO tgds admit universal solutions and allow for conjunctive query rewriting, but need *not* be closed under target homomorphisms. It remains an open problem to establish structural characterizations for such richer languages of dependencies. A particularly interesting question is whether there is a natural way to characterize *weakly acyclic* sets of target tgds,[7] a class of target dependencies that is of central importance in data exchange, as they guarantee termination of the chase procedure within a polynomial number of steps.

Among the properties of schema mappings considered here, closure under target homomorphisms, admitting universal solutions, and allowing for conjunctive query rewriting are arguably the most fundamental ones for data exchange and data integration. Proposition 4.4 tells that there are schema mappings satisfying these properties that cannot be defined by any finite set of s-t tgds. Is there a natural extension of the language of s-t tgds that is characterized by these three properties? In order to express transformations involving grouping and data merging, a proper extension of the language of s-t tgds, called *nested s-t tgds*, was proposed in Fuxman et al.[11] In the previous version of this paper,[17] we showed that schema mappings specified by a finite set of nested s-t tgds are closed under target homomorphisms, admit universal solutions, allow for conjunctive query rewriting (but may not be *n*-modular for any $n > 0$). It is an open problem to determine whether all schema mappings satisfying these properties can be defined by nested s-t tgds.

Acknowledgments

Research on this paper was partially supported by NSF grants IIS-0430994 and ARRA 0905276, the Netherlands Organization for Scientific Research (NWO) grant 639.021.508, and the ERC Advanced Grant Webdam on Foundations of Web data management. Part of the work was carried out during a visit by Balder ten Cate at UC Santa Cruz and the IBM Almaden Research Center.

References

1. Abiteboul, S., Duschka, O.M. Complexity of answering queries using materialized views. In *ACM Symposium on Principles of Database Systems (PODS)* (1998) 254–263.
2. Avigad, J., Dean, E., Mumma, J. A formal system for Euclid's Elements. *Rev. Symb. Logic* (2009). To appear.
3. Bernstein, P.A., Haas, L.M. Information integration in the enterprise. *Commun. ACM 51*, 9 (2008), 72–79.
4. Codd, E.F. A relational model for large shared data banks. *Commun. ACM 13* (1970), 377–387.
5. Codd, E.F. Relational completeness of data base sublanguages. *Database Systems*. R. Rustin, ed. Prentice-Hall, 1972, 33–64.
6. Duschka, O.M., Genesereth, M.R. Answering recursive queries using views. *ACM Symposium on Principles of Database Systems (PODS)* (1997), 109–116.
7. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L. Data exchange: semantics and query answering. *Theor. Comp. Sci. 336*, 1 (2005), 89–124.
8. Fagin, R., Kolaitis, P.G., Popa, L. Data exchange: getting to the core. *ACM Trans. Database Syst. 30*, 1 (2005), 174–210.
9. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.-C. Composing schema mappings: second-order dependencies to the rescue. *ACM Trans. Database Syst. 30*, 4 (2005), 994–1055.
10. Fagin R., Vardi, M.Y. The theory of data dependencies—a survey. *Mathematics of Information Processing*, volume 34 of *Proceedings of Symposia in Applied Mathematics*, American Mathematical Society, 1986, 19–71.
11. Fuxman, A., Hernandez, M.A., Ho, H., Miller, R.J., Papotti, P., Popa, L. Nested mappings: schema mapping reloaded. *International Conference on Very Large Data Bases (VLDB)* (2006), 67–78.
12. Haas, L.M. Beauty and the beast: the theory and practice of information integration. *International Conference on Database Theory (ICDT)* (2007), 28–43.
13. Haas, L.M., Hernández, M.A., Ho, H., Popa, L., Roth, M. Clio grows up: from research prototype to industrial t. *ACM International Conference on Management of Data (SIGMOD)* (2005), 805–810.
14. Kolaitis, P.G. Schema mappings, data exchange, and metadata management. *ACM Symposium on Principles of Database Systems (PODS)* (2005), 61–75.
15. Lenzerini, M. Data integration: a theoretical perspective. *ACM Symposium on Principles of Database Systems (PODS)* (2002), 233–246.
16. Rossman, B. Existential positive types and preservation under homomorphisms. *Symposium on Logic in Computer Science (LICS)* (2005), 467–476.
17. ten Cate, B., Kolaitis, P.G. Structural characterizations of schema-mapping languages. *International Conference on Database Theory (ICDT)* (2009), 63–72.
18. Trakhtenbrot, B. Impossibility of an algorithm for the decision problem on finite classes. *Dokl. Akad. Nauk. SSSR, 70* (1950), 569–572.
19. van der Meyden, R. Logical approaches to incomplete information: a survey. *Logics for Databases and Information Systems*. Kluwer, 1998, 307–356.

Balder ten Cate (balder.tencate@uva.nl), INRIA and ENS Cachan.

Phokion G. Kolaitis (kolaitis@cs.ucsc.edu), University of California, Santa Cruz and IBM Research–Almaden.

CAREERS

Bucknell University
Assistant Professor
Department of Computer Science

Applications are invited for a tenure track entry-level (up to four years of full-time teaching experience with a recent Ph.D.) assistant professor position in computer science beginning mid-August 2010. Outstanding candidates in all areas will be considered. We are particularly interested in candidates whose research area is in AI, data mining, bioinformatics, or databases. Candidates must have completed their Ph.D. requirements in computer science or a closely related field before the beginning of employment at Bucknell. A strong commitment to excellence in teaching and scholarship is required. The successful candidate must be able to participate in the teaching of required core courses and be able to develop elective courses in the candidate's area of expertise.

Bucknell is a highly selective private university emphasizing quality undergraduate education in engineering and in liberal arts and sciences. The B.S. programs in computer science are ABET accredited. The computing environment is Linux-based. More information about the department can be found at:

http://www.bucknell.edu/ComputerScience/

Applications will be considered as received and recruiting will continue until the position is filled. Candidates are asked to submit a cover letter, CV, graduate transcript, a statement of teaching philosophy and research interests, and the contact information for three references. Please submit your application to

http://jobs.bucknell.edu/

by searching for the "Computer Science Faculty Position".

Please direct any questions to Professor Xiannong Meng of the Computer Science Department at xmeng@bucknell.edu.

Bucknell University values a diverse college community and is committed to excellence through diversity in its faculty, staff and students. An Equal Opportunity/Affirmative Action Employer, Bucknell University especially welcomes applications from women and minority candidates.

Columbia University
Faculty Position in Computer Science

The Department of Computer Science at Columbia University in New York City invites applications for tenured or tenure-track faculty positions. Appointments at all levels, including assistant professor, associate professor and full professor, will be considered. Priority themes for the department include Computer Systems, Software, Artificial Intelligence, Theory and Computational Biology. Candidates who work in specific technical areas including, but not limited to, Computer Graphics, Human-Computer Interaction, and Simulation and Animation, with research programs that can significantly impact the above priority themes are particularly welcome to apply. Candidates doing research at the interface of computer sciences and the life sciences and the physical sciences are also encouraged to apply.

Candidates must have a Ph.D degree, or DES, and are expected to establish a strong research program and excel in teaching both undergraduate and graduate courses. Positions at Assistant Professor rank require demonstrated potential for scholarly success and teaching contributions. Positions at the Associate Professor rank require candidates to have a demonstrated record of scholarly and teaching achievement and have a national reputation in the field of their specialization. At the Professor level, candidates are expected to be scholars and teachers who are widely recognized internationally for their distinction in their chosen field. The Department is especially interested in qualified candidates who can contribute, through their research, teaching, and/or service, to the diversity and excellence of the academic community.

Our department of 34 tenure-track faculty and 1 lecturer attracts excellent Ph.D. students, virtually all of whom are fully supported by research grants. The department has active ties with major industry partners including Adobe, Autodesk, Disney, Dreamworks, Nvidia, Sony, Weta and also to the nearby research laboratories of AT&T, Google, IBM (T.J. Watson), NEC, Siemens, Telcordia Technologies and Verizon. Columbia University is one of the leading research universities in the United States, and New York City is one of the cultural, financial, and communications capitals of the world. Columbia's tree-lined campus is located in Morningside Heights on the Upper West Side.

Applicants should apply online at:

https://academicjobs.columbia.edu/applicants/Central?quickFind=52306

and should submit electronically the following: curriculum-vitae including a publication list, a statement of research interests and plans, a statement of teaching interests, names with contact information of three references, and up to four pre/reprints. Applicants can consult www.cs.columbia.edu for more information about the department.

The position will close no sooner than February 9, 2010, and will remain open until filled. Columbia University is an Equal Opportunity/Affirmative Action 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.

Florida International University
Director of the School of Computing and Information Sciences

Position number 45871: Florida International University (FIU) invites applications for the position of Director of School of Computing and Information Sciences (SCIS, www.cis.fiu.edu) in the College of Engineering and Computing to begin on July 1, 2010.

Candidates must have a doctorate in Computer Science or a closely related field, a reputation for academic and research excellence necessary for appointment as a tenured professor, demonstrated leadership skills, management or supervisory experience, and a strong interest in working with students and the community. Reporting to the Dean of the College of Engineering and Computing, the Director is responsible for promoting excellence in interdisciplinary research and teaching, mentoring faculty, building research teams, administration of School's budgetary and personnel operations, and collaborating with the community and industry.

With a student enrollment of over 39,000, FIU has achieved Carnegie Research University status (with high research activity) and is one of the 25 largest universities in the nation. SCIS is the 6th largest computer science program in the nation and is a major school in the College of Engineering and Computing at FIU. SCIS has 32 faculty members who support its offerings of BS (ABET accredited), MS, and PhD programs in Computer Science, BS/BA in IT, and MS in Telecommunications and Networking. The School serves approximately 1,150 undergraduates, 170 masters, and 90 doctoral students, and provides service courses to many students in various Engineering, Business, and Arts & Sciences departments. In 2008-09, the SCIS received external funding in excess of $4.2M. The School's IT staff operates and maintains research and instructional computing infrastructure and 25 laboratories, including a 10GB network backbone, virtualization-capable data center, multimedia classrooms, and open graduate and undergraduate labs, and IT hands-on teaching labs.

The School's faculty is involved in research areas including databases (data mining, visualization, knowledge discovery, GIS), software systems (high performance computing and networking, large scale modeling simulation, parallel and distributed systems, and adaptive, autonomic, pervasive, and grid computing), software engineering (architecture, design, verification, testing, component technology, and verification), machine learning, telecommunications and networking, security, health informatics, Bioinformatics, algorithms and programming languages.

One of the School's signature University/Industry partnership projects is the Latin American Grid (LA Grid, pronounced "lah grid," http://www.latinamericangrid.org). Co-founded by IBM, and with 12 academic and industry partners sharing over 1,500 systems in an experimental grid, LA Grid brings together computer scientists, domain experts, and industry experts to produce the next generation of leaders of the IT industry by synergistically combining research, education and workforce development activities. It explores "society critical" areas such as disaster mitigation, healthcare, and bioinformatics.

FIU offers a competitive salary and benefits package, and an excellent work environment.

Application review continues until the position is filled. A letter of application, CV, statements of administrative, teaching, and research philosophies, and a list of three references should be submitted to:

http://www.fiujobs.org

Applications are especially encouraged from members of under-represented minorities, women and persons with disabilities. FIU is an equal opportunity/ equal access/ affirmative action employer and institution.

www.fiu.edu/index.htm

Madrid Institute for Advanced Studies in Software Development Technologies (IMDEA-Software)

Open call for Tenured and Tenure-track Research positions in Computer Science

The Madrid Institute for Advanced Studies in Software Development Technologies (IMDEA-Software) invites applications for tenure-track (Research Assistant Professor) and tenured (Research Associate Professor and Research Professor) faculty positions in the following areas: language and system support for multicore platforms, security, rigorous software engineering (including rigorous approaches to validation and testing), distributed, networked and embedded systems, programming language implementation (including compilers, program transformation, garbage collection, the interplay between computer architecture and programming languages, etc.) and service-oriented architectures.

The primary mission of the recently created IMDEA Software Institute is to perform research of excellence at the highest international level in the area of software development technologies and, in particular, science and technology which will allow the cost-effective development of software products with sophisticated functionality and high quality, i.e., which are safe, reliable, and efficient.

Selection Process

The main selection criteria will be the candidate's demonstrated ability and commitment to research, the match of interests with the institute's mission, and how the candidate complements the institute's established strengths in the general areas of programming languages, program analysis, and verification. All positions require an earned doctoral degree in Computer Science or a closely related area. Candidates for tenure-track positions will have shown exceptional promise in research and will have displayed an ability to work independently as well as collaboratively. Candidates for tenured positions must possess an outstanding research record, have recognized international stature, and demonstrated leadership abilities.

Application materials are available at the URL https://www.imdea.org/internationalcall/Default.aspx?IdInstitute=17 wherein candidates must fill out an application form and attach a copy of their CV, a research statement (at most 2 pages), and provide the names and email adresses of referees. Only electronic applications are considered. For best consideration, complete applications must be received by January 15, 2010 (and applicants should arrange for the reference letters to arrive by that date), although applications will continue to be accepted until the positions are filled.

Salaries

Salaries at IMDEA-Software are internationally competitive and are established on an individual basis within a range that guarantees fair and attractive conditions with adequate and equitable social security provision in accordance with existing national Spanish legislation. This includes access to an excellent public healthcare system.

Work Environment
The working language at IMDEA-Software is English. IMDEA-Software is located in the lively area of Madrid, Spain. It offers an ideal working environment, open and collaborative, where researchers can focus on developing new ideas and projects. A generous startup package is offered. Researchers are also encouraged to participate in national and international research projects.

For more information please visit the web pages of IMDEA Software at
www.software.imdea.org

IMDEA is an Equal Opportunity Employer and strongly encourages applications from a diverse and international community. IMDEA complies with the European Charter for Researchers.

Northeastern University
Associate/Full Professor

Joint position with the Colleges of Computer and Information Science and Bouve Health Sciences. The candidate would play a key role in establishing a new interdisciplinary PhD-level Health Informatics degree program. The faculty in our colleges are currently working on multiple NIH-funded research projects. We are interested in faculty who can expand/complement our work in these areas. PhD in Health or Medical Informatics, CS, IS, or a related health-related discipline, together with proven ability to secure grant funding for health informatics research. Apply URL: http://www.neu.edu

Prince Mohammad Bin Fahd University
ARAMCO Endowed Chair in Technology and Information Management

Visit the complete description at:
http://www.pmu.edu.sa/cit/endowed.pdf
or contact Dr. Nassar Shaikh
(C)+966505820407; (O)+96638499300,
nshaikh@pmu.edu.sa

Vacancy open until filled. Review starts 26th October 2009.

The ideal candidate is a leader in technology and information management. Required is a doctorate degree in a closely related field; full professorship and significant experience in teaching, research and consultations; a record of publication and community outreach; experience as editor of a scholarly journal; international network and ability to attract participation in conferences held in Saudi Arabia.

Princeton University
Postdoctoral Researchers

Department of Computer Science, Princeton University is seeking applications for postdoctoral research positions in theoretical computer science. Postdocs will be affiliated with the Center for Com-

M²CI
CLUSTER OF EXCELLENCE

UNIVERSITÄT
DES
SAARLANDES

Saarland University is seeking to establish several

Junior Research Groups (W1/W2)

within the Cluster of Excellence “Multimodal Computing and Interaction” which was established by the German Research Foundation (DFG) within the framework of the German Excellence Initiative.

The term “multimodal” describes the different types of digital information such as text, speech, images, video, graphics, and high-dimensional data, and the way it is perceived and communicated, particularly through vision, hearing, and human expression. The challenge is now to organize, understand, and search this multimodal information in a robust, efficient and intelligent way, and to create dependable systems that allow natural and intuitive multimodal interaction. We are looking for highly motivated young researchers with a background in the research areas of the cluster, including algorithmic foundations, secure and autonomous networked systems, open science web, information processing in the life sciences, visual computing, large-scale virtual environments, synthetic virtual characters, text and speech processing and multimodal dialog systems. Additional information on the Cluster of Excellence is available on *http://www.mmci.uni-saarland.de*. Group leaders will receive junior faculty status at Saarland University, including the right to supervise Bachelor, Master and PhD students. Positions are limited to five years.

Applicants for W1 positions (phase I of the program) must have completed an outstanding PhD. Upon successful evaluation after two years, W1 group leaders are eligible for promotion to W2. Direct applicants for W2 positions (phase II of the program) must have completed a postdoc stay and must have demonstrated outstanding research potential and the ability to successfully lead their own research group. Junior research groups are equipped with a budget of 80k to 100k Euros per year to cover research personnel and other costs.

Saarland University has leading departments in computer science and computational linguistics, with more than 200 PhD students working on topics related to the cluster (see *http://www.informatik-saarland.de* for additional information). The German Excellence Initiative recently awarded multi-million grants to the Cluster of Excellence “Multimodal Computing and Interaction” as well as to the “Saarbrücken Graduate School of Computer Science”. An important factor to this success were the close ties to the Max Planck Institute for Informatics, the German Research Center for Artificial Intelligence (DFKI), and the Max Planck Institute for Software Systems which are co-located on the same campus.

Candidates should submit their application (curriculum vitae, photograph, list of publications, short research plan, copies of degree certificates, copies of the five most important publications, list of five references) to the coordinator of the cluster, Prof. Hans-Peter Seidel, MPI for Computer Science, Campus E1 4, 66123 Saarbrücken, Germany. Please, also send your application as a single PDF file to *applications@mmci.uni-saarland.de*.

The review of applications will begin on January 15, 2010, and applicants are strongly encouraged to submit applications by that date; however, applications will continue to be accepted until January 31, 2010. Final decisions will be made following a candidate symposium that will be held during March 8 – 12, 2010.

Saarland University is an equal opportunity employer. In accordance with its policy of increasing the proportion of women in this type of employment, the University actively encourages applications from women. For candidates with equal qualification, preference will be given to people with physical disabilities.

putational Intractability (CCI) or the Princeton Center for Theoretical Computer Science. Candidates should have a PhD in Computer Science, or related field, or on track to finish by August 2010. Candidates affiliated with the CCI will have visiting privileges at partner institutions NYU, Rutgers University, and Institute for Advanced Study. Apply by Jan 1, (later applications will be accepted) by sending a CV and research statement. Three letters of recommendation are required. They should send letters directly to the website listed below.

http://jobs.princeton.edu, or for general application information and how to self-identify, see

http://www.princeton.edu/dof/ApplicantsInfo.htm

Princeton University is an equal opportunity employer and complies with applicable EEO and affirmative action regulations.

Purdue University School of ECE

Computer Engineering Faculty Position in Human-Centered Computing

The School of Electrical and Computer Engineering at Purdue University invites applications for a faculty position at any level in human-centered computing, including but not limited to visualization, visual analytics, human computer interaction (HCI), and graphics. The Computer Engineering Area of the school (http://engineering.purdue.edu/ECE/Research/Areas/CompEng) has nineteen faculty members who have active research programs in areas including AI, architecture, compilers, computer vision, distributed systems, embedded systems, graphics, haptics, HCI, machine learning, multimedia systems, networking, networking applications, NLP, OS, robotics, software engineering, and visualization. Eligible candidates are required to have a PhD in computer science/engineering or a related field and a significant demonstrated research record commensurate with the level of the position applied for. Applications should consist of a cover letter, a CV, a research statement, names and contact information for at least three references, and URLs for three to five online papers. Applications should be submitted to

https://engineering.purdue.edu/Engr/AboutUs/Employment/Applications.

Review of applications will begin on 1 December 2009. Inquiries may be sent to ece-hcc-search@ecn.purdue.edu. Applications will be considered as they are received, but for full consideration should arrive by 1 January 2010. Purdue University is an equal opportunity, equal access, affirmative action employer fully committed to achieving a diverse workforce.

Rutgers, The State University of New Jersey

Department of Management Science and Information Systems

The Department of Management Science and Information Systems (MSIS) has a tenure-track opening starting Fall 2010 at either the Assistant or Associate Professor level. Candidates should have expertise in information technology. A candidate must be an active researcher and have a strong record of scholarly excellence. Special consideration will be given to candidates with knowledge in data mining, security, data management and other analytical methods related to business operations. Teaching and curriculum development at the undergraduate, MBA, and Ph.D. levels will be expected.

Rutgers University is an affirmative action equal opportunity employer. Applications received by January 18, 2010 are guaranteed full consideration. All applicants should have completed a Ph.D. degree in a relevant subject area by the Fall-2010 Semester. Applicants should send curriculum vitae, cover letter, and the names of three references to:

Ms. Carol Gibson
(CGibson@rci.rutgers.edu, pdf files only)
Department of Management Science and Information Systems
Rutgers Business School
Rutgers, The State University of New Jersey
94 Rockafeller Road
Piscataway, NJ 08854-8054

Southern Methodist University

Department of Computer Science and Engineering

Faculty Position in Computer Engineering, Position number 50679

The Department of Computer Science and Engineering in the Lyle School of Engineering at Southern Methodist University invites applications for a faculty position **at Assistant Professor level** in computer engineering beginning Fall 2010. Individuals with experience and research interests in all areas of computer engineering are encouraged to apply. Priority will be given to individuals with expertise in computer architecture, embedded systems, and related areas. Candidates at all ranks will be considered. The successful candidates must have or expect to have a Ph.D. in Computer Engineering or a closely related area by date of hire. Successful applicants will demonstrate a deep commitment to research activity in computer engineering and a strong potential for excellence in teaching.

The Dallas/Fort Worth area, one of the top three high-tech industrial centers in the country, has the largest concentration of telecommunications corporations in the US, providing abundant opportunities for industrial research cooperation and consulting. Dallas/Fort Worth is a multifaceted business and high-tech community, offering exceptional museums, diverse cultural attractions, and a vibrant economy.

The CSE Department resides within the Lyle School of Engineering and offers BS, MS, and Ph.D. degrees in Computer Engineering and Computer Science, as well as the MS in Security Engineering and Software Engineering. The department currently has 15 faculty members with research concentrations in security engineering, VLSI and digital systems, computer arithmetic, bioinformatics, software engineering, data mining and database systems, network and telecommunication software systems, and related areas. Additional information may be found at: www.lyle.smu.edu/cse.

Interested individuals should send a complete resume and names of three references, including a one-page statement of research interests and accomplishments to:

csesearch@lyle.smu.edu

or

CSE Faculty Search
Department of Computer Science and Engineering
SMU
Dallas, TX 75275-0122

The committee will begin its review of the applications on or about February 1, 2010. To ensure full consideration, applications must be time and date stamped before February 1, 2010. However, the committee will continue to accept applications until the position is filled. Hiring is contingent upon the satisfactory completion of a background check.

SMU will not discriminate on the basis of race, color, religion, national origin, sex, age, disability, or veteran status. SMU is committed to nondiscrimination on the basis of sexual orientation.

Swarthmore College

Computer Science Department

Visiting Assistant Professor

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.

Texas A&M University

Department of Computer Science and Engineering

Tenure-Track Assistant Professor

Applications are invited for tenure-track positions, starting fall 2010, in **the Department of Computer Science and Engineering of the Dwight Look College of Engineering at Texas A&M University**. As part of a long-term plan to increase the size and improve quality, the department is expanding with an assistant professor position in the area of security, with the goal of having the faculty member take advantage of some unique opportunities made available through the Multi-program Research and Education Facility (MREF) at TAMU. MREF allows for faculty to conduct top secret or confidential research in a secure facility. Hence, preference will be given to faculty candidates who will be able to obtain federal security clearance within the first two years. Top candidates in other areas at all professor levels (Assistant, Associate, and Full) will also be considered. Candidates must have a Ph.D. degree and will be expected to teach, perform research, and supervise graduate students.

Texas A&M University CS faculty applicants should apply online at

https://apply2.cse.tamu.edu/gts/applicant/faculty/.

For questions about the positions, contact: search@cse.tamu.edu . Applications are welcome from dual career couples.

Texas A&M University is an equal opportunity/affirmative action employer and actively seeks candidacy of women and minorities.

Toyota Technological Institute at Chicago

Faculty Positions at All Levels

Toyota Technological Institute at Chicago (TTI-C) is a philanthropically endowed degree-granting institute for computer science located on the University of Chicago campus. The Institute is expected to soon reach a steady-state of 12 traditional faculty (tenure and tenure track), and 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

Tufts University

Computers Science Dept.

Faculty Search 2010 in Graphics and/or Visualization (all ranks)

The Department of Computer Science at Tufts University invites applications for a faculty appointment in Graphics and/or Visualization to begin in September 2010. We are particularly interested in candidates who will link to existing research strengths in the department.

We invite applications at all ranks. Senior applicants should have an internationally renowned research program and a strong track record of outside funding. Junior applicants must hold a Ph.D. in Computer Science or closely related field at time of appointment, and are expected to develop a high-quality funded research program. At all levels, we seek outstanding candidates with a strong commitment to undergraduate and graduate teaching and mentoring.

Tufts is among the smallest universities to have been nationally ranked as a "Research Class 1" University. Located in Boston, it has a dynamically growing Computer Science Department. We are part of the Tufts Engineering school which has recently entered an exciting growth phase with a focus on interdisciplinary research. The Tufts Center for Scientific Visualization provides a unique opportunity for collaboration on graphics and visualization within the school. Tufts faculty have many opportunities for cross school interdisciplinary collaborations and benefit from the rich intellectual life of the Boston area.

We request that applications include the following materials (a) a curriculum vitae, (b) a statement describing current and planned research, (c) a statement of teaching philosophy, (d) names and affiliations of three to five potential references and (e) a sample of scholarly work. All these should be submitted online though . Letters of recommendation will be solicited only with the candidate's explicit approval.

Review of applications will begin January 5, 2010 and will continue until the position is filled.

For more information about the department, the position please visit http://www.cs.tufts.edu.

To apply for the position go to:

https://academicjobsonline.org/ajo/TuftsCS/ComputerScience/256.

Inquiries should be emailed to cssearch@cs.tufts.edu

Tufts University is an Affirmative Action/Equal Opportunity employer. We are committed to increasing the diversity of our faculty. Members of underrepresented groups are strongly encouraged to apply.

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 Cincinnati

Open Faculty Position: Assistant Professor of Computer Science

The University of Cincinnati's College of Engineering and Applied Sciences invites applications for a junior tenure-track faculty position in the Department of Computer Science. The position starts in the Autumn quarter of 2010. Candidates should have strong commitment to advancing research and education in computer science. Candidates in all areas of computer science will be considered, but preference will be given to those in software engineering and databases.

Qualifications for this position include a doctoral degree in computer science or a closely related field, the ability to develop and sustain an externally funded academic research program, commitment to quality teaching at graduate and undergraduate levels, and interpersonal skills to motivate and engage students.

The CS Department offers BS and MS degrees in Computer Science and a Ph.D. in Computer Science and Engineering. The Department has well equipped research and teaching laboratories, a mandatory co-op program for the BSCS, and a strong and motivated student body.

The University, having completed a major building campaign, has one of the finest urban settings in American higher education. The University of Cincinnati is a state supported, comprehensive Research 1 institution with an endowment that is 22nd largest among public institutions in the nation, a research program that is funded at a level of more than $370 million annually, and student enrollment of about 37,000.

Applicants must apply through www.jobsatuc.com (Position # 28UC3197) and include a cover letter, curriculum vitae, and contact information of three references. The position will remain open until filled. Additional information is available at http://www.cs.uc.edu/about-cs/job-openings.

The University of Cincinnati is an affirmative action/equal opportunity employer. Qualified women, minorities, veterans, and individuals with disabilities are encouraged to apply. The University of Cincinnati buildings are a smoke-free environment.

University of Denver

Tenure-Track Open Rank Faculty-Computer Science

The Computer Science Department at the School of Engineering and Computer Science of the University of Denver is entering a five year period of rapid growth and expansion. We are currently seeking qualified candidates for two positions, one at the Assistant Professor level and on Open Rank, Tenure Track or Tenured. The primary focus is in Game Development related areas and in Computer Security. However, truly exceptional candidates in other computer science areas will be considered.

Minimum Qualifications: ABD in Computer Science or closely related field. If the candidate has not completed PhD by hire date, they will be hired with title of "Instructor" until the PhD is complete. Preferred Qualifications: PhD in Computer Science or closely related field.

To apply for this position, please visit our website at www.dujobs.org. The University of Denver is an EEO/AA Employer.

University of Kentucky
Computer Science Department
Assistant Professor Level

The University of Kentucky Computer Science Department invites applications for a tenure-track position beginning August 15, 2010 at the assistant professor level in vision/graphics. Candidates must have a PhD in Computer Science. Specific information about the position and the application process is available at

http://www.cs.uky.edu/employment/
positions.php.

The University of Kentucky Computer Science Doctoral Program recently ranked in the top 20% of such programs (30 out of 157) in a nationwide analysis. The rankings -- produced by Academic Analytics -- are based on the Faculty Scholarly Productivity Index(tm), a measure of actual faculty publication, citation, and funding rates. Among doctoral programs at public universities, UKCS was ranked 16th.

The University of Kentucky is an equal opportunity employer and encourages applications from minorities and women.

University of Massachusetts Lowell
Tenure-Track or Tenured Associate Professor Positions in Computer Science

The Computer Science Department at UMass Lowell invites applications for two faculty positions at the rank of Associate Professor to start in September 2010. These are tenure-track or tenured positions depending on qualifications, where an offer at the rank of Full Professor may also be considered. Applicants must hold a PhD in computer science or a closely related discipline and be committed to developing and sustaining externally funded research programs. We are looking for faculty members who have made substantial contributions to their fields and have strong ongoing research projects funded by major US funding agencies. All mainstream research areas will be considered. Successful applicants will be expected to create new programs and contribute to the collaborative research programs of the existing departmental groups.

UMass Lowell is located 25 miles northwest of Boston in the high-tech corridor of Massachusetts. The Computer Science department has 16 tenured and tenure-track faculty. It offers the BS, MS, and PhD degree programs. The Computer Science faculty received approximately $4M in the last two years in external research funding from the NSF, DOD, DOH, and local companies. For information about faculty research areas and the degree programs please visit http://www.cs.uml.edu.

Initial review of applications will begin on December 20, 2009. Applications received by January 20, 2010 are assured of full consideration. Women and under-represented minorities are strongly encouraged to apply.

Please follow the following directions to apply: Submit a cover letter, a current CV, a research statement, a teaching statement, and selected relevant publications through the University of Massachusetts Lowell's website at http://jobs.uml.edu under "Faculty Positions." Submissions directly to the department will not be accepted.

Other optional documents: Please attach to your application summaries of Teaching Evaluations if available.

Arrange for at least three letters of recommendations to be sent directly by email (PDF format preferred) to refs@cs.uml.edu (Applicant's name should appear in the subject line)

The University of Massachusetts is an Equal Opportunity/Affirmative Action Title IX, H/V, ADA 1990 Employer and Executive Order 11246, 41 CFR60-741 4, 41 CRF60-250 4, 41CRF60-1 40 and 41 CFR60-1,4 are hereby incorporated.

University of Nevada Las Vegas
Data Visualization Faculty Position

The Howard R. Hughes College of Engineering, University of Nevada Las Vegas (UNLV) invites applications from all engineering and computer science disciplines for a full-time tenure-track faculty position in Data Visualization at the Assistant Professor level commencing Fall 2010.

Candidates are required to have a B.S. degree in an engineering field or computer science, and a PhD in engineering, computer science, or closely related field. A strong background in data visualization with applications in a relevant engineering field is required. The College is looking for an individual who can conduct innovative research using various forms of data to communicate both abstract and concrete ideas. This is a strategic hire as part of an NSF-funded research project on climate change. The successful candidate will be part of an interdisciplinary team conducting research on the regional impacts of climate change with an emphasis on climate modeling, water resources and ecosystems. Review of applications will begin immediately. A complete job description with application details may be obtained by visiting http://jobs.unlv.edu or call (702) 895-2894 for recruitment assistance. UNLV is an EEO/AA institution.

University of North Texas
Announcement of Faculty Position in Computer Engineering
Department of Computer Science and Engineering

Applications and nominations are invited for an Assistant Professor position in the Department of Computer Science and Engineering at the **University of North Texas**. Applicant must hold an earned doctoral degree (or must receive the degree prior to the appointment date) in Computer Engineering, Computer Science, or a closely related field. Applicant's record must include an indication of quality research such as high quality publications. The preferred research areas include but are not limited to Digital design, Hardware/software co-design, and CAD. Duties include teach at the graduate and/or undergraduate levels in areas of disciplinary expertise and in other CSE areas, conduct research and supervise graduate students. Review of applications will begin on Jan 4, 2010 and will continue until the search is closed. For additional information and to apply please visit: https://facultyjobs.unt.edu/applicants/Central?quickFind=50584 and submit a letter of application, curriculum vitae along with three letters of recommendation. More information about the department can be found at http://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 Texas at El Paso
Tenure-track Position

Computer Science at the University of Texas at El Paso is seeking applicants for at least one tenure-track position. We welcome candidates at all ranks and in all areas of computer science and engineering, especially large-scale systems. For more information see

www.cs.utep.edu/employment.

University of Toronto
Assistant Professor - Computer Science

The Department of Computer and Mathematical Sciences, University of Toronto Scarborough (UTSC), and the Graduate Department of Computer Science, University of Toronto, invite applications for a tenure-stream appointment at the rank of Assistant Professor, to begin July 1, 2010.

We are interested in candidates with research expertise in Computer Systems, including Operating Systems, Networks, Distributed Systems, Database Systems, Computer Architecture, Programming Languages, and Software Engineering.

Candidates should have, or be about to receive, a Ph.D. in computer science or a related field. They must demonstrate an ability to pursue innovative research at the highest level, and a commitment to undergraduate and graduate teaching. Evidence of excellence in teaching and research is necessary. Salary will be commensurate with qualifications and experience.

The University of Toronto is an international leader in computer science research and education, and the Department of Computer and Mathematical Sciences enjoys strong ties to other units within the University. The successful candidate for this position will be expected to participate actively in the Graduate Department of Computer Science at the University of Toronto, as well as to contribute to the enrichment of computer science academic programs at the University's Scarborough campus.

Application materials, including curriculum vitae, research statement, teaching statement, and three to five letters of recommendation, should be submitted online at www.mathjobs.org, preferably well before our deadline of January 17, 2010.

PLEASE NOTE THAT WE ARE ONLY ACCEPTING APPLICATIONS AT: www.mathjob.org

For more information about the Department of Computer & Mathematical Sciences @ UTSC, please visit our home page www.utsc.utoronto.ca/˜csms

University of Toronto
Lecturer - Computer Science

The Department of Computer and Mathematical Sciences, University of Toronto Scarborough (UTSC), invites applications for a full-time position in Computer Science at the rank of Lecturer, to begin July 1, 2010.

We are especially interested in candidates who will help advance our curriculum in the areas of computer systems, computer architecture, and software engineering.

Appointments at the rank of Lecturer may be renewed annually to a maximum of five years. In the fifth year of service, Lecturers shall be reviewed and a recommendation made with respect to promotion to the rank of Senior Lecturer.

Salary will be commensurate with qualifications and experience.

Responsibilities include lecturing, conducting tutorials, grading, and curriculum development in a variety of undergraduate courses.

Candidates should have a post-graduate degree, preferably a PhD, in Computer Science or a related field, and must demonstrate potential for excellence in teaching at the undergraduate level.

Application materials, including curriculum vitae, a statement of career goals and teaching philosophy, evidence of teaching excellence, and a minimum of three reference letters should be submitted online at: www.mathjobs.org, preferably well before our deadline of March 1, 2010.

PLEASE NOTE THAT WE ARE ONLY ACCEPTING APPLICATIONS AT: www.mathjobs.org

For more information about the Department of Computer & Mathematical Sciences @ UTSC, please visit our home at: www.utsc.utoronto.ca/˜csms

University of Toronto
Mendelzon Visiting Assistant Professor

The Department of Computer and Mathematical Sciences, University of Toronto Scarborough invites applications for a non-tenure-stream, two-year appointment as the Mendelzon Visiting Assistant Professor, to begin July 1, 2010.

We will consider applicants in all areas of computer science, but are especially interested in applicants who will help advance our curriculum in computer systems and software engineering.

The University of Toronto is an international leader in computer science research and education, and the Department of Computer and Mathematical Sciences enjoys strong ties to other units within the University.

The successful candidate for this position will be encouraged to engage in collaborative research with other computer science faculty at the university, as well as to contribute to the enrichment of computer science academic programs at the University's Scarborough campus.

Candidates should have, or be about to receive, a Ph.D. in computer science or a related field. They must demonstrate an ability to pursue innovative research, and a commitment to undergraduate teaching.

Application materials, including curriculum vitae, research statement, teaching statement, and three to five letters of recommendation, should be submitted online at www.mathjobs.org, preferably well before our deadline of January 17, 2010.

The University of Toronto is strongly committed to diversity within its community and especially welcomes applications from visible minority group members, women, Aboriginal persons, persons with disabilities, members of sexual minority groups, and others who may contribute to the further diversification of ideas. All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

The Mendelzon Visiting Assistant Professorship is a position created in memory of Alberto Mendelzon, FRSC, distinguished computer scientist, and former chair of the Department of Computer and Mathematical Sciences, University of Toronto Scarborough.

PLEASE NOTE THAT WE ARE ONLY ACCEPTING APPLICATIONS AT: www.mathjobs.org

For more information about the Department of Computer & Mathematical Sciences @ UTSC, please visit our home page: www.utsc.utoronto.ca/˜csms

The University of Washington Tacoma
Director of the Institute of Technology (Associate or Full Professor)

The University of Washington Tacoma is accepting applications and nominations for a full-time, 12 month position as Director of the Institute of Technology, beginning July 1, 2010.

An earned Ph.D. in Computing, Information, Engineering or related fields is required and eligibility for appointment at the rank of Associate or Full Professor.

Submit materials electronically to
academic@u.washington.edu.

Screening of applicants will begin January 3rd, and will continue until the position is filled. For the complete position description, please visit our website:

http://www.tacoma.washington.edu/
academic_affairs/job_opportunities.html

The University of Washington is building a culturally diverse faculty and strongly encourages racial and ethnic minorities, women, and persons with disabilities to apply. University of Washington Tacoma faculty engage in teaching, research and service.

University of Wisconsin-Madison
Assistant, Associate, or Full Professor

The Computer Sciences Department at the University of Wisconsin-Madison has an opening for a tenure-track Assistant Professor in any area of Computer Sciences, or a tenured Associate or Full Professor with a specialization in programming languages/software engineering/verification. In addition to programming languages/software engineering we are especially interested in candidates in HCI, theory, natural language processing, and robotics, but our search is not limited to these areas.

Applicants must have a Ph.D. in Computer Sciences or in a closely related field prior to the start of the appointment. Candidates for a tenured appointment must have a record of distinguished teaching and scholarly research. Candidates for a tenure-track appointment must show potential for developing outstanding and highly visible scholarly research as well as excellence in undergraduate and graduate teaching.

Applicants should submit a curriculum vita, a statement of research objectives and sample publications, and arrange to have at least three letters of reference sent directly to the department. Electronic submission of all application materials is preferred (see http://www.cs.wisc.edu/recruiting for details).

Applicants are encouraged to submit their applications along with supporting material as soon as possible, but no later than January 15, 2010 to ensure full consideration.

The University is an Equal Opportunity/Affirmative Action employer and encourages women and minorities to apply. Unless confidentiality is requested in writing, information regarding the applicants must be released upon request. Finalists cannot be guaranteed confidentiality. A criminal background check may be conducted prior to hiring.

For further information, send mail to
recruiting@cs.wisc.edu.

ACM's Online Books & Courses Programs!

Helping Members Meet Today's Career Challenges

NEW! Over 2,500 Online Courses in Multiple Languages Plus 1,000 Virtual Labs from Element K!

element k®
your learning solutions partner

ACM's new Online Course Collection includes over **2,500 online courses in multiple languages, 1,000 virtual labs, e-reference tools, and offline capability**. Program highlights:

The ACM E-Learning Catalog - round-the-clock access to 2,500+ online courses on a wide range of computing and business topics, in multiple languages.

Exclusive vLab® Virtual Labs - 1,000 unique vLab® exercises place users on systems using real hardware and software allowing them to gain important job-related experience.

Reference Tools - an e-Reference Library extends technical knowledge outside of the classroom, plus online Executive Summaries and quick reference cards to answer on-the-job questions instantly.

Offline Player - members can access assessments and self-study courses offline, anywhere and anytime, without a live Internet connection.

A downloadable Quick Reference Guide and a 15-minute site orientation course for new users are also available to help members get started.

The ACM Online Course Program is open to ACM Professional and Student Members.

600 Online Books from Safari

ACM members are eligible for a **special 40% savings** offer to upgrade to a Premium or Full Library subscription.
For more details visit:
http://pd.acm.org/books/about_sel.cfm

The ACM Online Books Collection includes **full access to 600 online books** from Safari® Books Online, featuring leading publishers including O'Reilly. Safari puts a complete IT and business e-reference library right on your desktop. Available to ACM Professional Members, Safari will help you zero in on exactly the information you need, right when you need it.

Safari®
Books Online

500 Online Books from Books24x7

All Professional and Student Members also have **full access to 500 online books** from Books24x7®, in ACM's rotating collection of complete unabridged books on the hottest computing topics. This virtual library puts information at your fingertips. Search, bookmark, or read cover-to-cover. Your bookshelf allows for quick retrieval and bookmarks let you easily return to specific places in a book.

books24x7®
a SkillSoft Company

acm
Association for
Computing Machinery

Advancing Computing as a Science & Profession

pd.acm.org
www.acm.org/join

[CONTINUED FROM P. 120]

Are you pleased with the results?

There's been a really good return on investment. Researchers are not only inventing new stuff, they are jointly writing top quality, peer-reviewed conference and journal articles.

You've worked hard to transfer the results of HP Labs research into actual business products.

We formed a research advisory board that's composed of lab directors, technologists from all across HP, and business unit leaders. Jointly, we meet and select which projects get funded at HP Labs. We also provide a much better match between what we build in HP Labs and what a business unit can actually receive. Every big bet project gets reviewed twice a year. That way, the business unit members are part of selecting the project, they see how it's progressing, and when the project has ended, they are ready to catch the technology, so to speak.

> **"In the ideal world of academic purity, each researcher does whatever they want and you hope for the best. But our approach has been an approach of focus, and that's really paid off."**

How do you inspire your researchers to think big?

Unless I engage with the researchers on a regular basis, the passion will not be there, the intensity will not be there. I say, "What are you trying to build? Can we not do something bigger?" Then I say, "The specific research you are doing, how will it make that dream come true?" There's an unrelenting pressure that I want to be able to feel—that if we don't do it, someone else will.

Before you joined HP, you spent more than 20 years in academia. Did you find the transition difficult?

Many people warned me, "Prith, you will not survive." Fortunately, my experience in academia was not in only one position. I also took two exits into the world of startups, and what I learned from those startups are things I've tried to preach and practice at HP Labs.

Such as?

In academia, professors can work and work and refine their papers to the last word. The world of startups doesn't give you that luxury. I learned how to deliver fast. I learned how to run—I am always running. I try to convey that to my colleagues at HP Labs—you have to find the right trade-off. I want you to document your work. I want you to advance the state of the art, but don't just keep on publishing and refining—run, right? Start making stuff, as well. C

Leah Hoffmann is a Brooklyn-based technology writer.

DOI:10.1145/1629175.1645394

Leah Hoffmann

Q&A
HP's Running Man

Prith Banerjee discusses collaborating with universities, his startup experiences, and Hewlett-Packard Lab's approach to research and development.

IN 2007, AFTER spending 20-plus years in academia, founding two successful companies, and being honored by some of the industry's most prestigious technical societies, Prith Banerjee turned his considerable energies to the world of corporate research. Here, the director of Hewlett-Packard Labs talks about how to balance scientific and business goals, collaborate with university researchers, and inspire people to think big.

PHOTOGRAPH BY DAVID PAUL MORRIS

When you joined HP Labs, you undertook a fairly ambitious reorganization. Can you describe your new approach?

Our approach is focused on several themes. One is to work on what we call high-impact research, or "big bets." In the past, we used to have 150 smaller projects. Each was interesting, but none had the chance to move the needle for HP. Now we focus our energies on about 20 big bet projects. These projects are simultaneously trying to advance the state of the art—to do something that the world doesn't know how to do today—and to have a significant business impact for HP.

You've also organized your research around eight key topics, such as analytics, immersive interaction, information management, and so on.

In the ideal world of academic purity, each researcher does whatever they want and you hope for the best. But our approach has been an approach of focus, and that's really paid off.

Many people have begun to question the role of the corporate research lab in an era of low-cost communication and shrinking profit margins.

HP invests a lot in R&D, but much of the research is done in the various product divisions. Our role is to look beyond that and provide the company with opportunities that will be interesting in three, five, 10, 15 years. One-third of our research is devoted to fairly basic fundamental science. One-third of our research is related to a product. The third area is applied research, which is somewhere in between.

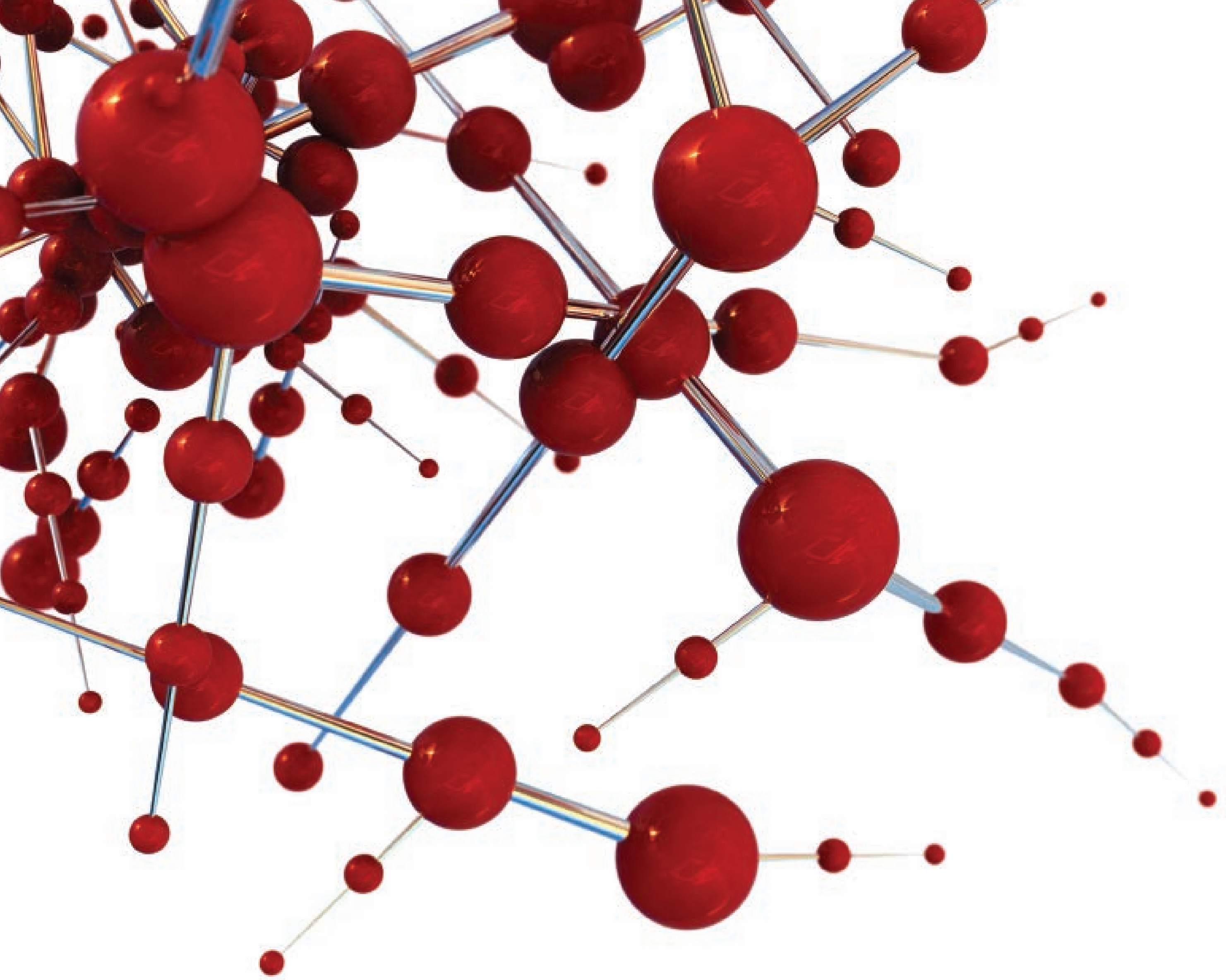
You've also made a big push toward open innovation with your annual Innovation Research Awards.

Working with academia is something many companies do, so that part is not new. The novelty of the HP approach is the specific way we sought alignment with our academic colleagues. Most companies fund research in academia with so-called random acts of charity—they don't tell you what research to do. In a way it's good, because they want you to be completely independent. The trouble is, that research is hardly relevant to the company.

Instead, you solicit ideas for collaborative projects based upon your eight research themes and your 20 big bet projects.

We channel our funding into activities that, after a lot of thought, we have decided to invest in. Then we tell our university colleagues, "These are very hard and important problems, and we need your help." Each grant is somewhere on the order of $75,000 to $100,000 per year for a duration of three years.

[CONTINUED ON P. 119]



CALL FOR PAPERS

http://fdg2010.org

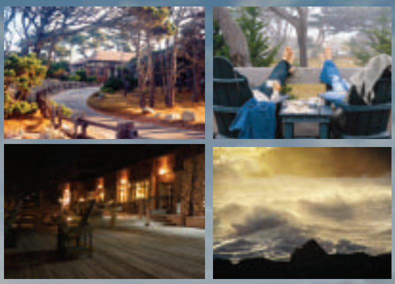
the International Conference on the

Foundations of Digital Games

June 19 - 21 2010

Submission Deadlines

Papers & Posters: 5th Feb
Doctoral Consortium: 12th Feb
Demos: 2nd April



Asilomar Conference Grounds
Monterey, California, USA.

Game Studies

Infrastructure (Databases, Networks, Security)

Learning in Games

Computer Science & Games Education

Game Design

Graphics & Interfaces

Artificial Intelligence

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

In Cooperation With ACM
and its Special Interest Groups on
Computer Science Education
and Artificial Intelligence

SIGART

SIG CSE

acm

FDG 2010 Supported by:
Microsoft Research

© SASDG. Photo credits:
1. (cc) (by=) Veronica Vale - www.flickr.com/people/vbv
2. © Asilomar Conference Grounds - www.visitasilomar.com
3. (cc) (by$=) Andrew Fitzhugh - www.flickr.com/people/fitzhugh
4. (cc) (by$=) Steve Schnwarz - www.flickr.com/people/vasculata