Editorial Pointers

We all recognize how first impressions color potential human relationships. In real life, we tend to be drawn to others who exhibit proper etiquette and suitable conduct similar to our own. An encounter with someone ill-mannered or with bad habits typically sparks a reaction of distrust or displeasure. So, as we invite software systems to play an increasingly intimate and autonomous role in our daily lives, is it any wonder we look for these tools to display the kind of etiquette rules we favor in human relationships?

A growing community of researchers, practitioners, educators, psychologists, and sociologists are exploring the etiquette perspective, believing systems that display proper behavior not only win user acceptance, but enhance the accuracy and speed with which users develop trust and confidence in systems and software. This month's special section examines human-computer etiquette with intentional agents, namely, complex systems built with "social actors" displaying subtle rules of decorum befitting human-like collaborators. Guest editor Christopher Miller, chief scientist at Smart Information Flow Technologies, explains the etiquette perspective is a field in its infancy; it is the goal of this section to bring the early research to the forefront. The authors provide a variety of perspectives on the use and potential of etiquette-based design, including many examples of its application and utility.

Also in this issue, Krishna et al. contend a critical step in managing global software projects is to examine the cross-cultural issues that may hinder progress. Wallace and Keil explore the relationship between the risks related to software projects and their ultimate outcomes. With multimillion-dollar failures dominating track records, their findings offer valuable insight managers must consider before initiating a project.

It may be unrealistic to expect users to apply (and *remember*) a different password for every online account they keep, but as Ives et al. warn, the practice of reusing passwords poses a real domino threat of untold damage on heretofore secure systems. Li discusses the implications of "herding," that is, the practice of managers following the same IT adoption decisions rather than thinking strategically and independently.

In "The Profession of IT," Peter Denning celebrates innovation by tracing the roots of some of the industry's classic innovative notions and illustrating ways innovation can be better understood. And in "Technical Opinion," John Gerdes offers a rare look at the mechanisms that support anonymous employment while addressing government reporting requirements.

Diane Crawford
Editor

COMMUNICATIONS OF THE ACM • *A monthly publication of the ACM Publications Office*

ACM, 1515 Broadway, New York, New York 10036-5701 USA (212) 869-7440 FAX: (212) 869-0481

Advertising
ACM Advertising Department
1515 Broadway, New York, NY 10036-5701
(212) 869-7440; Fax: (212) 869-0481
Account Executive:
William R. Kooney email: acm-advertising@acm.org

For the latest media kit—including rates—contact:
Graciela Jacome: jacome@acm.org

Contact Points
CACM editorial: crawford_d@acm.org
Copyright permission: permissions@acm.org
Calendar items: calendar@acm.org
Change of address: acmcoa@acm.org

Communications of the ACM
(ISSN 0001-0782) is published monthly by the ACM, 1515 Broadway, New York, NY 10036-5701. Periodicals postage paid at New York, NY 10001, and other mailing offices. POSTMASTER: Please send address changes to Communications of the ACM, 1515 Broadway, New York, NY 10036-5701 USA

Printed in the U.S.A.

ACM *The Association for Computing Machinery*

ACM (founded 1947) is an international scientific and educational organization dedicated to advancing The art, science, engineering, and application of information technology, serving both professional and public interests by fostering the open interchange of information and by promoting the highest professional and ethical standards.

Executive Director and CEO:	John White
Associate Director, Office of Director, ACM U.S. Public Policy Office:	Jeff Grove
Deputy Executive Director and COO:	Patricia Ryan
Director, Office of Information Systems:	Wayne Graves
Director, Office of Financial Services:	Russell Harris
Financial Operations Planning:	Darren Ramdin
Director, Office of Membership:	Lillian Israel
Director, Office of Publications:	Mark Mandelbaum
Deputy Director:	Bernard Rous
Deputy Director, Magazine Development:	Diane Crawford
Publisher, ACM Books and Journals:	Jono Hardjowirogo
Director, Office of SIG Services:	Donna Baglio
Assistant Director, Office of SIG Services:	Erica Johnson
Program Director:	Ginger Ignatoff

ACM Council

President	Maria Klawe
Vice-President	David S. Wise
Secretary/Treasurer	Telle Whitney
Past President	Stephen R. Bourne
Chair, SGB Board	Alan Berenbaum
Chair, Publications Board	Robert Allen

Members-at-Large: Roscoe Giles (2002–2006); Denise Gürer (2000–2004); David S. Johnson (1996–2004); Michel Beaudouin-Lafon (2000–2004); Edward Lazowska (2000–2004); Barbara Ryder (2002–2004); Gabriel Silberman (2002–2006)
SGB Council Representatives: Stuart Feldman (2002–2004); Mark Scott Johnson (2001–2005); Jim Cohoon (2001–2005)

Board Chairs and Standing Committees

Education Board: Russell Shackelford; **SGB Board:** Alan Berenbaum; **Membership Board:** Terry Coatta; Mike Macfaden; **Publications Board:** Robert Allen **USACM Committee:** Eugene Spafford, Barbara Simons

SIG Chairs

SIGACT: Harold Gabow; **SIGAda:** Currie Colket; **SIGAPL:** Robert G. Brown; **SIGAPP:** Barrett Bryant; **SIGARCH:** Norman Jouppi; **SIGART:** Maria Gini; **SIGBED:** Janos Sztipanovits; **SIGCAPH:** John Goldthwaite; **SIGCAS:** Tom Jewett; **SIGCHI:** Joseph Konstan; **SIGCOMM:** Jennifer Rexford; **SIGCSE:** Henry Walker; **SIGDA:** Robert Walker; **SIGDOC:** Scott Tilley; **SIGecom** Michael Wellman; **SIGGRAPH:** Alain Chesnais; **SIGGROUP:** Wolfgang Prinz; **SIGIR:** Jaime Callan; **SIGITE:** Edith Lawson; **SIGKDD:** Won Kim; **SIGMETRICS:** Leanna Golubchik; **SIGMICRO:** Kemal Ebcioglu; **SIGMIS:** Janice Sipior; **SIGMOBILE:** Victor Bahl; **SIGMOD:** M. Tamer Ozsu; **SIGMULTIMEDIA:** Ramesh Jain; **SIGOPS:** Keith Marzullo; **SIGPLAN:** Michael Burke; **SIGSAC:** Sushil Jajodia; **SIGSAM:** Emil Volcheck; **SIGSIM:** John Tufarolo; **SIGSOFT:** Alexander Wolf; **SIGUCCS:** Robert Paterson; **SIGWEB:** Peter Nuernberg

For information from Headquarters: (212) 869-7440

ACM U.S. Public Policy Office:
Jeff Grove, Director
1100 Seventeenth St., NW
Suite 507
Washington, DC 20036 USA
+1-202-659-9711—office
+1-202-667-1066—fax
jeff_grove@acm.org

COMMUNICATIONS OF THE ACM • *A monthly publication of the ACM Publications Office*

ACM, 1515 Broadway, New York, New York 10036-5701 USA (212) 869-7440 FAX: (212) 869-0481

Subscriptions

Annual subscription cost is included in the society member dues of $99.00 (for students, cost is included in $40.00 dues); the nonmember annual subscription is $179.00 See top line of mailing label for subscription expiration date coded in four digits: the first two are year, last two, month of expiration. Microfilm and microfiche are available from University Microfilms International, 300 North Zeeb Road, Dept. PR, Ann Arbor, MI 48106; (800) 521-0600.

Single Copies are $8 to members and $17 to nonmembers. Please send orders prepaid plus $7 for shipping and handling to ACM Order Dept., P.O. Box 11414, New York, NY 10286-1414 or call (212) 626-0500. For credit card orders call (800) 342-6626. Order personnel on duty 8:30-4:30 EST. After hours, please leave message and order personnel will return your call.

Notice to Past Authors of ACM-Published Articles ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you were previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG newsletter at any time, and you do not want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

News Track

Political Agendas

The offshoring of IT jobs has become such a political tinderbox in the U.S. that India's outsourcing specialists fear its repercussions. The chairman of a leading Indian software firm recently told the *New York Times*: "The dramatic buildup of opposition before the U.S. elections is disturbing," pointing out political reaction against outsourcing has already culminated in almost two dozen U.S. states voting to ban government work from being contracted to non-Americans. Moreover, the U.S. Senate approved a bill aimed at restricting outsourcing contracts from two federal departments. Technology researchers Gartner Group predict the outsourcing reaction will continue to escalate at least through the fall. Indian officials hope the effect of the U.S. legislations will be minimal and that common economic interests will override political differences that could lead to trade protectionism. At stake: India's IT technology and services industry earned $12 billion in revenue in FY03, $9.5 billion of which came from exports. FY04 is projecting $15.5 billion in revenues with exports of $12 billion.

Spam Rage

A growing number of Internet users are doing more than hitting the delete button when it comes to eliminating spam from their lives; they are in fact squaring off against spammers in more extreme, sometimes violent, ways. *USA Today* reports that spam rage, like road and air rage, is an escalating trend, where many spam activists spend hours tracking spammers and reporting them to authorities. Others engage in cyberwarfare by shutting down spammers' Web pages or putting spammers' addresses on Web sites. Others sue. And some are even resorting to violent threats. A usually mild-mannered 44-year-old Silicon Valley programmer was recently arrested by the FBI and charged with eight violations of interstate communications for threatening to torture and shoot an employee of a company that spammed him relentlessly. Experts say there are far more cases of spam rage than ever reported because spammers do not want to invite the law to scrutinize their operations.

Digital Doubts

"I thought digital was better, but apparently it's not," contends a consultant for the Philadelphia public defender's office. The "digital" in question is photography, once favored by police departments for its cost and enhancement benefits, now more often confounding the justice system. The Associated Press reports that digital courtroom evidence is now frequently challenged, especially when the word "enhancement" is whispered. Because digital images are mere bits of data, manipulating them is much easier than any potential darkroom tricks with film. "What you can do in a darkroom is 2% of what Photoshop is capable of doing," says a former head of photography for State Farm Insurance. Forensic specialists worry more about unwitting errors introduced by poorly trained examiners than by intentional

ALTERED EGOS: IDENTITY THEFT IS NUMBER-ONE CONSUMER COMPLAINT

Consumers filed more than 516,000 complaints with the Federal Trade Commission (FTC) last year, up from 404,000 in 2002. For a fourth straight year, identity theft was the most common gripe, the agency reported recently, although claims involving Internet-related fraud—from bogus auctions to get-rich-quick schemes—accounted for about 55% of the total. Some of the top categories of consumer-fraud complaints in 2003, by percentage:

Identity theft	42%
Internet auctions	15%
Shop-at-home/catalog sales	9%
Internet services/computers	6%
Prizes, sweepstakes, and lotteries	5%
Foreign money offers	4%
Telephone services	3%
Other	16%

Source: The Associated Press

ILLUSTRATIONS BY MASAKO EBATA

changes, which are quite rare. Concerns about the impeachability of digital photos are one reason many police departments have been hesitant to ditch film for crime scene photographs and forensic analysis. However, others point out newer software such as More Hits and recent versions of Photoshop can automatically log changes made to an image so the alterations can be reproduced by others.

Women Got Game

A new survey from America Online claims that when it comes to burning the midnight oil playing online games, older women take the lead. Findings of the AOL study indicate U.S. women over the age of 40 spend nearly 50% more time each week playing online games than men and are more likely to play online games daily than men or teens. More than a quarter of those women play their favorite games between midnight and 5 a.m.; the majority tends to favor word and puzzle games. AOL also researched the gaming habits of major U.S. cities and found that people who play games online in Los Angeles are more likely (31%) to form offline relationships than the national average (18%). Atlanta and Boston were the most game-happy cities overall, at about eight hours per capita per week.

Perfect Fit

Those who loathe the shopping experience, especially trying on clothes, will be relieved to know that Toshiba has teamed up with Osaka-based Digital Fashion to create a 3D virtual you to dress. BBC News Online reports the process of turning a shopper into a photo-realistic avatar occurs in real time; video cameras snap the shopper and then clothes and accessories are selected and displayed immediately on screen. The avatar replicates the shopper's exact measurements, giving him or her a true sense of how the clothes will look as they walk and move. Toshiba expects the system will revolutionize online shopping by eliminating the current mix of static mannequins. The system could be in use by 2006.

Awkshun Serchs

Some lonely eBay sellers, long puzzled by the lack of online enthusiasm for their goods, are now the favored folks for a growing cult of bargain hunters who search the auction site for those who simply, well, can't spell. Yes, a staunch group of eBay hunters are finding some serious bargains by ferreting out such products as labtop computers, Art Deko vases, camras, saffires, comferters, antiks, and dimonds (spellings all found in a recent eBay search). The *New York Times* reports a growing number of eBayistas search specifically for misspellings, knowing there is likely a frustrated seller on the other end who will accept a lowball bid just to get rid of the item(s). Often these buyers will then turn around and sell the item all over again on eBay for a much higher price simply because they spelled the item's name correctly. Educators say it's not so much a matter of more bad spellers in the population; it's that the online world has done a great deal toward publicly exposing them.

Send items of interest to cacm@acm.org

Image Makeover Do-Over

We've heard from several readers regarding an item in last month's Newstrack column (Mar. 2004, p.9). The "Image Makeover" entry points to results from a three-year UCLA study of Net and non-Net users in 14 countries. Included in the remarks were findings of gender gaps in Net usage among the surveyed countries. The correct statement should have read: The gender gap is most prevalent in Italy where 41.7% of the men and 20.1% of the women are Internet users. The lowest gap was in Taiwan where 25% of the men and 23.5% of the women use the Net.

Open Source vs. Capitalism and Communism

Robert L. Glass related open source/free software and communism in his "Practical Programmer" column ("A Look at the Economics of Open Source," Feb. 2004), emphasizing the failure of the latter in order to question the soundness of the former. However, open source/free software development and communist economies differ in many ways:

- Nobody is forced to write "free software." In a communist economy, the collective or, in many regimes, the bureaucracy assigns tasks to workers.
- Many free software contributors have a vested interest in what they write. Programmers may, for example, write a device driver for a device they bought but are unable to use under Linux. Having written the driver, the programmers are likely better off making it available to the general public, allowing others to improve it while discharging the original authors from its maintenance—clearly a win/win outcome.
- Open source software is often developed for a profit. For instance, hardware manufacturers might finance the development of free software drivers in order to improve their sales in competitive markets.
- Software is largely geared to particular industries in that it generally involves low equipment costs, and the marginal cost of an additional copy can be driven close to zero. The collaborative writing of software—where a product may be used by thousands of users—thus makes more sense than the collaborative building of industrial products where each copy requires expensive materials and a dedicated work force.

Glass also assigned too much importance to the claims of "columnists" and "open-source advocates." Do their opinions represent the majority of open source contributors—or are they just the most outspoken?

David Monniaux
Paris, France

Robert L. Glass seems to think "success" for open source software means other forms of software development will be driven out of business. There is, in fact, room for many models and styles of developing software, and the bazaar style of open source software development is but one of them.

An article on the economics of open source should have described when open source has and when it lacks competitive advantage. A key feature of the bazaar style is that programmers are also users, and requirements analysis and requirements documents are not needed. This works when the software is used primarily by programmers but not when used by other specialists.

Since developers are distributed, design communication is difficult. The bazaar style works better with systems reflecting a well-known design (such as anything taught in a CS course). It works less well with innovative designs. Innovative designs work

PAUL WATSON

better with a cathedral-style process or with a process that enables regular communication among developers. Fortunately for the bazaar style, most software does not involve innovative design. Unfortunately, most software is not designed to be used by programmers.

Beyond the bazaar style, some open source software, including qmail and ghostscript, were developed by individual genius programmers. Software development by geniuses is a well-known and successful way of developing software, limited mainly by the supply of geniuses. The bazaar style is interesting because it is so different from traditional software development and can be used by regular programmers.

Open source software is already a success. It will not eliminate other forms of software, though in a few cases it will dominate. Programmers who avoid all open source software are making a mistake. Few programmers will ever spend much time developing open source software, but most will take advantage of it. All software developers should understand the strengths and weaknesses of open source, as well as when to use it as a development strategy.

Ralph Johnson
Urbana, IL

Robert L. Glass's column did little more than provide evidence that Glass does not understand the core principle upon which the open source movement is based. The premise of the movement is that authors do not necessarily program for free; they do, however, expect to be paid for the value their effort adds to the software. As with any profession, they decide when to work pro bono.

Clark E. Hartsock, III
Emeryville, CA

From the point of view of someone living in the democratic and capitalist society of North America, it's impossible not to respect those who strongly support the open source movement. But because of our capitalist economy it's equally difficult not to see the possibility of it producing future economic problems, as discussed by Robert L. Glass.

In a capitalist society, deliberately providing intellectual effort for free seems a poor economic decision. Indeed, as our society has moved up the economic food chain from natural resources to manufacturing to products increasingly based on intellectual content the open source movement could ultimately affect our GDP much like piracy affects it.

Open source developers from democratic and capitalistic societies are "funded" by those societies' increasing GDP before they even write their code. Ironically, open source developers might negatively affect potential economic incentives to invest in future developers.

Linus Torvalds was initially drawn to his work on Linux because he could not afford a Unix license or the kind of high-end machine he would need to run Unix. I can't help but wonder whether open source would have still had its window of economic opportunity in the first place if the products of private software companies did not involve historically high margins and retail prices. One result of open source might be to force the lowering of prices of proprietary software and the squeezing of margins while also winning greater market share before the next Linus comes along.

I like the open source movement, especially its people and their passion, but worry that as increasingly dependent as North America is on intellectual property, the movement might in the long term erode our GDP and therefore our overall standard of living.

Ken Nickerson
Aurora, Ontario, Canada

Author Responds:

Far too many conversations about open source end in acrimony, with each side charging the other is at best ignorant and at worst evil. I was in general pleased with the objective tone of these Forum comments. Some important points were made: that open source use is successful and growing, but that growth is not destined to supersede proprietary source; that perhaps we tend to listen too much to columnists and gurus on this subject and not enough to ordinary practitioners; and that it is important to know when particular software development methods (including open source) are useful and when they are not.

I would like to add that my column should have said that the most interesting and unique characteristic of open source is that programmers tend to do it for free and not imply it is a necessary characteristic. However, I disagree with the letter writer who said that open sourcers are paid for the "value added" they create. This may be true, but it certainly is not commonly true. And I vigorously disagree with all claims that open source programmers and their programs are necessarily the best. There is no evidence to support such claims, and common sense says they are unlikely to be true.

Robert L. Glass
Bloomington, IN

What Worker Shortage?

In "Programming Languages and Gender" (Jan. 2004), Ronald Dattero and Stuart D. Galup began by stating: "The scarcity of information technology workers has been widely acknowledged."

The idea that there is a shortage of IT workers in the U.S. is false, and I was disappointed that these authors adhered to such a view. A leading U.S. academic expert on the computer technology industry, Norman Matloff of the University of California, Davis, has demonstrated there is no shortage of U.S. workers to fill these jobs, and several studies confirm his conclusion.

As far back as 1998, at the height of the worker-shortage concerns, the U.S. General Accounting Office could not substantiate the claims of such a shortage. In the same year, the Information Technology Association of America admitted, after its own study of job vacancies, said that the supposed shortfall had failed "to live up to its prior billing." Meanwhile, the high-tech industry actually laid off workers at four times the rate of other industries.

The truth is that H-1B visa workers and outsourcing have led to lower wages and higher unemployment among high-tech workers.

Not only did a 2001 National Research Council report conclude that H-1B visas have had an adverse effect on wage levels, a UCLA study cited by Matloff showed that H-1B workers are paid 30% less than comparable Americans. A Cornell University study found the difference to be 20% to 30%.

In 2003, the U.S. Bureau of Labor Statistics reported that unemployment among electronics engineers had soared to 7% and among computer hardware engineers to 6.5%. The IEEE stated that these employees had lost 241,000 jobs over the previous two years, while computer scientists and systems analysts lost 175,000 jobs.

Companies can increase profits by spending less on labor. The claims over the past several years that there is a shortage of high-tech workers has been nothing more than an effort to employ cheaper labor, first by bringing foreign workers to the U.S. and, more recently, by increasing outsourcing to foreign nations.

Highly skilled and competent workers are readily available in the U.S. I hope the authors take note of this.

Marc Bellusci
Harrison, NY

Design for Pedagogy Not Just for Software

Two points need to be made about Cong-cong Xing's and Boumi Belkhouche's "On Pseudo Object-Oriented Programming Considered Harmful" (Technical Opinion, Oct. 2003).

The authors overemphasized the distinction between design and coding, saying: "We need to separate the essence of OOP from OOP languages to avoid entangling coding and design issues." Preventing entanglement of coding and design issues is impossible since they are an unbroken continuum; design leads naturally to implementation and implementation back to design, especially in debugging. To regard each as a distinct activity contradicts the axiom that the objective is to produce a working system."

A prime OOP objective is to model real-world objects using software objects. The out object and methods of Java's system classes model the output stream well enough—and is all one needs to display, say, "Hello world." For educational purposes, a number of authors have wrapped these good-enough elements into two additional classes—HelloWorld and UseHello—that add nothing to displaying the string. Adding unnecessary complexity, even for an apparently good purpose, is not a good idea. Designing-in pedagogy is as important as design itself in software.

Alex Simonelis
Montreal

Please address all Forum correspondence to the Editor, *Communications*, 1515 Broadway, New York, NY 10036; email: crawfordd@acm.org.

John White

ACM's Professional Development Centre Expands Learning Opportunities

One of ACM's goals is to continually provide our members with meaningful professional development opportunities, supporting life-long career growth. In accordance with this goal, we are proud to announce the expansion of the ACM Professional Development Centre (PDC).

Ongoing interest from members for additional professional development resources has led to the current PDC expansion.

The PDC was originally created in response to member feedback emphasizing a need for greater professional development opportunities. The site was launched in 2002 with 250 online courses offered in conjunction with Sun Educational Services. Since that time, the PDC has become one of the most highly valued membership benefits of ACM.

Ongoing interest from members for additional professional development resources has led to the current expansion. The PDC now offers over 350 courses hosted at the Sun Learning Center (SLC). The expanded PDC encompasses greater content areas, including specific topics such as C/C++, Oracle 9i, NET, and Flash MX while adding courses in some of the most popular content areas—Java Technology, Project Management, and Networking. There are also new courses in Business Skills, such as Introduction to Finance, Effective Business Writing, and Working on Global Teams.

The PDC remains free, with unlimited access for all ACM members. It is a membership benefit we are constantly working to improve, and to that end we will continue to seek ways of increasing its value. Two features we are exploring for the immediate future include adding courses with Continuing Education Units (CEUs) and adding individual course ratings based on member feedback.

We encourage you to use the PDC and give us your feedback via Course Evaluations (pd.acm.org/info/eval.cfm) and Course Topic Suggestions (pd.acm.org/info/tell_us.cfm). Our goal is to provide the most up-to-date, effective learning resources possible.

John White is the executive director and CEO of ACM.

The Profession of IT

Peter J. Denning

The Social Life of Innovation

Fostering a change of practice in a community is much more difficult than inventing a new technology. The practice of innovation can be learned—once you know what it is.

Innovation is one of the most celebrated aspects of technology. Business leaders consider innovation a core competency, the only way to assure marketplace strength for lean organizations. Popular technology magazines annually venerate top innovators with special articles and profiles. Books on innovation—for example, *The Innovator's Dilemma* [2], *Creative Destruction* [5], and *Value Migration* [9]—are frequent bestsellers. Our computing graduates have been steeped in stories of computing technologies that changed the world—and many dream of one day doing likewise. Considering these circumstances, I cited innovation as one of computing's core practices; a practice without which one cannot be a complete professional (see my November 2003 column).

Many organizational leaders speak of their desire to establish a "culture of innovation." They mean: without special urgings by leadership, everyone in the organization is busy finding ways to help customers (and themselves) improve their practice. A culture of innovation cannot be achieved without cultivating personal practices of innovation throughout the organization. What are these practices? What must one learn to become a skilled innovator? How can teachers help those who seek the skill? This column suggests answers to these questions.

Innovation versus Invention

The word innovation has been used to mean either new ideas or new practices. Since ideas have no impact unless adopted into practice, I use innovation to mean the adoption of a new practice in a community. Innovation is therefore a social transformation in a community.

I draw a sharp distinction between innovation and invention. Invention means simply the creation of something new—an idea, an artifact, a device, a procedure (see [6–8] for recent examples). There is no guarantee that an idea or invention, no matter how clever, will become an innovation. Preserving the distinction is crucial because, as will be discussed shortly, the practice of innovation is not a practice of inventing. Innovation requires attention to other people, what they value and will adopt; invention requires only attention to technology.

Bob Metcalfe, the inventor of Ethernet, speaks of this distinction colorfully. In a 1999 interview, his interlocutor exclaimed, "Wow, it was the invention of the Ethernet that enabled you to buy your house in Boston's Back Bay!" Metcalfe responded: "No, I was able to afford that house because I sold Ethernets for 10 years!"

Although business innovations get the lion's share of attention,

RANDAL ENOS

innovation in fact occurs in all kinds of organizations and communities. And it occurs at all scales from a handful of people adopting new practice, to billions of the human populace.

The term innovation also refers to the discipline and practice of innovation—the work of innovators. Innovation can be carried out systematically and its principles taught. Success is not a matter of psychology, charisma, inspiration, talent, or flash of genius; it is a matter of education.

The remainder of this column is about the practice of innovation. If you are not concerned about selling your ideas or seeing them through to practice, the following will be of limited interest to you.

Innovation as a Discipline

In 1985, Peter Drucker published the definitive work, *Innovation and Entrepreneurship* [4]. He focused on two main aspects: The practice of innovation—the innovator searches for opportunities and transforms them into a new practice in the marketplace; and the practice of entrepreneurship—institutional ways and processes embed the practice of innovation into an organization. He analyzed a large number of cases to reveal the five elements of the process of innovation (see Table 1). Drucker wrote his book after working 30 years with innovators. The dot-com bust 15 years later might have been avoided if the leaders of the new ventures had all read it.

Searching for opportunity	Noticing an opportunity in one of the eight innovation sources.
Analysis	Developing a project or business plan, identifying costs, resources, and people, analyzing risk and benefits.
Listening	Going out into the community, listening for concerns, finding what they are receptive to; adapting the proposal to match.
Focus	Developing a simple articulation of the central idea and sticking to it despite temptations to embellish or extend prematurely.
Leadership	Positioning the technology to be the best of breed; mobilizing people and market for it.

Table 1. Elements of the process of innovation (source: Drucker [4]).

Unexpected events	Unexpected successes or failures; outside events.
Incongruities	A gap between reality and common belief; aspects that do not fit together.
Process Need	A bottleneck in a critical process.
Change of industry structure	New business models, distribution channels, modes of business.
Demographics	Changes in groups by age, politics, religion, income, and so on.
Change of mood or perception	Change in the way people see the world (for example, post-9/11 terrorism), of fashion, convention, and so on.
New knowledge	Application of new knowledge, often involving scientific advances and convergence of different areas.
Marginal practices	Fringe practices that may resolve persistent breakdowns in current central practices.

Table 2. Opportunities for innovation (sources: first seven, Drucker [4]; last, Spinoza et al. [10]).

Drucker devoted over half his book to the search for opportunities, which he grouped into seven categories or sources (see Table 2). The whole point is to look for opportunities in breakdowns, problems, changes, and challenges. The first four sources normally appear internally as challenges to the operation of the business; they can be pursued without the pressure of external competition. The next three sources appear externally, as part of the context in which the firm does business; they are complicated by competition with other firms. The eighth source, marginal practices, is adopted from Spinoza et al. [10]. A marginal practice is an existing practice in another field that may appear irrelevant in yours, but offers an opportunity to solve your problem. Hypertext, for example, was a marginal practice in computing until Tim Berners-Lee transformed it into a central practice of the Web.

Drucker warns that innovating by applying new knowledge is a special challenge. It is the most risky, has a long gestation period, depends on precise timing within a narrow window of opportunity, has numerous competitors, and relies on a convergence of two or more knowledge areas. Even though it is not the only source of innovation, new knowledge gets the most attention.

Drucker says the iconic risk-taking entrepreneur is a myth: successful entrepreneurs engage in the five-part process, which actually reduces their risk. He also maintains that bright ideas are not innovations; without the other four parts of the process, they lead nowhere.

Personal Foundational Practices of Innovation

For the past 10 years I have been teaching engineering students the personal foundational practices of innovation [3]. I have read innovators' stories, talked to them, and watched them carefully, looking for the common patterns in their behavior: I have found eight, as shown in Table 3. The innovation process is unlikely to work if attempted by practitioners who lack these practices.

Awareness	Ability to perceive opportunities and concerns, distinguishing them from your own agenda and concerns; ability to overcome cognitive blindness.
Focus and Persistence	Ability to maintain attention on the mission and avoid distractions; holding to the mission amidst chaos, challenge, or opposition; refusal to give up in the face of obstacles and challenges to the mission.
Listening and Blending	Listening for deeply held concerns and interests and adapting actions to fit ("finding the win-win")
Declarations	Ability to make simple, powerful, moving, eloquent declarations that create possibilities and open attractive new worlds for others.
Destiny	Operating from a sense of a larger purpose than yourself; the purpose drives you.
Offers	Making and fulfilling offers that bring services, practices, or artifacts of value to your customers; organizing groups and managing their commitments toward delivery of the results; maintaining a deep commitment to doing whatever is needed to obtain the results.
Networks and Institutions	Gathering allies, defending against objectors, and creating institutions to further the innovation, develop common standards, and widen its acceptance.
Learning	Making time to learn new skills, acquire new knowledge; making well-grounded assessments in preparation for new learning and action.

Table 3. Personal foundational practices of innovation.

Note that declarations and destiny are closely related, and that offers, networks, and institutions are components of "leadership action." Note also that these skills support "marketing," at which most successful innovators excel.

These practices support Drucker's process. For example, in searching for opportunities, the innovator needs a heightened sense of awareness to be able to actually see them. The central challenge is to overcome "cognitive blindness," a condition where we can't see something and we can't see that we can't see it. Collaborations with other people, especially those of markedly different views or from different communities of practice, can be especially helpful to overcoming cognitive blindness. As another example, the innovator needs to focus on the simple core of the innovation; it takes practice to articulate the essential core and discipline to resist distractions and embellishments.

The personal foundational practices are primarily social, not technical. This is because the primary work of innovation is seeing something missing for other people, defining a new approach, and working with them toward adoption and acceptance. Technical innovations require the innovator to be skilled in both the social dimension as well as the technical; one or the other will not do. It is indeed true that innovation has a social life!

The World Wide Web: A Case Study

Tim Berners-Lee, inventor of the World Wide Web and director of the WWW Consortium (W3C), has written his story about the development of the Web and where he thinks it is going [1]. His book is a rare opportunity to examine a contemporary innovation in detail from the perspective of the person who masterminded it.

Berners-Lee's invention was a Web browsing system on a NeXT computer circa 1990. This system included HTML, a new markup language for documents containing hyperlinks, HTTP, a new protocol for downloading an object designated by a hyperlink, URL, a scheme for global Internet names, and a graphical user interface. Berners-Lee engineered a convergence among these technologies by drawing upon well-known ideas and practices, including Gopher (the University of Minnesota's file fetching system), FRESS and ZOG (hypertext document management systems), SGML (the digital publishing markup language), TCP/IP and FTP (standard Internet protocols), operating systems (the global identifier concept of capability systems), and Usenet news and discussion groups. Some say this invention is not remarkable. What is remarkable is the innovation that came from it. How the seven foundational practices helped the inventor of the Web turn it into an innovation is explored in the following paragraphs.

Awareness. Berners-Lee knew from his conversations with people that many in the 1980s felt the Internet was not moving toward its ultimate promise. It was good for email and limited file sharing, but data incompatibilities and clumsy protocols impeded productive collaboration. He concluded that hypertext would be an enabling technology for collaboration and advancement of global knowledge, envisioning a universal hypertext linking protocol for all Internet computers. At the time, hypertext was a marginal practice in computing, essentially ignored except in a small, vigorous research community. Rather than write research papers about this possibility, he looked for process needs at CERN (his employer) that could be solved by hypertext linking. He found that document sharing among internal scientists and with their external collaborators was a hot issue within CERN and in 1989 he wrote a proposal for a document sharing system based on hyperlinks. This became his avenue for transforming hypertext to a central practice of the Internet.

Listening and Blending. Berners-Lee was taken aback when key individuals did not respond to his proposal, even after he modified it on advice of colleagues who understood the decision-making process. He did not give up. He talked to various people who might be prime users of his proposed technology. He saw ways the technology would help them—for example, giving access to the CERN phone book and to news and discussion groups, or sharing research papers within the high-energy physics community. Through his willingness to blend his ideas with those of his colleagues, he gradually enlisted allies. His ability to listen and blend was key to his ultimate success.

Focus and Persistence. Once Berners-Lee received approval in 1990 to start an implementation, he never wavered from a small set of simple guiding principles: no single controlling authorities, universal identifiers, the markup language HTML, and the protocol HTTP. These principles were realized in his browsing system. He steadfastly maintained they were the essence of the Web; all else would be a distraction. He resisted all efforts to complicate the basic system. He analyzed all new proposals to make sure they were true to these principles.

Declarations and Destiny. Berners-Lee was not afraid to make bold declarations about the kind of world he would like to see evolve under the influence of the Web. His declaration of simple, basic elements became a powerful gathering force. Every chapter of his book is pervaded with a sense of a larger purpose behind his declarations—world intelligence, global learning, social good, economic development, advancing developing countries, reducing world tensions. He declined his friends' urgings to start an Internet company and become wealthy from his own invention because he felt he could not accomplish the mission by wavering from his commitment that all the Web software be in the public domain, free for anyone to use without license.

Learning. Berners-Lee did most of his learning by visiting people, listening to their ideas, and seeing their technology.

Leadership Action. Berners-Lee is a man of action. He accomplished his innovation by making offers, building alliances, and creating institutions. His 1989 proposal was his initial offer. He made critical alliances—Robert Cailliau at CERN helped work the political system at CERN, and Michael Dertouzos at MIT helped establish the W3C, modeled after the successful MIT X Windows consortium. He visited with many people to promote his idea and look for ways to help them achieve their own goals by getting involved. He recruited great programmers to build the prototypes. He campaigned tirelessly to get people to build browsers that could be run on all major platforms; Netscape was the first commercial browser. He leveraged the hypertext community by giving the first public demonstration of the Web at one of their conferences; they were the first to establish Web sites.

When various commercial interests started jockeying to control the technology, Berners-Lee established the W3C to provide a forum for improving and evolving the Web through consensus. With help from powerful allies at MIT, CERN, and INRIA he eventually recruited 450 major companies to join the consortium. He insisted that the consortium abide by the principle of non-control: they develop and promulgate non-binding recommendations, which like the Internet RFCs have become de facto standards.

Ilkka Tuomi wrote his own account of the Web, the Internet, and Linux [11]. He reaffirms that the leaders of those innovations understood they were working for a social change and not just inventing a new technology.

What It Means For You

To be an innovator, you must understand the process, the opportunities, and the foundational practices. A reading program will help to better understand the process of innovation. Read Drucker. Read accounts of particular innovations (for example, Berners-Lee and Tuomi): do you see Drucker's model at work? Have some innovations succeeded without following all five steps? Have some used steps not discussed by Drucker?

Use a journaling practice to train yourself to be an observer of opportunities and a listener of concerns. Regularly record opportunities for innovation that you became aware of. Record your impressions of what people you talked to care deeply about. Record how you overcame cognitive blindness by collaborating with co-workers, especially those of different perspectives.

Learning the personal foundational practices is more difficult. Here you will need a coach or teacher who can show you how to build your awareness, make and embody declarations, maintain a focus, discover a sense of destiny in what you care about, formulate action plans, make offers, and complete actions. Many people have found that a daily meditation practice helps train their ability to focus.

Don't get discouraged by these misconceptions:

- *Innovations must be big*. In fact, an innovation can occur in a small group. Most skilled innovators first learned to make small innovations and increased their scope as they got better at it.
- *Innovations are the work of a few gifted people*. The celebrity innovators are too few in number to account for all innovations. Most innovations come from unheralded people doing their jobs. Anyone can learn the innovation practice and become skilled at it.
- *Innovations depend on novel ideas*. New knowledge, such as is generated in universities and research labs, is one of eight sources of innovation, and is often the riskiest. The other seven sources can be treasure troves of opportunities.
- *Innovations occur only in commercial markets*. Innovations occur in communities of all sizes, in business, government, education, and non-profit sectors.

As you increase your mastery of these practices, you will find yourself making more and bigger innovations. C

References

1. Berners-Lee, T. *Weaving the Web*. Harper Business, 2000.
2. Christenson, C. *The Innovator's Dilemma*. Harvard Business, 1997.
3. Denning, P. The somatic engineer. In R.S. Heckler, Ed., *Being Human at Work*, 2003, 133–143.
4. Drucker, P. *Innovation and Entrepreneurship*. Harper Business (1993). (First published by Harper Perennial in 1985.)
5. Foster, R. *Creative Destruction*. Currency, 2001.
6. Hawn, C. If he's so smart... *Fast Company* (Jan. 2004), 68–74.
7. McKie, S. Let innovation thrive. *Intelligent Enterprise 7* (Jan. 2004).
8. Sahin, K. Our innovation backlog. *Technology Review* (Jan. 2004), 56–57.
9. Slywotzky, A. *Value Migration*. Harvard Business School Press, 1995.
10. Spinoza, C., Dreyfus, H. and Flores, F. *Disclosing New Worlds*. MIT Press, 1997.
11. Tuomi, I. *Networks of Innovation*. Oxford Press, 2003.

Peter Denning (pjd@nps.navy.mil) is past president of ACM.

Digital Village

Hal Berghel and Natasa Brajkovska

Wading into Alternate Data Streams

The open-ended nature of ADSs makes them an extremely powerful Windows resource worthy of deeper exploration.

The concept of non-monolithic file systems is not new. "File forks" were an integral part of the original Macintosh Hierarchical File System. Apple separated file information into data and resource forks, where the resource fork contains a resource map followed by the resources, or links to resources, needed to use the data. Typical resources would include program code, icons, font specifications, last location of application window, and other file-related metadata that is not stored in the disk catalog (for example, file signatures). File forks may be null, and many files contain both forks, the accidental separation of which produces the dreaded "error 39." The Macintosh Resource Manager could control up to 2,700 resources per file, though the linear search, single-access control strategy makes managing more than a few hundred resources cumbersome and slow. File forks remained in use until Apple introduced OS X. Built on a Linux kernel, OS X stores resources in separate .rsrc files.

Microsoft introduced non-monolithic file structures in its New Technology File System (NTFS) along with the Windows NT operating system. Since the loss of a non-null resource fork renders the data fork useless, Microsoft had to accommodate non-monolithic files for compatibility with Macintosh files and Apple Talk. Among the many innovations in NT, Microsoft ported non-monolithic file structures from the Macintosh world over to the PC. In Microsoft parlance, non-monolithic file structures are called file streams.

MICHAEL SHCRÖTER

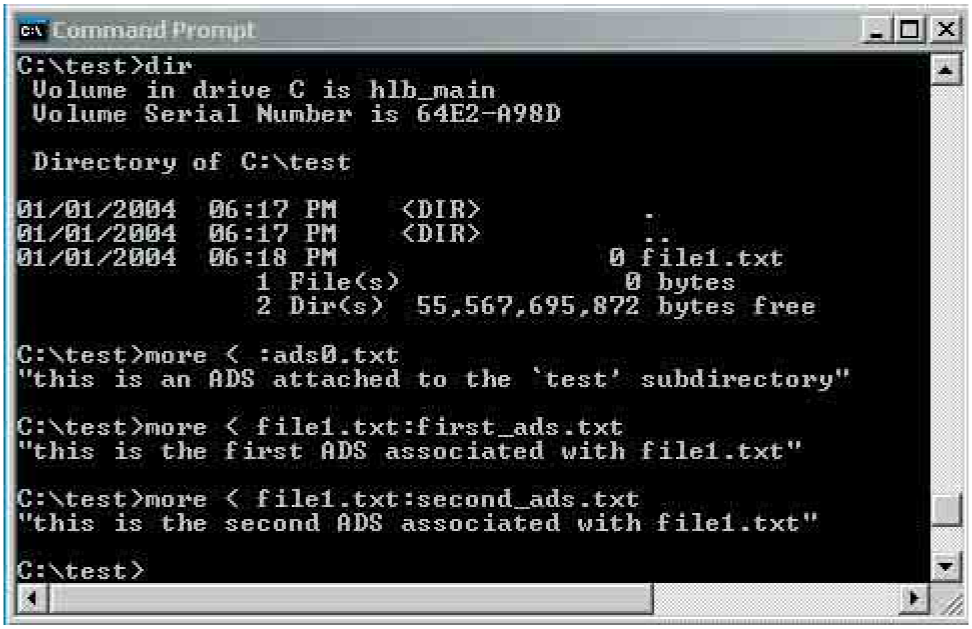
Alternate Data Streams

One may make reasonable comparisons between the Macintosh data and resource forks and the Microsoft primary and alternate data streams, respectively. In NTFS the primary data stream (aka default data stream or unnamed data stream or, simply, file) is called $DATA. A large number of alternate data streams (ADSs) may be associated with a single primary data stream (PDS). We emphasize that ADSs are associated with, and not attached to, primary data streams. The associations are maintained in the Master File Table (MFT), and managed by a variety of application program interfaces (APIs). As a simple illustration, a right mouse click on any NTFS file and subsequent selection of the properties tab will recover ADS metadata through a standard Windows MFT API.

Microsoft's approach to non-monotonic file structures has some unusual properties:

- Anything digital can become an ADS: thumbnails and icons, metadata, passwords, permissions, encryption hashes, checksums, links, movies, sound files, binaries, whatever. We'll return to the "whatever" in a moment.

- NTFS supports a very large number of ADSs. The Purdue COAST project estimated an upper bound of 6,748 ADSs per primary data stream on Windows NT 4. (see www.cerias.purdue.edu/coast/ms_penetration_testing/v50.html).



```
C:\test>dir
 Volume in drive C is hlb_main
 Volume Serial Number is 64E2-A98D

 Directory of C:\test

01/01/2004  06:17 PM    <DIR>          .
01/01/2004  06:17 PM    <DIR>          ..
01/01/2004  06:18 PM                 0 file1.txt
               1 File(s)              0 bytes
               2 Dir(s)  55,567,695,872 bytes free

C:\test>more < :ads0.txt
"this is an ADS attached to the `test' subdirectory"

C:\test>more < file1.txt:first_ads.txt
"this is the first ADS associated with file1.txt"

C:\test>more < file1.txt:second_ads.txt
"this is the second ADS associated with file1.txt"

C:\test>
```

Figure 1. Recovering hidden Alternate Data Streams.

- Since ADSs are separate files, they appear hidden from applications that call routine Windows APIs. The use of ADS is completely transparent to standard Windows file management tools—it's as if they don't exist. This includes such key system applications as Windows Explorer and Task Manager, and core command line executables like DIR.
- The normal data stream ADS API syntax in the Windows NT/2000/XP series is <filename>:<ADSname>:<ADStype> (where ADStype is optional).
- ADSs form a tree structure that has a maximum depth of 1. The decision to prevent nesting of ADSs seems to us to be an unnecessarily arbitrary and shortsighted limitation.

The best way to get a feel for ADS is with some hands-on experience. Our example shown in the sidebar here may seem clumsier than necessary at first, but it has the benefit of simplicity because the entire demonstration can be completed within a single DOS command prompt window.

Phishing and Executable Streams

As previously stated, ADSs may contain anything—text, images, sound and video files—anything. The most interesting type of "anything" is the binary executable. Presuming readers of this column have completed the example in our sidebar and are up for the challenge, we'll now attach an executable to an empty text file. In this case we're assuming the Windows XP system calculator is stored in the location C:\windows\system32\calc.exe. If not, create a path to the harmless executable of choice.

We now rename <calc.exe> as the ADS, <d.exe>, and associate it with the empty text file <test.txt>:

```
C:\...\test>type c:\windows\
system32\calc.exe > .\test.txt: d.exe
```

and execute the ADS directly:

```
C:\...\test>start .\test.txt:d.exe
```

You should see the Windows calculator appear on

ADS Example

This exercise was run on Windows XP Pro with administrative privileges. Similar results would result if it were run under Windows 2000 or even Windows NT, with slight adjustments to the syntax.

First, open a DOS command prompt window. Then make and move into a new directory—in our case <test>. This step is important, because we'll need to remove this entire directory after our experiment to restore the computer to its pre-experiment state. Do not perform this experiment in a root directory, or any other directory that you are unwilling to erase. The commands

```
mkdir test
cd test
```

will create a new directory, <test>, to experiment in. We then create the simplest of ADS, one that is attached to a folder, as follows:

```
C:\...\test>echo “this is an ADS attached
to the ‘test’ subdirectory” > :ads0.txt
```

A display of the contents of the directory:

```
C:\...\test>dir
```

reveals an empty directory. How can that be? This is where the NTFS sleight-of-hand comes in: ADS are hidden to Windows applications that rely on the standard MFT APIs. "DIR" is one such example, as is Windows Explorer and the file properties box.

In the preceding example, since the PDSname field is null, <ads0.txt> is by default associated with the subdirectory name in the MFT. Directories in Windows are themselves files that reference other files.

Next, we'll attach an ADS to an empty file:

```
C:\...\test>echo “this is the first ADS
associated with file1.txt”>
file1.txt:first_ads.txt
```

Note that we never created <file1.txt> to begin with. Since the colon is not a legitimate filename character, the Windows command processor understands that the data to be echoed is intended for the associated ADS. Since the ADS has to be associated with something in the MFT, Windows conveniently creates an empty file named <file1.txt> for the association. In Microsoft parlance, we think of this as a single file where <file1.txt> is the primary/default/unnamed data stream, and a named stream labeled <first_ads.txt>.

This time the display of the contents of the directory reveals only an empty file <file1.txt>. Now, let's add a second ADS to file ads1.txt

```
C:\...\test>echo “this is the second ADS
associated with file1.txt”>
file1.txt:second_ads.txt
```

Repeat

```
C:\...\test>dir
```

and observe that nothing has changed. We still have an empty file, <file1.txt>, in a directory consisting of 0 bytes.

Appearances to the contrary, we may confirm that these ADSs do indeed exist by typing the following:

```
C:\...\test>more < :ads0.txt
C:\...\test>more <file1.txt:first_ads.txt
C:\...\test>more <file1.txt:second_ads.txt
```

Your results should look like those in Figure 1.

Of course, the more useful ADS will be associated with non-null files. To wit:

```
C:\...\test>echo “this is the data for
file 2” > file2.txt
C:\...\test>echo “this is the data for
ADS1 of file 2” > file2.txt:ads1.txt
C:\...\test>echo “this is the data for
ADS2 of file 2” > file2.txt:ads2.txt
```

The remaining 6,746 repetitions are left to the reader.

your screen. A directory listing still reveals only primary data streams. The executable is entirely hidden, but very much there—and obviously active—as can be confirmed by looking at the active processes listing under Windows Task Manager by pressing the <CTL><ALT><DEL> keys simultaneously (see Figure 2).

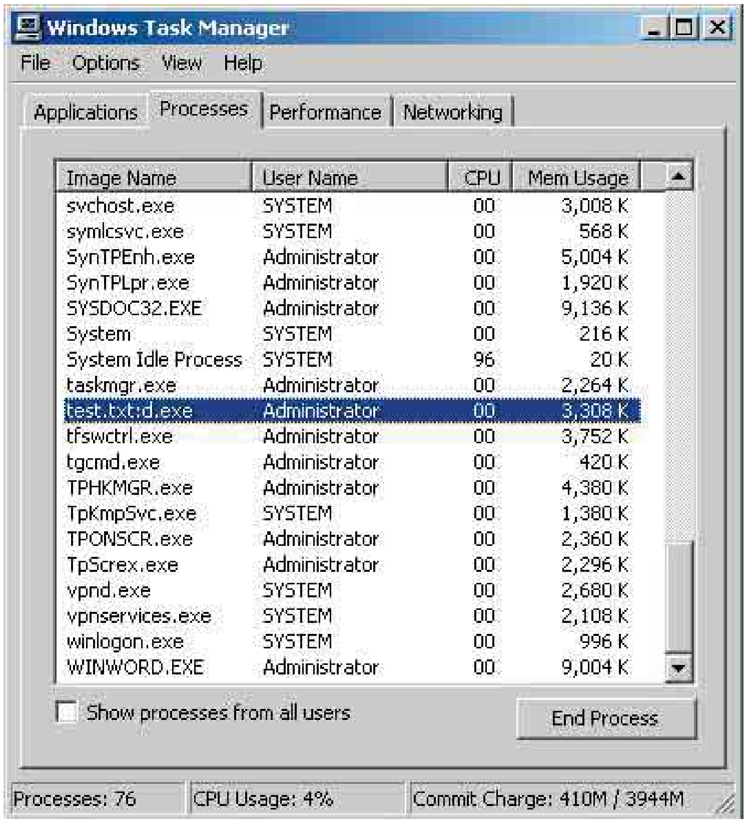


Image Name	User Name	CPU	Mem Usage
svchost.exe	SYSTEM	00	3,008 K
symlcsvc.exe	SYSTEM	00	568 K
SynTPEnh.exe	Administrator	00	5,004 K
SynTPLpr.exe	Administrator	00	1,920 K
SYSDOC32.EXE	Administrator	00	9,136 K
System	SYSTEM	00	216 K
System Idle Process	SYSTEM	96	20 K
taskmgr.exe	Administrator	00	2,264 K
test.txt:d.exe	Administrator	00	3,308 K
tfswctrl.exe	Administrator	00	3,752 K
tgcmd.exe	Administrator	00	420 K
TPHKMGR.exe	Administrator	00	4,380 K
TpKmpSvc.exe	SYSTEM	00	1,380 K
TPONSCR.exe	Administrator	00	2,360 K
TpScrex.exe	Administrator	00	2,296 K
vpnd.exe	SYSTEM	00	2,680 K
vpnservices.exe	SYSTEM	00	2,108 K
winlogon.exe	SYSTEM	00	996 K
WINWORD.EXE	Administrator	00	9,004 K

☐ Show processes from all users

Processes: 76 CPU Usage: 4% Commit Charge: 410M / 3944M

Figure 2. Windows Task Manager's report of the Windows calculator executing as the Alternate Data Stream <test.txt:d.exe>.

It is apparent that with minimal effort one can sufficiently mask the hidden executable so that its function is obscured. Of course, masking an executable by just changing the filename is neither clever nor particularly deceptive, so a hacker might add the requisite stealth by invoking the native Windows Scripting Host with the control parameters set to execute files with non-executable file extents. In this way one could rename <trojan.exe> as something innocuous like <help.fil>, and execute it with WSH.

It is interesting to note that prior to Windows XP, the ADS didn't even appear in the process listing. Had we hidden the ADS behind something innocuous like <cmd.exe> or <notepad.exe>, the execution of the ADS would be undetected.

The hiding of the function behind an innocuous appearing executable is akin to Internet scams where the unsuspecting are lured to spoofed Web sites that appear harmless while actually harvesting personal or private information—a technique called "phishing." For lack of a better term, we may think of planting hostile executables in ADS as "file phishing": creating an environment in which things are not as they appear.

Before we proceed, let's clean up the data streams, directories, and files on your computer. ADSs can only be deleted if their associated primary data stream is deleted, so once you're done experimenting, erase all of the files in your test directory, go up one directory and remove the directory itself, or simply erase the entire <test> directory with Windows Explorer.

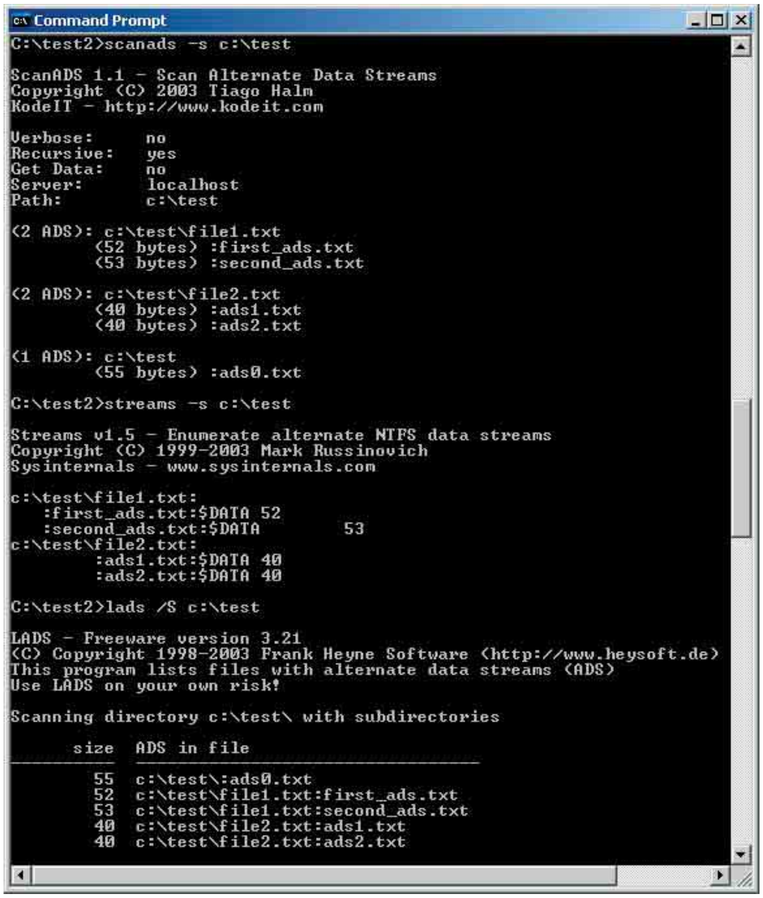
At this point you're back where you started, no worse for wear.

NTFS Master File Tables

To understand ADS, one must investigate the way the Windows MFT record works. The MFT is a relational database in which the records correspond to files and the columns to file attributes. Following 16 records of metadata files, there is at least one MFT record for each file and folder (subdirectory) on all hosted disk volumes. All records are 1Kb in size, allowing the data and attributes of very small files or folders to be stored in an MFT record itself. Larger files and folders are referenced by pointers within their records. Folders are externally organized as B-trees.

Each record contains basic file attributes relating to date and time stamps, permissions, filename and extent, security descriptors, and so forth. The utility of the MFT results from the fact that it is set up to automatically use external pointers for all data that cannot fit within the 1Kb record. Having this in place allows virtually unrestricted, scalable control over file management. All data streams (regardless of whether they're primary or alternate) maintain all of the necessary information for the Windows APIs to manipulate them: for example, allocation size, actual data length, whether they're compressed or encrypted, and so forth. Given this situation, the proliferation of data streams involves little more than the proliferation of pointers within the MFT. The only mitigating factor is that the ADSs cannot be nested, which means any ADS file organization beyond a one-level deep tree would have to be accomplished at the applications layer.

As we indicated, the low-level Windows APIs (such as CreateFile, DeleteFile, ReadFile, WriteFile) were designed to treat all data streams the same, ADSs or PDSs. Under NTFS, ADS support is completely transparent at that level. The weakness is that the higher-level utilities (DIR, Windows Explorer, Task Manager) were not intended to support ADS. This is where the truly baroque nature of Microsoft's ADS design logic makes itself known. One can use the low-level APIs to manipulate ADSs easily, but the higher-level utilities conceal their presence. From the end user's perspective, it's hard to delete a data stream without first being able to find it! Fortunately there are third-party utilities such as Lads, ScanADS, Streams, and Crucial that help locate ADS by working directly with the low-level APIs (especially, the "backup_" functions). Figure 3 illustrates their use on our <test> directory after we completed the experiment described previously. Note that Streams requires a separate test for folder ADS (remove the "-s" parameter). Crucial has a GUI interface and only scans entire drives, and will not be shown here.



```
C:\test2>scanads -s c:\test

ScanADS 1.1 - Scan Alternate Data Streams
Copyright (C) 2003 Tiago Halm
KodeIT - http://www.kodeit.com

Verbose:     no
Recursive:   yes
Get Data:    no
Server:      localhost
Path:        c:\test

(2 ADS): c:\test\file1.txt
        (52 bytes) :first_ads.txt
        (53 bytes) :second_ads.txt

(2 ADS): c:\test\file2.txt
        (40 bytes) :ads1.txt
        (40 bytes) :ads2.txt

(1 ADS): c:\test
        (55 bytes) :ads0.txt

C:\test2>streams -s c:\test

Streams v1.5 - Enumerate alternate NTFS data streams
Copyright (C) 1999-2003 Mark Russinovich
Sysinternals - www.sysinternals.com

c:\test\file1.txt:
   :first_ads.txt:$DATA 52
   :second_ads.txt:$DATA        53
c:\test\file2.txt:
        :ads1.txt:$DATA 40
        :ads2.txt:$DATA 40

C:\test2>lads /S c:\test

LADS - Freeware version 3.21
(C) Copyright 1998-2003 Frank Heyne Software (http://www.heysoft.de)
This program lists files with alternate data streams (ADS)
Use LADS on your own risk!

Scanning directory c:\test\ with subdirectories

      size  ADS in file
----------  ---------------------------------
        55  c:\test\:ads0.txt
        52  c:\test\file1.txt:first_ads.txt
        53  c:\test\file1.txt:second_ads.txt
        40  c:\test\file2.txt:ads1.txt
        40  c:\test\file2.txt:ads2.txt
```

Figure 3. Typical ADS reporting utilities at work.

Security Implications of ADSs

A Google search for the phrase "alternate data streams" will yield several thousand hits, most of an alarmist nature. This is unfortunate in many ways, because the power of ADSs has yet to be realized. While it is true that there is malware that takes advantage of ADSs (W2k.stream is one example), that malware has not proven to be as destructive as the more mainstream varieties that rely on buffer overflows, NetBIOS and RPC vulnerabilities, session hijacking, or address spoofing. As a datapoint, all W2k.stream threat vectors were assessed "low" by Semantec (see www.sarc.com/avcenter/venc/data/w2k.stream.html).

What created most of the alarm was the "hidden" nature of ADS combined with the absence of Microsoft utilities that supported direct access and control within native file utilities—but, then, that wasn't why Microsoft built ADS into NTFS in the first place. The mere mention of a hidden feature to anyone with even a slight anti-Microsoft bias is guaranteed to produce an animated response. Unfortunately, Microsoft added fuel to the fire by failing to include a "display ADS" checkbox as a Windows Explorer configuration option and direct ADS control at the utility level. Most users wouldn't be bothered with ADS management, but full file disclosure would have been comforting to those prone to anxiety attacks.

URL Pearls

A good overview of Macintosh file forks is available at the Apple Web site: developer.apple.com/documentation/mac/Files/Files-14.html.

Microsoft's MSDN Library (msdn.microsoft.com/library/) is a reliable source of information on Windows NT operating systems, including APIs and ADS. Detailed documentation on NTFS is available at the Microsoft Technet site (www.microsoft.com/technet/).

The 1998 overview of NTFS by Richter and Cabrera in the *Microsoft Systems Journal* at www.microsoft.com/msj/1198/ntfs/ntfs.aspx is still worth reading. This article also has links to sample code.

The general-purpose Windows utility that is ideal for ADS manipulation is <cp.exe>, variations of which may be found in either the Windows Resource Kit or from the Sourceforge.net GNU utilities library at unxutils.sourceforge.net, the latter being free.

Two good sources for the fundamentals of ADS programming are Dino Esposito's *A Programmer's Perspective on NTFS 2000 Part 1: Stream and Hard Link* at msdn.microsoft.com/library/en-us/dnfiles/html/ntfs5.asp?frame=true, and Eugene Kaspersky and Denis Zenkin's, "NTFS Alternate Data Streams," Document 19878 in *Windows and .NET Magazine* (www.winnetmag.com). Esposito's article contains an interesting code fragment that adds a new property page with stream information. The Code Project (www.codeproject.com/csharp/CsADSDetectorArticle.asp) also contains useful code fragments.

Effective ADS detectors include ScanADS (www.securiteam.com/tools/5JP0C2KAAQ.html), LADS (www.heysoft.de/Frames/f_sw_la_en.htm), CrucialADS (www.crucialsecurity.com), and Streams (www.sysinternals.com/ntw2k/source/misc.shtml).

The facts are less alarming than the several thousand Google hits might lead us to believe. While it is true that ADS could be used as a conduit for malware executables, so can email attachments. Further, modern anti-virus scanners routinely scan for malware in all Windows data streams, including ADS, so the risk of intrusion should be no greater with ADS than PDS.

The same is true for covert channeling. ADS could be used for that purpose, but so could the options field in a normal ICMP packet. With the ease that malware such as Loki conducts encrypted covert data channeling at the IP level, why would a hacker become involved with the applications layer?

The claim that ADSs are difficult to delete is equally misleading. ADS file space gets reallocated in just the same way that PDS and directory space does. File-wiping products such as Cyberscrub (www.cyberscrub.com) even include ADS "scramblers" for extra safety.

By any reasonable measure, ADS vulnerability has been overstated.

Conclusion

Alternate Data Streams have demonstrated considerable potential in object-oriented OSs and application environments, or those that involve complex file and data structures. While Microsoft is officially committed only to the Object Linking and Embedding (OLE) 2.0 model of structured storage, ADS will likely remain with us as long as Windows OSs continue to use NFTS. To quote Microsoft:

"Alternate data streams are strictly a feature of the NTFS file system and may not be supported in future file systems. However, NTFS will be supported in future versions of Windows NT. [including Windows 2000 and XP] *Future file systems will support a model*

based on OLE 2.0 structured storage (IStream and IStorage). By using OLE 2.0, an application can support multiple streams on any file system and all supported operating systems (Windows, Macintosh, Windows NT, and Win32s), not just Windows NT." (See the Microsoft Knowledge Base article "HOWTO: Use NTFS Alternate Data Streams" (number 105763), available at support.microsoft.com/default.aspx?scid=kb;en-us;105763.)

There is no question that ADSs are underutilized in Windows. Like previous major software houses, Microsoft felt compelled to opt in favor of backward compatibility. For example, to use <desktop.ini> files to parse the contents of a directory and <.tmp> files to hold transitory data, seems retrogressive at best when ADS could have accomplished the same thing in a far more straightforward manner. After all, neither file type has any meaning apart from the associated directory or primary data stream anyway, so using ADS is the natural way to handle them. But, that would have meant that such information could not be shared with Windows platforms with FAT16 and FAT32 file systems. To hobble the OS is less costly than dealing with 40 million additional hits at the help desk.

But the most unfortunate aspect of ADS is that the negative press and exaggerated claims of vulnerability have polluted the waters to such as degree that the true potential of ADS may never be realized. C

Hal Berghel (www.acm.org/hlb) is a professor and director of the School of Computer Science and director of the Center for Cybermedia Research at the University of Nevada, Las Vegas.
Natasa Brajkovska (natasa@crlmail.i2.nscee.edu) is a research assistant at the Center for Cybermedia Research at the University of Nevada, Las Vegas.

Eric M. Uslaner

Trust Online, Trust Offline

Online businesses would do well to cultivate their visitors' adventurous, trusting, optimistic, risk-taking world views.

LISA HANEY

Commercial Web sites have been making concerted efforts to reassure their customers that their transactions and personal information are safe. This site won't violate your privacy, we are told. This is all to the good, but Web site designers must also be concerned with something over which they have no control: Are people generally trusting? If not, they may steer away from risky activities, including Web commerce.

Over a recent lunch with a survey researcher who works at a nonprofit institution in Washington, D.C., we discussed my work on trust. I had just finished a talk at his organization where I distinguished between strategic trust, or the kind of trust that reflects our experience with particular people doing particular things, and moralistic (or generalized) trust, or a more general value we learn early in life. Strategic trust can help us decide whether a specific Web site is safe: Is our information secure there? Will it install spyware on our computers? Will it redirect us to places we don't want to go on the Internet? And if it's a commercial Web site, will it deliver what it promises? We can learn about these sites by reading about them or by our own direct experience.

Moralistic trust is not based upon everyday experience but instead on an optimistic world view we learn at an early age: The world is a good place; it is going to get better; I can make it better; and it is thus not so great a risk to agree that most people can be trusted. Moralistic trust won't tell us anything about a particular Web site or personalities on the Web. However, moralistic trust will give us sufficient faith to take risks on the Web in the first place. Mistrusters will simply stay away altogether.

My colleague asked: You say trust is not based upon experience, but what about the Internet? All sorts of danger lurks there; just to get online one must establish a firewall, because people are constantly trying to hack into your system. He runs a virus checker constantly and at least once a week uses a spyware search utility to see which companies are trying to track his every move. His email has a spam filter to isolate the dozens of daily invitations to pornographic Web sites and other attempts to sell him stuff he doesn't want. Then there is the teenager haven of instant messaging, which, we now learn, is a major source of identity theft online. So how can we expect people to suffer through all of this insecurity and still believe that most people can be trusted?

My colleague thinks the Internet is a source of trust and mistrust. But the Internet really depends upon trust rather than creates trust. People who trust each other are more likely to be comfortable with new technology, even if they don't trust particular Web sites. Most online retailers go to great lengths to demonstrate their trustworthiness. EBay's "Safe Harbor" promises a (limited) guarantee against online fraud, giving each seller a rating for integrity. Online merchants trade in strategic trust, the good reputation of sellers, and the company itself. The company that claims to ensure privacy online calls itself Truste. This is all to the good, but moralistic trust plays a more important role in shaping whether people view the Internet as a great new opportunity or as a threat.

Consider the optimists first: Michael and Ronda Hauben in their 1997 book *Netizens* called the Internet "a grand intellectual and social commune in the spirit of the collective nature present at the origins of human society." Others view new Web sites (such as friendster.com) as great hopes for getting people together in pursuit of common goals or just simple love. Meetup.com brought 150,000 Americans together to support Howard Dean, who set records for raising campaign funds on the Internet in 2003. Everett Ehrlich, a former U.S. Undersecretary of Commerce, wrote in The *Washington Post* that the days of political parties are over; citizens can now take politics away from the professionals and put it into their own hands.

Alex Salcedo, a 13-year-old boy from Kensington, MD, was hit by a car near his home in 1999. He was rushed to a local hospital and put into an induced coma; the prognosis was not good. His father created a Web page so the family could keep apprised of his medical condition; eventually Alex died. The Web site received 66,000 hits, with 2,000 messages posted.

Mistrusters bring us back to reality. Kaycee Nicole Swenson, a 19-year-old girl from Kansas suffering from leukemia, created a blog of her illness and remissions called Living Colours; thousands of people visited her site for almost a year. Many sent her gifts, and all were sad when the site announced her death May 15, 2003. It was all a hoax; there never was a Kaycee. The blogger was really a 40-year-old woman with two healthy children.

Back to reality again. Friendster has almost four million registered users, but nobody knows how many of them are real people. The *New York Times* (Nov. 27, 2003) reported 2,619 "pretendsters" have been terminated by the site, but it is still pretty easy to go on a cyberdate with someone who isn't really there.

Mistrusters would argue that Kaycee Swenson illustrates you can't be too careful dealing with people. It might do little harm to read her story, but don't send her gifts, and be very wary of sending anyone money on the Internet. Trusters would say miscreants are everywhere, and a handful of bad experiences should not cause us to withdraw from social connections, especially with people we don't know.

The Internet can seem a trusting or a mistrusting place. But it is largely a reflection, perhaps in bas relief, of the larger society. All around us we see both nice people and scoundrels. The Internet can scare away someone who already doesn't trust other people, as I showed in the *2000 Trust and Privacy Survey* of the Pew Internet and American Life Project. Mistrusters overestimate the amount of risk in their worlds. Hackers might steal their credit card number; businesses can get their personal information; their Web dealings might not be private; others will know where they have been on the Web; they might download viruses; and others will learn private things about their lives. Mistrusters are less likely to use their real names on the Web and more likely to use fake identifications and email addresses.

People who trust others underestimate the risks of daily life and are more likely to go online and give out personal information. They adapt more quickly to new technologies and dismiss the chance that their computers will be infected by viruses when they're online.

Before deciding whether to buy from an online retailer, we must first be comfortable with new technology. Moralistic trusters are risk takers and don't feel threatened by online transactions. Online retailers should pay as much attention to this type of trust as they do to their own trustworthiness. At a site like eBay, you can fall prey to sellers who manipulate their ratings or to people with no history at all but who offer deals too good to be true; I was taken in by both.

Strategic trust may lead people away from online commerce after bad experiences. Moralistic trusters look at a negative experience as just that and discount bad news. A new technology like the Internet needs more than just the strategic trust a retailer can provide. Its future may depend upon a more general faith in humanity, not just in machines. **C**

Eric M. Uslaner (euslaner@gvpt.umd.edu) is a professor of government and politics in the Department of Government and Politics at the University of Maryland–College Park.

Moralistic trust plays an important role in shaping whether people view the Internet as a great new opportunity or as a threat.

Human-Computer Etiquette: MANAGING EXPECTATIONS *with* INTENTIONAL AGENTS

By Christopher A. Miller, Guest Editor

When I hit my thumb with a hammer, I get mad at myself, at the person making me do the work, and at the cruel and malignant universe, but I don't get mad at the hammer. By contrast, when my computer crashes, or fails to give me a file that I know is there, or tells me that I failed to shut it down properly when it was the one that crashed, I get mad at *it*. True, if I stop to think about it, I may get mad at the people who designed, programmed, or last updated it, but my immediate reaction is more personal. Even my language, as I write this, is illustrative: *I* hit *myself* with the hammer, while my computer *does* things to me.

ILLUSTRATION BY LISA HANEY

Why should that be? The computer is, at some level, just a hunk of silicon and plastic—every bit as inert as the hammer. Part of the answer has to do with its complexity and autonomy, and with its accompanying unpredictability in use. Somewhere between the hammer and the computer lies a threshold of complexity and capacity for autonomous action that, when surpassed, tempts us to ascribe agent-like qualities such as awareness and intent.

We interact with agents on a social level, and increasingly so with increases in the complexity of behavior and of interaction which the agent affords [2, 8]. As Brown and Levinson [1] point out, social interactions are full of ambiguities, especially in understanding another's beliefs, intents, and mental processes. Since we can't predict or understand all of an agent's actions, we develop codes and expectations about how agents *should* behave in various roles and contexts. These codes enable us to infer what specific behaviors mean, in terms of the agent's intentions and beliefs. Not surprisingly, we have evolved a wealth of such expectations for interactions with the most complex and unpredictable agents with whom we regularly work and live—other humans. Some of these codes are quite explicit—such as the communication protocols in which pilots and air traffic controllers are trained [3], or the rituals of a religious service—but others are subtle and largely implicit. These patterns or rules of expectation are what we call "etiquette."

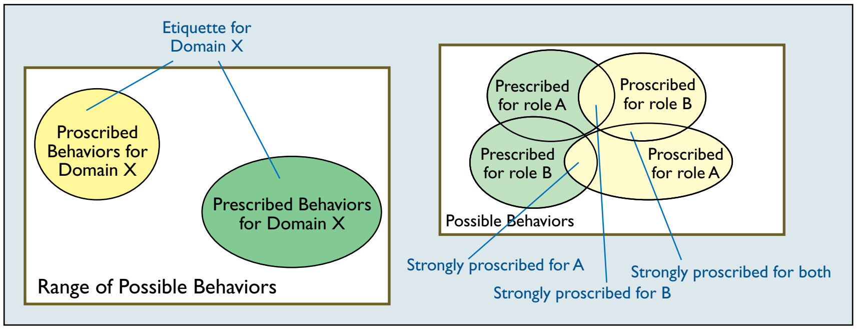

Etiquette for Domain X
Proscribed Behaviors for Domain X
Prescribed Behaviors for Domain X
Range of Possible Behaviors
Prescribed for role A
Proscribed for role B
Prescribed for role B
Proscribed for role A
Possible Behaviors
Strongly proscribed for A
Strongly proscribed for both
Strongly proscribed for B

Figure 1. Illustration of etiquette as prescribed and proscribed behaviors (collectively and separately) by role.

Etiquette has two related definitions in common usage [5]: Etiquette is a (frequently implicit) set of prescribed and proscribed behaviors that permits meaning and intent to be ascribed to actions, thus facilitating group identification, streamlining communication and generating expectations; and etiquette encodes "thoughtful consideration for others," that is, etiquette operates (when obeyed) to make social interactions more pleasant, polite, and cooperative and (when violated) to make them insulting, exploitative, and unpleasant. In both senses, etiquette defines behaviors expected of (or prohibited to) agents in specific contexts and roles (see Figure 1). Etiquette allows me to make predictions about what those around me will do (for example, thank me if I give them a gift, acknowledge my presence if I enter a room) and to ascribe meaning to their behaviors or lack of the same (for example, they didn't like the gift; they didn't notice me).

Computers long ago bypassed the "agentification barrier" and regularly elicit expectations and responses on our part as if they were human actors. In an extensive series of experiments, Reeves and Nass [9] demonstrated that humans frequently exhibit behaviors with computers similar to their behaviors with other humans. Such behaviors include attraction to agents (whether computer or human) whose characteristics are similar to their own, being less critical to an agent directly versus "behind its back," and being more willing to accept and believe in flattery versus criticism from an agent.

Reeves and Nass did not need to modify a basic windows and mouse interface to encourage perception of the computer as human. Humans readily generalize their expectations from human-human interaction to human-computer interaction *regardless* of whether or not that is the intent of system designers.

Nevertheless, comparatively little attention has been paid to understanding and manipulating this dimension of human-computer interaction. Since a computer system will be perceived in light of the etiquette behaviors it adheres to or violates, it behooves designers to consider what etiquette our systems *should* follow or flout to elicit appropriate perceptions. For most home-based usage purposes, this might mean politeness, subservience, helpfulness, and "the sensitivity of an intuitive, courteous butler" [4], but those might be inappropriate behaviors to exhibit to a pilot or a power plant operator. The study of Human-Computer Etiquette (HCE) should embrace how to make computers more polite or human-like and how to avoid those inappropriate effects when necessary.

My own journey along this path began with work on cockpit-aiding systems for fighter and rotorcraft pilots—not a population typically interested in polite or considerate behavior. Nevertheless, we noted that human pilots in dual-crew cockpits spent as much as a third of their time in intercrew coordination activi-

Taking the etiquette perspective **in design means acknowledging that complex, semiautonomous technologies will be regarded as agents and that human interactions with them will be colored by expectations from human-human etiquette.**

ties, that is, in "meta-communication" about their intents and plans. In designing a Cockpit Information Manager able to determine the pilots' needs and intents and to reconfigure cockpit displays to support those activities [7], we believed such a system would need to participate in meta-communication, taking instruction and declaring its intent and its understanding of the pilots' intent. We designed and implemented a simple interface that provided these capabilities in a minimal fashion (see Figure 2). Introducing this interface improved human + machine system performance, and contributed to gains in user acceptance [6]. In hindsight, I believe these improvements came from fitting into the existing etiquette that pilots expected from any new actor in the domain. The interface we implemented did not follow etiquette in the sense of politeness, but it did behave according to the established conventions for any agent filling that functional role.

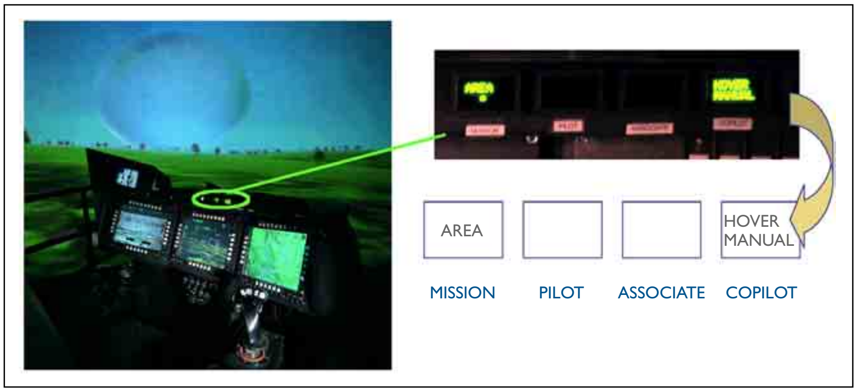


Figure 2. The Crew Coordination and Task Awareness Display of the Rotocraft Pilot's Associate. The system reports its interferences about high-level mission activities of task names; pilots can override via button presses.

Taking the etiquette perspective in design means acknowledging that complex, semiautonomous technologies will be regarded as agents and that human interactions with them will be colored by expectations from human-human etiquette. Taking the etiquette perspective forces us to consider aspects of human-computer relationships that traditional approaches do not. By placing the system to be designed in the role of a well-behaved, human-like collaborator, we gain insights into how users might prefer or expect a system to act. We can also infer how system actions (and failures to act) might be interpreted by users. Such insights rarely come from other design approaches (with the possible exception of usability reviews of systems already designed and implemented). I find it instructive to ask two questions of potentially agentified technologies: If this system were replaced by an ideal human assistant, albeit one constrained to act through the interface modalities available to the system, how would that assistant behave?" Alternatively; If a human assistant were to provide this interaction in this way, how would he or she be perceived by colleagues? To pick (unfairly, I acknowledge) on a well-known example: How would I regard a human office assistant who, several times a day, interrupted my work to offer me help writing a letter?

This collection of articles illustrates the beginnings of research that overtly considers etiquette in design and evaluation of human-computer interactions. Some research explicitly manipulates etiquette behaviors to get a desired effect; other research considers the effect of etiquette in the analysis of existing or prototyped systems. Clifford Nass begins with a review of his seminal work demonstrating that humans apply the same etiquette to human-computer interactions that they do to human-human interactions in many cases. He speculates on what dimensions of behavior in an interaction are prone to activate our expectations of human-like etiquette behaviors (and our willingness to provide them).

Timothy Bickmore reports recent work on Embodied Conversational Agents (ECAs)—computer systems with an explicit face and body that enable them to exhibit very complex and subtle etiquette behaviors we associate with body language, facial expressions, and so on. By virtue of their sophisticated and human-like physical embodiment, ECAs represent the high (or at least, most complex) end of the spectrum of computer agents that exhibit etiquette. Bickmore summarizes the range of conversational functions that etiquette plays and provides examples of several ECA systems striving to achieve these functions.

Punya Mishra and Kathryn Hershey discuss the role of etiquette in pedagogical systems, where issues of motivation, interest, and establishment of roles between student and teacher make the use of etiquette behaviors critical. They review the results of experiments that explicitly manipulate the implications of Nass's paradigm—that humans frequently react to computers as if they were other humans in social settings—in pedagogical applications and show some of the strengths and some of the limitations of this approach.

If the ECAs that Bickmore describes represent the high end of systems striving to achieve a wide range of human-like etiquette behaviors, Raja Parasuraman reports on work generally found at the other end of that spectrum—human interaction with high-criticality automation in domains such as aviation, power generation, and military systems. Here the consequences of human misunderstanding of automation capabilities and behaviors can be catastrophic and, consequently, there has been extreme skepticism about machines that exhibit subtle, human-like behaviors, much less politeness. Nevertheless, Parasuraman summarizes factors that produce and tune human trust in complex automation and then reports an experiment demonstrating that etiquette (whether good or bad) is a factor that must be included in that list since its effects are as significant as a 20% variance in automation reliability.

While the other authors examine direct human-computer interaction, Jennifer Preece is more interested in the effects the computer can introduce into computer-mediated human-human interaction—how computer technology can enhance or disrupt the etiquette of human-human, face-to-face interaction by introducing its own artifacts to distort that interaction. She reports survey data of what users perceive as etiquette violations in Internet interactions, analyzes causes of these perceived violations, and discusses the efficacy of potential solutions in the form of explicit "netiquette" rules and appeals to role models in the form of moderators and early adopters for smoothing the adoption of new etiquettes in different online settings.

As computers become more complex, smarter, and more capable, and as we allow them to take on autonomous or semiautonomous control of more critical aspects of our lives and society, it becomes increasingly important to define styles, norms, roles, and even mores of human and computer relationships that each side can live with. The rules that govern such relationships are etiquette rules; here we argue those who design and analyze human-computer interaction must become aware of those rules and how to incorporate their effects. The articles in this section illustrate a range of methods and outcomes that result from taking the etiquette perspective on human-computer interaction and, therefore, provide us a guide to the terrain we must explore more fully in the future. C

References

1. Brown, P. and Levinson, S. *Politeness: Some Universals in Language Usage.* Cambridge University Press, Cambridge, UK. 1987.
2. Dennett, D.C. *Brainstorms: Philosophical Essays on Mind and Psychology.* MIT Press, Cambridge, MA, 1978.
3. Foushee, H.C. and Helmreich, R.L. Group interaction and flight crew performance. *Human Factors in Aviation.* Academic Press, San Diego, CA, 1988.
4. Horvitz, E. Principles of mixed-initiative user interfaces. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems.* (Pittsburgh, PA, May 1999).
5. Miller, C.A. Definitions and dimensions of etiquette. Working notes of the AAAI Fall Symposium on Etiquette for Human-Computer Work (2002). Technical Report FS-02-02. AAAI, Menlo Park, CA, 1–7.
6. Miller, C.A. and Funk, H. Associates with etiquette: Meta-communication to make human-automation interaction more natural, productive and polite. In *Proceedings of the 8th European Conference on Cognitive Science Approaches to Process Control.* (Munich, Sept. 24–26, 2001), 329–338.
7. Miller, C.A. and Hannen, M. The Rotorcraft Pilot's Associate: Design and evaluation of an intelligent user interface for a Cockpit Information Manager. *Knowledge Based Systems 12* (1999), 443–456.
8. Pickering, J. Agents and artefacts. *Social Analysis 41,* 1 (1997), 45–62.
9. Reeves, B. and Nass, C. *The Media Equation.* Reeves, B. & Nass, C. The Media Equation: *How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge University Press/CSLI, NY, 1996.

Chistopher A. Miller (cmiller@sift.info) is chief scientist at Smart Information Flow Technologies (SIFT) Inc., Minneapolis, MN.

Etiquette Equality: EXHIBITIONS *and* EXPECTATIONS *of* COMPUTER POLITENESS

As much as people crave more human-like computers, experimental evidence shows that people do display polite behavior toward computers, perceiving human qualities they don't actually have.

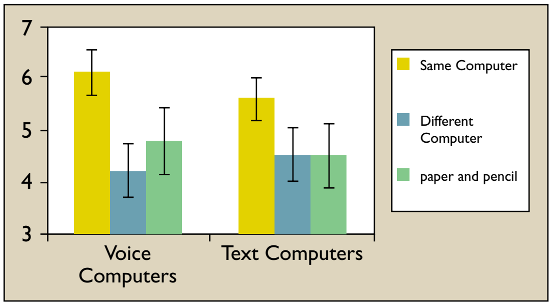
By CLIFFORD NASS

When my son Matthew was two years old, I told him, "It's not polite to eat so fast." He asked, "How come you don't yell at Honey (our dog)? She eats much faster than me." "Dogs don't have to be polite," I answered. Matthew replied, "Then why do you tell Honey it's bad when she barks at someone?"

Later that day, Matthew and I were walking along the street when he pointed and said, "Look at that ugly dog." I told him, "You shouldn't say things like that." He said, "Do I hurt a dog's feelings when I say bad things about it?" Once again, a teachable moment presented itself; like many parents, I resorted to, "Let's get some ice cream and eat it slowly."

The difficulty of explaining why people are polite to dogs opened up an opportunity for a fundamental insight into human-computer interaction: If people are polite to dogs, even when such behavior is inexplicable, might people also be polite to computers?

One of the most fundamental rules of politeness is that when one asks a question about oneself after stating one's own opinion, the respondent is supposed to agree [9]. Thus, the answer to the question, "I think I'm a great teacher; what do you think?" the polite response is, "you're great." If the question is asked by a neutral third party, one can give an honest answer rather than be polite. This leads us to the rather odd-sounding conclusion: If people apply the same politeness rules to computers that they do to other people, a computer that asked about itself would receive more positive evaluations than a different computer or paper-and-pencil iinquirey about the original computer's performance.



Mean assessment of the initial computer.

Inspired by this question, we developed an experimental paradigm (for complete details, see [8]) in which participants performed three tasks with a voice-based computer (a black-and-white NeXT computer with no text on the screen other than radio buttons at the bottom of the screen for the user's responses): They were tutored; they took a test; and they received (from the computer) an evaluation of their own performance and a (positive) evaluation of the computer's own performance.[1] We selected expert computer users to ensure the participants didn't believe it was reasonable to be polite to a computer.

After the evaluation session, individuals answered 21 questions concerning the computer's performance. One-third of the participants were presented with the textual questionnaire on the computer they had just worked with; one-third answered the identical questions on a computer on the other side of the room; and the final one-third answered the questionnaire with paper and pencil. Each adjective was presented with a 10-point scale ranging from "Describes Very Poorly" to "Describes Very Well."

The results of the experiment are presented in the figure here. Stunningly, participants behaved toward the computer in the same way they do toward other people. That is, they gave significantly more positive responses than when they were asked the identical questions in the identical order by either a computer on the other side of the room or via paper and pencil; that is, they were polite to the initial computer. A follow-up experiment demonstrated the same pattern held when people used a less human-like, text-based computer(s).

These studies initiated the idea that people were polite to computers, and they have become the canonical examples (for example, [3, 9, 11]) of the Computers Are Social Actors paradigm [7, 9], namely, people treat computers and expect computers to behave using the same rules and heuristics as they do when interacting with other people.

These responses are different in two ways from the well-known seeming social responses such as pleading with a car to start or screaming to a quarterback to throw the ball. First, in the latter behaviors, individuals are well aware they are behaving socially, while in our experiment(s), people adamantly deny a social response. Second, the responses in our experiments were not spontaneous and short-lived; individuals had as much time as they wished to make attributions.

Since these studies, our lab and others around the world have demonstrated that people extend a wide range of rules of etiquette to computers. People will reciprocate to a computer: In the U.S., an individualistic culture, users will help a computer that helps them more than if a different computer asks for the favor [1]. Conversely, in a collectivist culture like Japan, people will politely reciprocate to the second computer if it is the same *brand* as the first, but not a different brand [4]. People also prefer computers that are polite: A telephone-based system that blames the user for misrecognitions is viewed more negatively and is less effective than one that politely blames external circumstances, such as noise on the line. Flattery is also very effective: People respond to it as positively as praise known to be sincere [2]. In all of these experiments, people insisted they were not applying the rules of etiquette, even though it became clear they were.

Why Are People Polite to Non-Humans?

Why don't people hold dogs and computers to the same standards of polite behavior as people? The obvious answers seemed insufficient. "Dogs/computers

[1] For example, "You answered question 3 correctly. This computer (criterion 1) provided very useful information," or "You answered question 5 incorrectly. However, this computer provided useful facts for answering this question." The "self-praise" comments were employed to give participants the sense that the computer perceived its performance positively, thereby increasing the feeling the person, if they were to be polite, would respond positively in return. The computer did not say "I" or attribute emotion to itself in order to eliminate the possibility that participants believed that we intended them to anthropomorphize.

aren't expected to [or smart enough to] be polite" can be challenged by various situations in which polite behavior is demanded and learned.

Why is it that it's inappropriate to make negative comments to an animal and computer that doesn't understand what is being said? "The dog/computer might feel bad" attaches a level of intellectual sophistication that is clearly unwarranted. "It is good practice for when you talk with people" or "it's a habit" lead to questions of why we don't feel obligated to thank a rock for being particularly strong. "The owner/programmer might hear" is belied by the fact that people tend not to criticize dogs (except for strays) even when the owner isn't present. In essence, both adults and children expect that dogs and computers will be polite and should be treated politely.

Unfortunately, direct attempts to understand politeness toward dogs and computers are doomed to failure. "Do you expect dogs/computers to be polite?" or "Do you try to be polite to dogs/computers?" are questions that suggest the person should anthropomorphize before answering, and even if we could overcome that obstacle (which seems highly unlikely), it would be extremely difficult to explain what politeness would mean in a non-human context. Furthermore, the link between what people say they do and what they actually do is quite weak [5].

What makes people adopt an inclusive approach to polite responses and polite expectations, even when they know that those responses are illogical? In his introduction, Chris Miller suggests the underlying process is "agentification." Why is agentification so broadly assigned and why does that assignment lead to politeness? One possible explanation comes from evolutionary psychology.

Humans evolved in a world in which their most significant opportunities and problems, from food to shelter to physical harm, all revolved around other people. In this environment, there would be a significant evolutionary advantage to the rule: If there's even a low probability that it's human-like, assume it's human-like. This explains why the right ear (which corresponds to the left-side of the brain) shows a clear advantage in processing not only native language but also nonsense syllables, speech in foreign languages, and even speech played backward. The left ear attends to all other sounds [10]: If it's close to speech, humans assume it's speech (that is, a person speaking). It also explains why people see human faces in cars (for example, a Volkswagen Beetle starred in a number of movies), sock puppets, and even a simple circle with two dots in the upper half and a curved line at the bottom half (the "Smiley") [6].

What, then, are the cues that encourage people to treat a computer (or anything else) as a social actor that warrants and is expected to exhibit human speech? We hypothesize that at least each of the following triggers etiquette responses, with entities having more of these likely eliciting both broader and more robust polite responses and polite expectations:

- Language use (the basis of the Turing Test);
- Voice (including synthetic voices);
- Face (including dog-like faces);
- Emotion manifestation;
- Interactivity, especially rich contingency established over time;
- Engagement with and attention to the user (at least perceived engagement);
- Autonomy/unpredictability; and
- The filling of roles traditionally filled by humans (for example, teacher, telephone operator)

While humans encompass all of these elements, computers, pictorial agents, interactive voice response systems (which elicit a number of "please" and "thank you" responses), robot (and live) dogs, and many other technologies exhibit some of these characteristics and seem to elicit some social responses.

Pessimists examine the expansiveness of etiquette and frown at another faulty heuristic that demonstrates the limited cognitive processing skills of people. I take a different view: Polite responses to computers represent the best impulse of people, the impulse to err on the side of kindness and humanity. C

References

1. Fogg, B.J. *Persuasive Technology: Using Computers to Change What We Think and Do.* Morgan Kauffmann, San Mateo, CA, 2002.
2. Fogg, B.J. and Nass, C. Silicon sycophants: Effects of computers that flatter. *Intern. J. of Human-Computer Studies 46,* 5 (1997), 551–561.
3. Gates, B. *The Road Ahead*. Viking, New York, 1995.
4. Katagiri, Y., Nass, C., and Takeuchi, Y. Cross-cultural studies of the computers are social actors paradigm: The case of reciprocity. *Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents, and Virtual Reality.* M.J. Smith, G. Salvendy, D. Harris, and R. Koubek, Eds. Lawrence Erlbaum, Mahwah, NJ, 2001, 1558–1562.
5. LaPiere, R. Attitudes and actions. *Social Forces 13* (1934), 230–237.
6. McCloud, S. *Understanding Comics—The Invisible Art.* Kitchen Sink Press, Northampton, MA, 1993, 24–33.
7. Nass, C. and Moon, Y. Machines and mindlessness: Social responses to computers. *J. of Social Issues 56,* 1 (2000), 81–103.
8. Nass, C., Moon, Y., and Carney, P. Are respondents polite to computers? Social desirability and direct responses to computers. *J. of Applied Social Psychology 29,* 5 (1999), 1093–1110.
9. Reeves, B. and Nass, C. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge University Press/ICSLI, NY, 1996.
10. Slobin, D.I. *Psycholinguistics.* Foresman and Co., Glenview, IL, 1979.
11. Turkle, S. *Life on the Screen*. Simon & Schuster, New York, 1995.

Clifford Nass (nass@stanford.edu) is a professor in the Department of Communication, Stanford University, Stanford, CA.

UNSPOKEN RULES *of* SPOKEN INTERACTION

Body language and familiar silent signals are as much a part of social experience as the conversation. Building systems to recognize and respond to such moves will propel interface technology to the next horizon.

By Timothy W. Bickmore

Our face-to-face interactions with other people are governed by a complex set of rules, of which we are mostly unaware. For decades now, social scientists have been unraveling the threads of face-to-face interaction, investigating everything from descriptions of body posture used to indicate interest in starting a conversation, to eye gaze dynamics used to convey liking or disliking, to the myriad ways that language can convey attitude, social status, relationship status, and affective state. Even though we are not always aware of them, these rules underpin how we make sense of and navigate in our social world. These rules may seem uninteresting and irrelevant to many computer scientists, but to the extent that a given interaction rule is universally followed within a user population, it can be profitably incorporated into a human-machine interface in order to make the interface more natural and intuitive to use. Computers without anthropomorphic faces and bodies can (and already do) make use of a limited range of such rules—such as rules for conversational turn-taking in existing interfaces—but one kind of interface has the potential to make explicit, maximal use of these rules: embodied conversational agents (ECAs).

ECAs are animated humanoid computer characters that emulate face-to-face conversation through the use of hand gestures, facial display, head motion, gaze behavior, body posture, and speech intonation, in addition to speech content [5]. The use of verbal and nonverbal modalities gives ECAs the potential to fully employ the rules of etiquette observed in human face-to-face interaction. ECAs have been developed for research purposes, but there are also a growing number of commercial ECAs, such as those developed by Extempo, Artificial Life, and the Ananova newscaster. These systems vary greatly in their linguistic capabilities, input modalities (most are mouse/text/speech input only), and task domains, but all share the common feature of attempting to engage the user in natural, full-bodied (in some sense) conversation.

Social scientists have long recognized the utility of making a distinction between conversational behaviors (surface form, such as head nodding) and conversational function (the role played by the behavior, such as acknowledgement). This distinction is important if general rules of interaction are to be induced that capture the underlying regularities in conversation, enabling us to build ECA architectures that have manageable complexity, and that have the potential of working across languages and cultures. This distinction is particularly important given that there is usually a many-to-many mapping between functions and behaviors (for example, head nodding can also be used for emphasis and acknowledgment can also be indicated verbally).

Although classical linguistics have traditionally focused on the conveying of propositional information, there are actually many different kinds of conversational function. The following list reviews some of the functions most commonly implemented in ECAs and examines their range of conversational functions and associated behaviors:

Propositional functions of conversational behavior involve representing a thought to be conveyed to a listener. In addition to the role played by speech, hand gestures are used extensively to convey propositional information either redundant with, or complementary to, the information delivered in speech. In ECA systems developed to date,

LISA HANEY

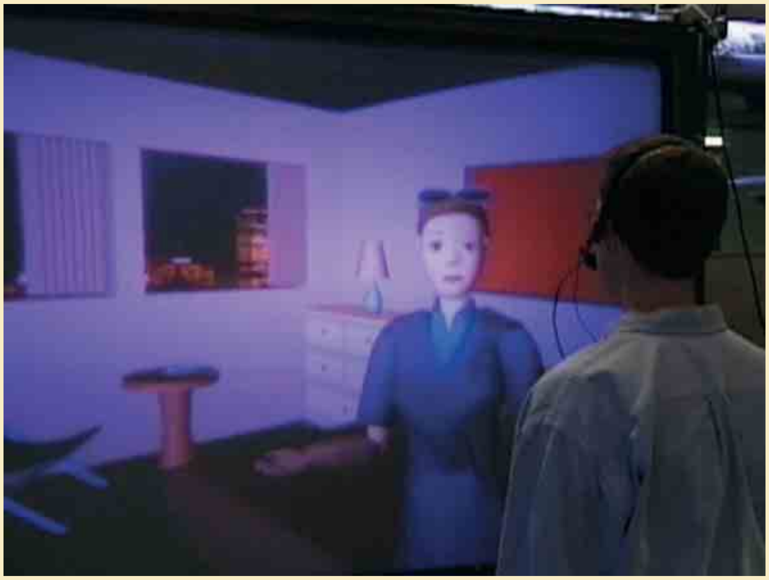
REA the Polite Real Estate Agent

REA is a virtual real estate agent who conducts initial interviews with potential home buyers, then shows them virtual houses she has for sale [4]. In these interviews—based on studies of human real estate agent dialogue—REA is capable of using a variable level of etiquette, which in turn conveys varying levels of sensitivity to users' "face needs" (needs for acceptance and autonomy). If the etiquette gain is turned up, she starts the conversation with small talk, gradually eases into the real estate conversation, and sequences to more threatening topics, like finance, toward the end of the interview. If the etiquette gain is turned down, her conversational moves are entirely driven by task goals, resulting in her asking the most important questions first (location and finance) and not conducting any small talk whatsoever. The amount of etiquette required at any given moment is dynamically updated each speaking turn of the conversation based on an assessment of the relationship between REA and the user, and how it changes as different topics are discussed.

REA's dialogue planner is based on an activation network that integrates information from the following sources to choose her next conversational move:

Task goals. REA has a list of prioritized goals to discover the user's housing needs in the initial interview. Conversational moves that directly work toward satisfying these goals (such as asking interview questions) are preferred (given activation energy).

Logical preconditions. Conversational moves have logical preconditions (for example, it makes no sense for REA to ask users how many bedrooms they want until she has established they are interested in buying a house), and are not selected for execution until all of their preconditions are satisfied. Activation energy flows through the network to prefer moves able to be executed ("forward chaining") or that support (directly or indirectly) REA's task goals ("backward chaining").



REA interviewing a buyer.

Face threat. Moves expected to cause face threats to the user, including threats due to overly invasive topics (like finance) are not preferred.

Face threat avoidance. Conversational moves that advance the user-agent relationship in order to achieve task goals that would otherwise be threatening (for example, small talk and conversational storytelling to build trust) are preferred.

Topic coherence. Conversational moves that are somehow linked to topics currently under discussion are preferred.

Relevance. Moves that involve topics known to be relevant to the user are preferred.

Topic enablement. REA can plan to execute a sequence of moves that gradually transition the topic from its current state to one that REA wants to talk about (for example, from talk about the weather, to talk about Boston weather, to talk about Boston real estate). Thus, energy is propagated from moves whose topics are not currently active to moves whose topics would cause them to become current. **C**

the most common kind of hand gesture implemented is the deictic, or pointing gesture. Steve [10], the DFKI Persona [1], and pedagogical agents developed by Lester et al. [7], use pointing gestures that can reference objects in the agent's immediate (virtual or real) environment.

Interactional functions are those that serve to regulate some aspect of the flow of conversation (also called "envelope" functions). Examples include turn-taking functions, such as signaling intent to take or give up a speaking turn, and conversation initiation and termination functions, such as greetings and farewells (used in REA, see pevious page). Other examples are "engagement" functions, which serve to continually verify that one's conversational partner is still engaged in and attending to the conversation, as implemented in the MEL robotic ECA [11]. Framing functions (enacted through behaviors called "contextualization cues") serve to signal changes in the kind of interaction taking place, such as problem-solving talk versus small talk versus joke-telling, and are used in the FitTrack Laura ECA (see "Managing Long-Term Relationships with Laura.")

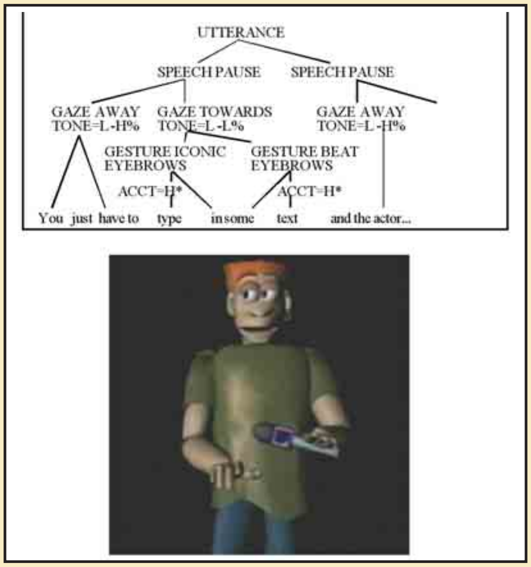
Attitudinal functions signal liking, disliking, or

Automatic Generation of Nonverbal Behavior in BEAT

ALTHOUGH the nonverbal behavior exhibited by an ECA can play a significant role in enacting rules of etiquette, the correct production of these behaviors can be a very complex undertaking. Not only must the form of each behavior be correct, but the timing of the behavior's occurrence relative to speech must be precise if the behavior is to have the intended effect on the user.

The BEAT system simplifies this task, by taking the text to be spoken by an animated human figure as input, and outputting appropriate and synchronized nonverbal behaviors and synthesized speech in a form that can be sent to a number of different animation systems [6]. The nonverbal behaviors are assigned on the basis of linguistic and contextual analysis of the text, relying on rules derived from research into human conversational behavior. BEAT can currently generate hand gestures, gaze behavior, eyebrow raises, head nods, and body posture shifts, as well as intonation commands for a text-to-speech synthesizer.

The BEAT system was designed to be modular, to operate in real time, and to be easily extensible. To this end, it is written in Java, is based on an input-to-output pipeline approach with support for user-defined extensions, and uses XML as its primary data structure. Processing is decomposed into modules that operate as XML transducers; each taking an XML object tree as input and producing a modified XML tree as output. The first module in the pipeline operates by reading in XML-tagged text representing the character's script and converting it into a parse tree. Subsequent modules augment this XML tree with suggestions for appropriate nonverbal behavior while filtering out suggestions in conflict or that do not meet specified criteria. The figure here shows an example XML tree at this stage of processing, with annotations for speech intonation (SPEECH-PAUSE, TONE, and ACCENT tags), gaze behavior (GAZE-AWAY and GAZE-TOWARDS, relative to the user), eyebrow raises (EYEBROWS), and hand gestures (GESTURE). In the final stage of processing, the tree is converted into a sequence of animation instructions and synchronized with the character's speech by querying the speech synthesizer for timing information.



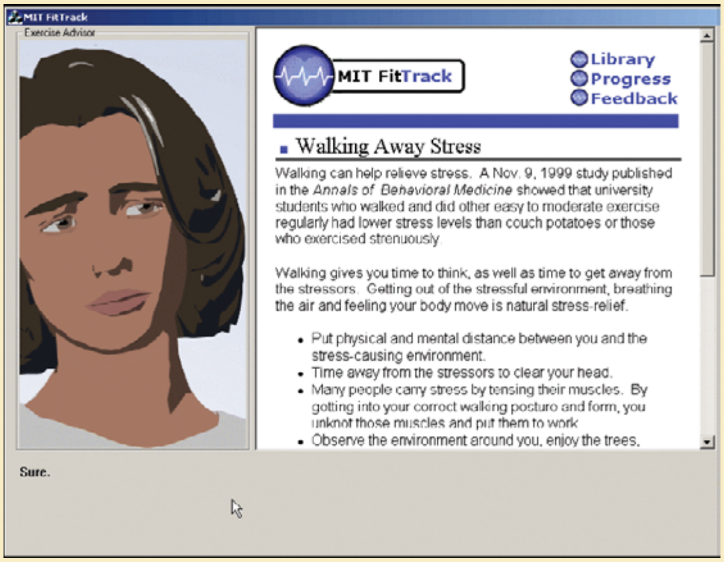
BEAT annotated parse tree and its performance.

BEAT provides a very flexible architecture for the generation of nonverbal conversational behavior, and is in use on a number of different projects at different research centers, including the FitTrack system (see "Managing Long-Term Relationships with Laura"). **C**

Managing Long-Term Relationships with Laura

THE effective establishment and maintenance of relationships requires the use of many subtle rules of etiquette that change over time as the nature of the relationship changes. The FitTrack system was developed to investigate the ability of ECAs to establish and maintain long-term, social-emotional relationships with users, and to determine if these relationships could be used to increase the efficacy of health behavior change programs delivered by the agent [3]. The system was designed to increase physical activity in sedentary users through the use of conventional health behavior change techniques combined with daily conversations with Laura, a virtual, embodied exercise advisor.

Laura's appearance and nonverbal behavior were based on a review of the health communication literature and a series of pretest surveys (see figure). BEAT (see "Automatic Generation of Nonverbal Behavior in BEAT") was used to generate nonverbal behavior for Laura, and was extended to generate different baseline nonverbal behaviors for high or low immediacy (liking or disliking of one's conversational participant demonstrated through nonverbal behaviors such as proximity and gaze) and different conversational frames (health dialogue, social dialogue, empathetic dialogue, and motivational dialogue). In addition to the nonverbal immediacy behaviors, verbal relationship-building strategies used by Laura include empathy dialogue, social dialogue, meta-relational communication (talk about the relationship), humor, reference to past interactions and future together, inclusive pronouns, expressing happiness to see the user, use of close forms of address (user's name), and appropriate politeness strategies.



Walking Away Stress

Walking can help relieve stress. A Nov. 9, 1999 study published in the *Annals of Behavioral Medicine* showed that university students who walked and did other easy to moderate exercise regularly had lower stress levels than couch potatoes or those who exercised strenuously.

Walking gives you time to think, as well as time to get away from the stressors. Getting out of the stressful environment, breathing the air and feeling your body move is natural stress-relief.

- Put physical and mental distance between you and the stress-causing environment.
- Time away from the stressors to clear your head.
- Many people carry stress by tensing their muscles. By getting into your correct walking posture and form, you unknot those muscles and put them to work.
- Observe the environment around you, enjoy the trees.

Sure.

Laura and the MIT FitTrack system.

The exercise-related portion of the daily dialogues Laura has with users was based on a review of the health behavior change literature, input from a cognitive-behavioral therapist, and observational studies of interactions between exercise trainers and MIT students. These interventions were coupled with goal-setting and self-monitoring, whereby users would enter daily pedometer readings and estimates of time in physical activity, and were then provided with graphs plotting their progress over time relative to their goals.

In a randomized trial of the FitTrack system, 60 users interacted daily with Laura for a month on their home computers, with one group interacting with the fully "relational" Laura, and the other interacting with an identical agent that had all relationship-building behaviors disabled. Users who interacted with the relational Laura reported significantly higher scores on measures of relationship quality, liking Laura, and desire to continue working with Laura, compared with users in the non-relational group, although no significant effects of relational behavior on exercise were found. Most users seemed to enjoy the relational aspects of the interaction (though there were definitely exceptions). As one user put it: "I like talking to Laura, especially those little conversations about school, weather, interests. She's very caring. Toward the end, I found myself looking forward to these fresh chats that pop up every now and then. They make Laura so much more like a real person." **C**

Etiquette **rules often serve as coordination devices and can be seen as enacting an interactional function.**

other attitudes directed toward one's conversational partner (as one researcher put it, "you can barely utter a word without indicating how you feel about the other"). One of the most consistent findings in this area is that the use of nonverbal immediacy behaviors—close conversational distance, direct body and facial orientation, forward lean, increased and direct gaze, smiling, pleasant facial expressions and facial animation in general, head nodding, frequent gesturing, and postural openness—projects liking for the other and engagement in the interaction, and is correlated with increased solidarity [2]. Attitudinal functions were built into the FitTrack ECA so it could signal liking when attempting to establish and maintain working relationships with users, and into the Cosmo pedagogical agent to express admiration or disappointment when students experienced success or difficulties [7].

Affective display functions. In addition to communicating attitudes about their conversational partners, people also communicate their overall affective state to each other using a wide range of verbal and nonverbal behaviors. Although researchers have widely differing opinions about the function of affective display in conversation, it seems clear it is the result of both spontaneous readouts of internal state and deliberate communicative action. Most ECA work in implementing affective display functions has focused on the use of facial display, such as the work by Poggi and Pelachaud [8].

Relational functions are those that either indicate a speaker's current assessment of his or her social relationship to the listener ("social deixis"), or serve to move an existing relationship along a desired trajectory (for example, increasing trust, decreasing intimacy, among others). Explicit management of the ECA-user relationship is important in applications in which the purpose of the ECA is to help the user undergo a significant change in behavior or cognitive or emotional state, such as in learning, psychotherapy, or health behavior change [3]. Both REA and Laura were developed to explore the implementation and utility of relational functions in ECA interactions.

While it is easiest to think of the occurrence (versus non-occurrence) of a conversational behavior as achieving a given function, conversational functions are often achieved by the manner in which a given behavior is performed. For example, a gentle rhythmic gesture communicates a very different affective state or interpersonal attitude compared to a sharp exaggerated gesture. Further, while a given conversational behavior may be used primarily to affect a single function, it can usually be seen to achieve functions from several (if not all) of the categories listed here. A well-told conversational story can communicate information, transition a conversation into a new topic, convey liking and appreciation of the listener, explicate the speaker's current emotional state, and serve to increase trust between the speaker and listener.

The Rules of Etiquette

Within this framework, rules of etiquette can be seen as those conversational behaviors that fulfill certain conversational functions. Emily Post would have us believe the primary purpose of etiquette is the explicit signaling of "consideration for the other"—that one's conversational partner is important and valued [9]—indicating these behaviors enact a certain type of attitudinal function. Etiquette rules often serve as coordination devices (for example, ceremonial protocols) and can be seen as enacting an interactional function. They can also be used to explicitly signal group membership or to indicate a desire to move a relationship in a given direction, in which case they are fulfilling a relational function. Each of these functions has been (partially) explored in existing ECA systems.

Is etiquette—especially as enacted in nonverbal behavior—important in all kinds of human-computer interactions? Probably not. However, for tasks more fundamentally social in nature, the rules of etiquette and the affordances of nonverbal behavior can certainly have an impact. Several studies of mediated human-human interaction have found that the additional nonverbal cues provided by video-mediated communication do not affect performance in task-ori-

***Will* users willingly engage in a social chat with an animated real estate agent or tell their troubles to a virtual coach? Evidence to date indicates the answer is yes.**

ented interactions, but in interactions of a more relational nature, such as getting acquainted, video is superior [12]. These studies have found that for social tasks, interactions were more personalized, less argumentative, and more polite when conducted via video-mediated communication, that participants believed video-mediated (and face-to-face) communication was superior, and that groups conversing using video-mediated communication tended to like each other more, compared to audio-only interactions. The importance of nonverbal behavior is also supported by the intuition of business people who still conduct important meetings face-to-face rather than on the phone. It would seem that when a user is performing these kinds of social tasks with a computer, an ECA would have a distinct advantage over nonembodied interfaces.

Conclusion

Will users willingly engage in a social chat with an animated real estate agent or tell their troubles to a virtual coach? Evidence to date indicates the answer is, for the most part, yes. In the commercial arena, people have shown willingness to engage artifacts such as Tamagotchis, Furbies, and robotic baby dolls in ever more sophisticated and encompassing social interactions. Experience in the laboratory also indicates users will not only readily engage in a wide range of social behavior appropriate to the task context, but that the computer's behavior will have the same effect on them as if they had been interacting with another person [3–5]. This trend seems to indicate a human readiness, or even need, to engage computational artifacts in deeper and more substantive social interactions.

Unfortunately, there is no cookbook defining all of the rules for human face-to-face interaction that human-computer interface practitioners can simply implement. However, many of the most fundamental rules have been codified in work by linguists, sociolinguists, and social psychologists (for example, [2]), and exploration that makes explicit use of these rules in work with ECAs and robotic interfaces has begun. By at least being cognizant of these rules, and at most by giving them explicit representation in system design, developers can build systems that are not only more natural, intuitive, and flexible to use, but result in better outcomes for many different kinds of tasks.

References

1. Andre, E., Muller, J. and Rist, T. The PPP persona: A multipurpose animated presentation agent. *Advanced Visual Interfaces,* (1996).
2. Argyle, M. *Bodily Communication*. Methuen, New York, 1988.
3. Bickmore, T. Relational agents: Effecting change through human-computer relationships. *Media Arts & Sciences,* MIT, Cambridge, MA, 2003.
4. Cassell, J. and Bickmore, T. Negotiated collusion: Modeling social language and its relationship effects in intellient agents. *User Modeling and Adaptive Interfaces 13* (1–2), 89–132.
5. Cassell, J., Sullivan, J., Prevost, S. and Churchill, E., Eds. *Embodied Conversational Agents.* The MIT Press, Cambridge, MA, 2000.
6. Cassell, J., Vilhj·lmsson, H. and Bickmore, T. BEAT: The Behavior Expression Animation Toolkit. In *Proceedings of ACM SIGGRAPH,* (Los Angeles, CA, 2001), 477–486.
7. Lester, J., Towns, S., Callaway, C., Voerman, J. and Fitzgerald, P. Deictic and emotive communication in animated pedagogical agents. Embodied Conversational Agents. J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds. MIT Press, Cambridge, MA, 2000.
8. Poggi, I. and Pelachaud, C. Performative facial expressions in animated faces. *Embodied Conversational Agents*. J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds. MIT Press, Cambridge, MA, 2000, 155–88.
9. Post, E. *Etiquette in Society, in Business, in Politics and at Home.* Funk and Wagnalls, New York, 1922.
10. Rickel, J. and Johnson, W.L. Animated agents for procedural training in virtual reality: Perception, cognition and motor control. *Applied Artificial Intelligence 13* (1999), 343–382.
11. Sidner, C., Lee, C. and Lesh, N. Engagement rules for human-computer collaborative interactions. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,* (2003), 3957–3962.
12. Whittaker, S. and O'Conaill, B. The role of vision in face-to-face and mediated communication. *Video-Mediated Communication.* K. Finn, A. Sellen, and S. Wilbur, Eds. Lawrence Erlbaum, 1997, 23–49.

Timothy W. Bickmore (bickmore@bu.edu) is an assistant professor in the Medical Information Systems Unit of Boston University School of Medicine, Boston, MA.

ETIQUETTE *and the* DESIGN *of* EDUCATIONAL TECHNOLOGY

Motives for designing tact, courtesy, and even humor into educational tools must be intrinsically understood and appreciated before applied.

By Punya Mishra and Kathryn A. Hershey

Educators have always emphasized good manners and etiquette, both in their own behavior and in attempting to inculcate good manners in their students. However, the etiquette of human-computer interaction (HCI) has not been of much concern to designers of educational technology, who typically consider computers as cognitively powerful but socially neutral tools. The presumed anti-social nature of computers was often argued as a key advantage for pedagogical purposes. Moreover, it has been argued that learners would find computer personalities implausible, scripted, stiff, unnatural, and lacking the vitality and dynamism of human personalities.

Recently, however, researchers and developers working on educational tools have been paying more attention to issues of etiquette. There appear to be two reasons for this change. First, evidence from the Computers As Social Actors (CASA) research paradigm [11] provided empirical evidence that, given the mere perception of agency, people often respond naturally to computers as they would to real people and events. This has been summarized as Topffer's law, which states, "All interfaces, however badly developed, have personality" [8]. This undermines the idea that computers are merely neutral tools and emphasizes the importance of the social relationship that can and will develop between a computer and a learner.

The second reason for an increased interest in etiquette has been recent work on interactive software agents. All software agents, (including Embodied Conversational Agents—ECAs—described by Timothy Bickmore in this section), utilize advances in natural language processing, affective computing, and multimodal interfaces to develop believable, anthropomorphic entities (see the sidebar "Pedagogical Software Agents"). The motivation to make use of these agents, with their explicit inclusion of social modalities, in pedagogical applications has forced researchers to grapple with issues of what is appropriate behavior on the part of a software agent.

Etiquette in Educational Technology

Educational technology is a broad field and incorporates many different kinds of interactions where etiquette can play a role: between students, between students and teachers and between the student and the computer. Our focus here is the interaction between student and computer, that is, the etiquette of educational HCI.

There are some fundamental differences between general HCI etiquette and etiquette for educational HCI. Etiquette considerations in educational technology are complicated by the fact that learning from a computer is not just about ease of use. Learning can be frustrating and difficult, particularly when it exposes learners' errors in thinking and gaps in knowledge and forces them to grapple with difficult subject matter. In an educational context, ease of use may be subservient to larger goals of learning content or subject matter, monitoring the learner's prior and developing knowledge, while maintaining a focus on issues of motivation and affect.

For example, consider receiving unsolicited help from a computer system. Making help systems useful and available on demand by inferring user needs has been an important goal of HCI researchers. However, in an educational context, help is not necessarily perceived as being good. Research has shown that students make complex attributions based on whether or not they receive help. For instance, students receiving unsolicited help may be perceived as less capable than students who did not [1]. A computer program offering help may be exhibiting generally appropriate HCI etiquette, but inappropriate educational etiquette.

Any discussion of etiquette in educational technology must contend with the wide variety of purposes for learning. One can learn in order to acquire factual knowledge; develop understanding; improve performance and skill; acquire strategies and techniques; improve self-esteem; enjoy and have fun; persuade people to act in certain ways; and inspire learners and motivate them to action.

Learning also plays out in widely different contexts, which constrain or determine the roles played by the computer and the learner. To take an extreme example, we have very different expectations of a drill instructor than of an elementary school teacher. These contexts (and the concomitant roles associated with them) dictate the range of behaviors considered appropriate as well as communicate shared goals and expectations to both parties. Crucial pedagogical issues such as who gets to direct conversation topics, who gets to raise questions and when, are often implicitly embedded within these contexts of activity. Since etiquette is highly context-dependent, what may be appropriate educational etiquette in one situation may be inappropriate in another. Educational etiquette is dependent upon how these relationships are understood and instantiated in the design of the interface. The larger context within which education is situated can also make a difference in how etiquette is considered. Issues of culture, ethnicity, and gender can play a significant role in determining appropriate and inappropriate etiquette as well.

We presented a set of different goal/context/role patterns [7] within which educational technologies and humans may function (such as computers as tutors, as "tutees," as tools for productivity, as tools for exploration, and as tools for assessment). Clearly different learning technologies can straddle across two or more of these categories. For instance, a simulation of frog dissection could be used as a tutorial and as an arena for exploration. What is important to note here is that each of these contexts implicitly assumes a particular set of values and beliefs about teaching and learning and the roles of the learner and the computer. For instance a tutorial system, such as those used for Computer-Aided Instruction (CAI), is driven much more by goals of the tutorial system than the intentions of the student. An example of such a system is Cardiac Tutor [12] that helped students

Pedagogical Software Agents

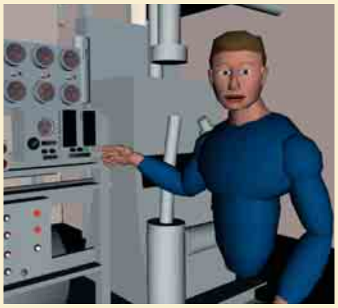
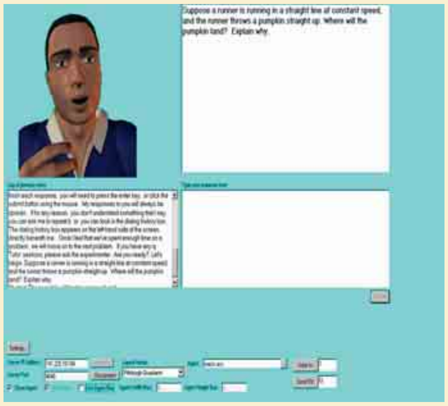


Figure 1. STEVE (Soar Training Expert for Virtual Environments) developed by the Center for Advanced Research in Technology for Education(CARTE). (Image © University of Southern California)

Pedagogical software agents are animated interface agents in instructional environments that draw upon human-human social communication scripts by embodying observable human characteristics (such as the use of gestures and facial expressions). Pedagogical software agents represent a new paradigm for teaching and learning based on research in the areas of animated interface agents and interactive learning environments [3]. As such, they represent ECAs applied specifically to the pedagogy task. Animated pedagogical agents broaden the bandwidth of tutorial communication and, it is hoped, increase student engagement and motivation. The social nature of interaction with such agents brings issues of etiquette to the forefront of research and design. There are many projects exploring the etiquette of learner-agent interaction.

The Center for Advanced Research in Technology for Education at the University of Southern California, under Lewis Johnson, is developing pedagogical software agents that exhibit various forms of etiquette such as politeness, expressiveness, and empathy for the purpose of promoting learning in interactive environments. An example of such an agent is STEVE (Soar Training Expert for Virtual Environments—see Figure 1). They have argued the effective use of such forms of etiquette requires agents to demonstrate a sense of "social intelligence." The Social Intelligence Project at CARTE is focused on such issues as tracking a student's cognitive and affective states, tracking learner-agent interaction as a social relationship, and managing interaction to improve communication effectiveness.

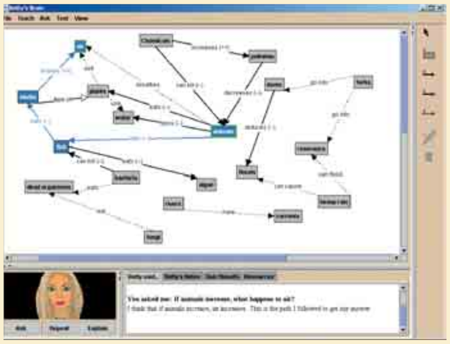
The Tutoring Research Group at the University of Memphis, led by Art Grasser, is developing a system called AutoTutor, which uses conversational agents that act as dialogue partners to assist learners in the active construction of knowledge. More importantly, they are developing agents that engage in a conversation using natural language (through Latent Semantic Analysis) to replicate the impression that the conversation is a cooperative effort, co-constructed by both participants, human and agent (see Figure 2).



Suppose a runner is running in a straight line at constant speed, and the runner throws a pumpkin straight up. Where will the pumpkin land? Explain why.

Figure 2. AutoTutor, developed by Tutoring Research Group, University of Memphis. (Image © University of Michigan)

The Teachable Agent Group at Vanderbilt University has developed social agents who play the role of the "tutee" rather than tutor. In this environment where students learn by teaching the agents, they are able to adjust the agent's attitude, and teach it relevant skills and concepts. The students are motivated by the prospect of teaching and learn from an iterative assessment-teaching cycle, which results from trying to assist the agent in solving problems anchored in rich contexts (see Figure 3).



You asked me: If animals increase, what happens to air?
I think that if animals increase, air increases. This is the path I followed to get my answer.

Figure 3. Betty's Brain, developed by the Teachable Agents Group at Vanderbilt University. (Image © Vanderbilt University)

Though the idea of psychological responses to social agents may be a relatively new one for the scientific research community, it has been around for a long time in the commercial development area—particularly in the area of children's educational software. Children may be quite vulnerable to demonstrating social responses to interactive media. Recently, more children's toys and software products have been deliberately designed to elicit social responses. From the animated dog in Jumpstart Toddlers to the "charming Professor P.T. Presto" on Microsoft's My Personal Tutor, anthropomorphic animated characters are designed to interest young minds and to allow for greater levels of engagement. However, most of this design has been proceeding on the basis of marketers' intuitions and without good theory to justify its successes and failures. C learn an established medical procedure through directed practice. Interaction in Cardiac Tutor is initiated by the tutor, and is controlled by it, providing feedback as and when needed. In contrast to this are open-ended exploratory systems such as complex simulations or games. These systems are driven more by the learner's interests than those of the software program. For instance, the game "Civilization" allows users to seek advice from expert "advisors" though it is up to the user whether or not to follow the advice. Developing smarter tools for learning requires getting a better understanding of these situated practices and implicit commitments, as well as the requirements, preferences, and background knowledge of the learner.

Integrating Etiquette in Educational Tools: Where Do We Begin?

The multiple goals/contexts/role patterns within which educational technology functions makes determining how to integrate principles of etiquette into such systems challenge. One simple rule of thumb (indeed, one which Reeves and Nass's 1996 CASA paradigm encourages us to follow) is to apply what has been found appropriate for human-human interaction (HHI) to the design of HCI. To understand how this could work we look at three different studies that attempt to apply HHI etiquette rules to HCI.

Personalized messages from agents and computer systems. We know that in most contexts personalizing conversations by addressing people by name is good etiquette. Not doing so makes the conversation stilted and formal (which, ironically, is true of most computer messages). Moreno and Mayer [9] conducted a study that looked at whether changing a pedagogical software agent's language style (personalized dialogue versus neutral monologue) would affect student learning. They found that students who learned by communicating with a pedagogic agent through personalized dialogue were able to recall more information and were better able to use what they have learned to solve problems, than students who learned via a neutral message. Clearly this is a case where a simple rule of HHI etiquette carried over to the HCI case as well.

Affective feedback from computer systems. Research has shown that praise and blame feedback from teachers can have complicated and paradoxical effects. Praise (and criticism) can be interpreted in many different ways and these interpretations (depending on the perceived difficulty of the task, innate sense of ability of the student, and their success and failure at completing the task relative to other students) can influence how the recipient responds to the feedback. Henderlong and Lepper [2] examined studies that show, for instance, that being praised for success in a task perceived as easy may have a negative effect on a student's self-confidence while being blamed for failing a task perceived as difficult may actually lead to a positive effect. The design of feedback in educational technology systems is often based on a simplistic (and erroneous) framework where praise is assumed to affect behavior positively irrespective of context. We conducted an experimental study [6] where participants received differential affective feedback (praise or blame) after success at an easy task or failure at a difficult task. We measured the effect of this feedback on the participants' motivation and self-perception of ability.

This study, framed within the CASA paradigm, replicated an HHI study [5] except that feedback was provided by computers (albeit via a simple textual interface) and not by humans. The results demonstrated students preferred praise from the computer and found it more motivating, irrespective of the difficulty of the task and their success at it. The fact that students accepted praise from the computer indicates they did at some level respond psychologically to it. However, the fact that their responses did not fully match the HHI experimental results indicates there are limits to what they considered appropriate or acceptable feedback from the computer (at least as it was presented in this study). We argue this may be because the participants did not engage in the same level of "deep psychological processing" about intentionality as they do with human respondents. Of course, one of the implications of ECA work is that a richer, more fully embodied agent might have altered these responses.

Humor and HCI. Humor plays a very important role in HHI—including teaching—as a way in which etiquette problems are resolved without conflict. Morkes, Kernal, and Nass [10] conducted a pair of experiments in which they looked at the effects of humor in a computer-mediated communication (CMC) task. In one case the participants were told they were interacting with another human being, in the other they were told they were interacting with a computer. In both cases the participants were provided preprogrammed comments. Set up as a direct test of the CASA hypothesis, the experiment found that though the results between the two groups were generally consistent, the participants in the HCI condition were less sociable, demonstrated less smiling and laughing behavior, felt less similar to their interaction partner, and spent less time on the task.

The results of the last two studies indicate there is validity to the CASA paradigm. For instance, participants in the study did respond to affective feedback from the computer or did smile at the humor exhibited by the computer. However, they also indicate the psychological aspects of HCI are complex and difficult to explain using simplistic frameworks such as "computers are neutral tools" or "interacting with computers is just the same as interacting with humans."

Clearly computers are not humans and the current state of technology does not allow us to be consistently fooled into thinking they are. But even if we could fool people into believing computers were sentient agents, it could be ethically problematic to do so. Indeed, there may be pragmatic reasons why computers should not become too social. For instance, certain characteristics of computer systems (such as consistency, adaptability, inability to take offense or be bored) can be pedagogically valuable. An emphasis on etiquette and enhancing sociability in our systems should not blind us to these advantages. Thus, by adhering to the CASA philosophy we run the risk of not only having the computer appear artificial and/or stupid, but of actually undermining the positive attributes that computers currently possess.

Conclusion

There has been a slow but growing realization on the part of the educational technology research community that designers of educational tools must go beyond the purely cognitive aspects of working with computers and factor in the social and psychological aspects as well. The design of appropriate etiquette in educational systems requires adding an additional layer to the issues of traditional interest to HCI researchers and developers. Etiquette is closely connected to contexts of activity and practice—to the goal/contexts/role patterns that structure interactions in a domain. A better understanding of these patterns [7] is essential to building tact and courtesy into our computer programs.

We also need to learn from existing research, particularly that of teacher behavior and its effect on student learning and motivation. For instance, a review of the literature on nonverbal behavior indicates that eye contact, gestures, vocal inflections, body movement, and combinations of nonverbal behaviors conveying enthusiasm, animation, and variation of stimuli can positively affect student motivation, attention, teacher ratings, immediate recall, and achievement [4]. This research can be of great utility in the design of appropriate behaviors for pedagogical agents. However, as the three studies described here suggest, we must be careful not to apply these findings to the HCI context indiscriminately. This strategy, though easy to follow, may not always be the most appropriate. Instead, pedagogical etiquette in educational software must be carefully crafted based on sound empirical research sensitive to the complexities of learning and human psychology. C

References

1. Graham, S., and Barker, G. The downside of help: An attributional-developmental analysis of helping behavior as a low ability cue. *J. Educational Psychology 82* (1990) 187–194.
2. Henderlong, J., and Lepper, M.R. The effects of praise on children's intrinsic motivation: A review and synthesis. *Psychological Bulletin 128* (2002), 774–795.
3. Johnson, W.L., Rickel, J.W., and Lester, J.C. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International J. AI in Education 11* (2000), 47–78.
4. Klingzing, H.G., and Tisher, T.P. Expressive nonverbal behaviors; A review of research on training with consequent recommendations for teacher education. *Advances in Teacher Education, Vol. 2.* J.D. Raths and L.G. Katz, Eds. Ablex Publishing, 89–133, 1986.
5. Meyer, W.U., Mittag, W., and Engler, U. Some effects of praise and blame on perceived ability and affect. *Social Cognition 4*, 3 (1986), 293–308.
6. Mishra, P. Affective feedback and its effect on perceived ability and affect: A test of the Computers as Social Actors Hypothesis. Submitted to the 2004 Annual Conference of the American Educational Research Assoc.
7. Mishra, P. and Hershey, K. A framework for designing etiquette for educational technology. In *Proceedings of the AAAI Fall 2002 Symposium on Etiquette in Human-Computer Work.* AAAI Press, Washington, DC.
8. Mishra, P., Nicholson, M., and Wojcikiewicz, S. Does my word processor have a personality? Topffer's law and educational technology. *J. Adolescent and Adult Literacy 44*, 7 (2001–2003), 634–641.
9. Moreno, R., and Mayer, R.E. Engaging students in active learning: The case for personalized multimedia messages. *J. Educational Psychology 92*, 4 (2000), 724–733.
10. Morkes, J., Kernal, H., and Nass, C. Effects of humor in task-oriented human-computer interaction and computer-mediated communication: A direct test of SRCT theory. *Human-Computer Interaction 14*, 4 (1999), 395–435.
11. Reeves, B., and Nass, C. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge University Press/CSLI, NY, 1996
12. Woolf, B.P., Beck. J., Eliot, C., and Stern, M. Growth and maturity of intelligent tutoring systems: A status report. *Smart Machines in Education: The Coming Revolution in Educational Technology.* K.D. Forbus and P.J. Feltorich, Eds. AAAI/MIT Press, Metro Park, CA, 2001.

Punya Mishra (punya@msu.edu) is an assistant professor in the Learning, Technology, and Culture Program, College of Education, Michigan State University, East Lansing, MI.

Kathryn A. Hershey (hersheyk@msu.edu) is a doctoral student in the Learning, Technology, and Culture Program, College of Education, and a research associate of MIND Lab, at Michigan State University, East Lansing, MI.

TRUST *and* ETIQUETTE *in* HIGH-CRITICALITY AUTOMATED SYSTEMS

By Raja Parasuraman and Christopher A. Miller

Whereas the other articles in this section discuss human-computer etiquette in traditional social interactions involving the use of computers that explicitly strive to elicit a perception of "personhood" from the human participant, we focus on computers that occupy more traditional roles as complex and largely unpersonified machines involved in high-criticality working relationships with humans—where the consequences of failure can be catastrophic in terms of lives, money, or both.

Politeness and social niceties are important in many human-human social interactions, but in critical, highly technical work, there is the common misperception that we can "dispense with protocol" and get down to business, even with those who are not particularly courteous. In fact, excessive adherence to polite norms can seem stilted and sometimes frustrating in such settings. Here, we argue the etiquette exhibited[1] by non-personified machines (that is, ones largely without human methods of expressiveness

[1] It might, perhaps, be more accurate to say the etiquette is perceived by human users rather than exhibited by the automation itself, but that subtlety is largely irrelevant to the work we review here.

such as facial expressions, speech, voice tones, and gestures) and computer-based automation can profoundly affect users' perception and correct, optimal usage of them in work domains such as aviation, manufacturing, and power plant operation, among others.

A primary candidate for a mechanism through which machine etiquette can influence usage is user trust [2, 6]. Trust influences when and whether users decide to use automation, and can therefore seriously affect the overall human + machine system reliability, efficiency, and safety.

The latter half of the 20th century witnessed explosive growth in computational power, speed, and intelligence. Engineers and designers flocked to the new information technology initially to harness it for efficiency, productivity, and safety in complex and high-criticality domains. Computers were used as faceless (if you discount blinking lights and spinning tape reels), mechanical boxes that calculated ballistic trajectories, controlled an aircraft's flight, or regulated a chemical process, long before they could interact socially in anything like natural language, much less gesture, tone of voice, and facial expression.

There is little doubt that such automation has markedly improved efficiency and productivity in many high-criticality systems. For example, the accident rate in modern automated aircraft is significantly lower than for previous generations. But widespread introduction of automation has not been entirely benign. Indeed, automation has been associated with unforeseen errors and has even been shown to cause catastrophic accidents [8].

Given this record, it is not surprising that automation has been viewed as a double-edged sword [1]. The growth of automation has motivated repeated questions about how human-automation systems should be designed to promote optimal overall system reliability, safety, and efficiency [12] and research has examined a number of factors that influence the effective use of automation. One important, intensively studied factor is trust: that is, users' willingness to believe information from a system or make use of its capabilities.

User Trust in Automation

For at least 30 years, researchers have examined the causes and effects of trust in automation (see [6] and [10] for extensive reviews). Trust is important because operators may not use a well-designed, reliable system if they believe it untrustworthy. Conversely, they may continue to rely on automation even when it malfunctions and may not monitor it effectively if they have unwarranted trust in it. In practice, user trust should be calibrated to the system and context of its use; users should have *accurate* beliefs about the reliability of automation (and about their own capabilities).

Several factors are known to influence this trust calibration process. The user's experience of the system's reliability and accuracy is clearly important. Intuitively, operator trust should increase as experiences of correct automation behaviors increase. This has been found in studies in which users rated their trust in automation after use [6, 9]. The influence of automation reliability is typically moderated by other factors, however. For example, a user's perception of his or her own ability will be judged against the perceived reliability of the automation, and the user will typically opt for the method believed to be best [5]. Other factors have also proved important, such as the risk associated with the decision and whether a given situation matches previous automation verification experiences, operator workload, and the time criticality of the situation [10].

All of these factors, however, imply an empirical and largely analytical model by which users tune their trust in an automated system. It is also clear that as automation (and the computers that support it) increases in complexity and capabilities, there is a decline in ability and willingness to experience a range of behaviors in context and/or to think deeply about and learn the detailed mechanisms by which automation behaviors are produced. Lee and See [6], in reviewing studies from the sociological literature, argue that trust between humans can also be influenced by shortcuts to the empirical and analytical methods described here. These shortcuts involve analogical or affective responses—that is, at least in human-human relations, we tend to trust those who behave similar to people we find trustworthy and/or those with whom we enjoy interacting. We tend to trust that person more than interacting with someone who produces a negative effect (see [2] for a review of models of the role of social language in producing trust and positive effects).

The Role of Etiquette in Calibrating Trust

We define etiquette as noted in the introduction to this section as "the set of prescribed and proscribed behaviors that permits meaning and intent to be ascribed to actions." Insofar as etiquette encodes the set of behaviors that mark individuals as members of trustworthy groups or behaviors as pleasing according to cultural norms, adherence to that etiquette can influence trust via the analogic or affective mechanisms that Lee and See [6] described in human-human interactions.

But does this link hold for human-machine interactions? Reeves and Nass [11] illustrated that people often respond socially to computer technology in ways that are similar to social interaction with humans. As one example, individuals are typically most attracted to others who appear to have personalities similar to them.

This phenomenon, called the *social attraction hypothesis* by psychologists, also predicts user acceptance of computer software [7].

Thus, we might expect aspects of etiquette that moderate the quality of human-human interaction to influence human relations to and use of technology, including automation. This might be especially true in systems where complexity makes it difficult to understand exactly how they will operate in all circumstances and, therefore, makes it attractive to rely on other cues (such as affect, group membership, certification, among others) in determining a level of trust.

Here, we explore evidence that the norms of human-human etiquette can and do affect the calibration of human trust in, and usage of, non-personified automation. First, we examine two instances of miscalibrated trust in high criticality automation and offer an etiquette-based explanation for why that miscalibration occurred. Then we present the results of a preliminary experiment where we explicitly manipulate one dimension of human-machine etiquette in order to examine its effects on user trust and usage decisions.

Etiquette and the Miscalibration of Human-Automation Trust

The consequences of inappropriate calibration of user trust can be catastrophic. An example from the maritime industry illustrates the effects of excessive trust. The cruise ship Royal Majesty ran aground off Nantucket after veering several miles off course toward shallow waters. Fortunately there were no injuries or fatalities, but losses totaled $2 million in structural damage and $5 million in lost revenue. The automated systems in this ship included an autopilot and an Automatic Radar Plotting Aid (ARPA) tied to signals received by a Global Positioning System (GPS). The autopilot normally used GPS signals to keep the ship on course, but GPS signals were lost when the cable from the antenna frayed. The autopilot then switched to dead-reckoning mode, no longer correcting for winds and tides, which carried the ship toward the shore.

According to the National Transportation Safety Board report on the accident, the probable cause was the crew's overreliance on the automation (ARPA) and management failure to ensure the crew was adequately trained in understanding automation capabilities and limitations. The report stated that "all the watch-standing officers were overly reliant on the automated position display ... and were, for all intents and purposes, sailing the map display instead of using navigation aids or lookout information."

This accident not only represents a classic case of "automation complacency" related to inappropriately high trust in the automation [6], but also suggests the role of etiquette in the form of expectation violations. If the crew had been interacting with a human responsible for providing course recommendations from a GPS signal, they might reasonably expect to be informed in an appropriate way if the signal was lost. However, the loss of GPS signal was not displayed by the automation in a highly salient manner. As a result, the crew's etiquette-based assumptions about the lack of notification (namely, that all was as expected) proved disastrously wrong.

The opposite of overreliance—disuse of automation due to inappropriate distrust—has also been observed. We often see disuse of automated alerting systems with high false alarm rates [10], even though the systems are designed (and users are trained) to accept and investigate such false alarms. Early versions of the Traffic Collision and Alerting System (TCAS) used in commercial aviation were plagued with disuse following perceived high alarm rates. The added mental workload and perception of false alarms as a nuisance can also be interpreted in etiquette terms. Professional members of a cockpit crew are expected to be highly accurate and not to needlessly sound alarms that demand their superior's attention. A human subordinate who frequently "cries wolf" will be seen not only as unreliable but also as inappropriately demanding a supervisor's time and resources—another etiquette violation.

An Etiquette Experiment

As these case studies illustrate, inappropriate levels of trust in automation can be interpreted as resulting from etiquette violations and can profoundly impact efficiency and safety. But these are merely our interpretations of real-world instances; we wanted results from a more controlled experiment to demonstrate the interaction of etiquette, trust, and non-personified automation in high-criticality environments. As previously discussed, decreased reliability is generally correlated with decreased trust and decreased use of automation, but various factors moderate this general phenomenon. Is etiquette one such factor? Will good etiquette compensate for poor reliability and result in increased usage decisions? Will poor etiquette wipe out the benefits of good reliability? Such phenomena are not uncommon in human-human relationships, but will they extend to human-automation interactions—even in high-criticality domains?

The following experiment illustrates the effects of one dimension of etiquette on human-automation interaction. It also represents an initial attempt to establish a paradigm for investigating etiquette in non-personified human-automation interactions in a simulated high-criticality context. While its results are preliminary and based on a small sample size, they provide initial evidence that machine etiquette can strongly affect human trust, usage decisions, and overall human + automation system performance.

Experimental design. From the vast range of human etiquette behaviors, we chose to concentrate on a single dimension we call *communication style*, which refers to the "interruptiveness" and "impatience" of delivering relevant text messages. This was chosen as an etiquette dimension available to even fairly primitive and non-personified automation interfaces. The question of when and how a computer should interrupt a user with information is one that Horvitz [3, 4] has considered in his work on "courteous computing;" However, Horvitz' applications have not been in high-criticality domains, but rather desktop computer applications and email. It is quite possible the definition of "courteous" might either be irrelevant or substantially different on the flight deck than in the office.

We tested 16 participants (both general aviation pilots and non-pilots) on the Multi-Attribute Task (MAT) Battery, a flight simulator extensively used in previous high-criticality automation research [10]. The MAT incorporates primary flight (that is, joystick) maneuvering, fuel management, and an engine monitoring/diagnosis task (see Figure 1).

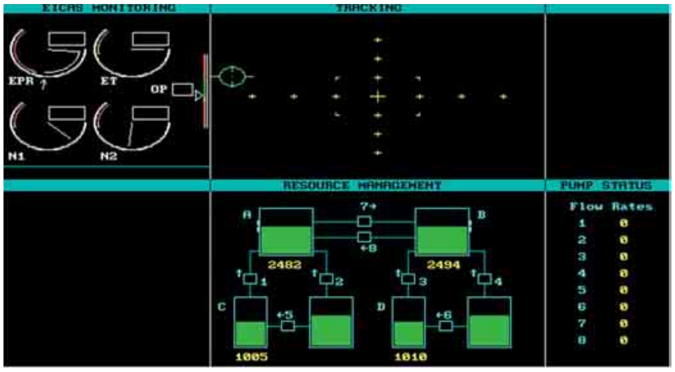


Figure 1. Sample interfaces from the MAT Battery and the EICAS display used in this experiment.

Participants always performed the first two tasks manually to simulate the busy operator of a high-criticality system. Intelligent automation support, modeled after the Engine Indicator and Crew Alerting System (EICAS) common in modern automated aircraft, was provided for the engine monitoring/diagnosis task. Automation monitored engine parameters, detecting potential engine faults, and advised participants when and what to examine to diagnose the faults. An example advisory message, presented as text, might be, "The Engine Pressure Ratio (EPR) is approaching Yellow Zone. Please check. Also, cross-check Exhaust Gas Temperature (EGT). There is a possible flame out of Engine 1."

We operationally defined good automation etiquette in this experiment as a communication style that was "non-interruptive" and "patient" while poor automation etiquette was "interruptive" and "impatient." In the non-interruptive case, the automation support was provided after a five-second warning and not at all when the operator was already doing the requested action (querying the EICAS system for engine parameters). Also, in this condition, automation was "patient" in that it would not issue a new query until the user had finished the current one. In the interruptive/impatient case, automation provided advice without warning and came on when the user was already querying EICAS. Automation also urged the next query before the user was finished with the current one (impatience).

These good and poor etiquette (in terms of communication style) conditions were crossed with the effects of automation reliability. Based on previous research [9], we chose two levels of reliability: low, in which automation provided correct advice on 6 out of 10 malfunctions, and high, when it was correct on 8 out of 10 malfunctions.

Results of experiment. Open-ended post-session interviews largely confirmed that participants saw our automation etiquette manipulations as intended: as either good or poor etiquette. For example, comments in the poor etiquette condition included "I hated it when it kept interrupting me when I was in the middle of diagnosis." And "I wished the computer would wait for me before asking the next question; that really bugged me." Few such comments were made in the good etiquette condition, which elicited statements such as "The computer was nice to direct me to the next question—that was helpful," or "One of the features I liked about the expert aid was that it was not intrusive—it came on when I needed it but not at other times."

We were primarily interested in the effects of etiquette and reliability on users' performance and on their rated trust in the automation. The percentage of correct diagnoses of engine malfunctions in all four conditions is shown in Figure 2. As expected, user diagnostic performance was significantly ($p < 0.01$) better when automation reliability was high (80%) than low (60%). Less obviously, good automation etiquette significantly ($p < 0.05$) enhanced diagnostic performance, regardless of automation reliability. Perhaps most interestingly, the effects of automation etiquette were powerful enough to overcome low reliability ($p < 0.05$). As the dotted line in Figure 2 indicates, performance in the low reliability/good etiquette condition was almost as good as (and not significantly different from) that in the high reliability/poor etiquette condition. These findings on diagnostic performance were mirrored in the results for user trust, shown in Figure 3. High reliability increased trust ratings significantly, but so did good automation etiquette.

A possible objection to these findings is that any interruption might be expected to degrade user performance. While this might itself be seen as an aspect of human-machine etiquette, we wondered whether our findings were due to the "rudeness" of this automation (which interrupted to "nag" users with instructions they were already performing) or due to the simple interruption itself. It seemed to us that "rudeness" came primarily from offering redundant, task-specific information that lent itself to a perception of being told to "hurry up." Accordingly, we ran a control group of four participants using non-specific interruptions, for example, "Maintaining primary flight performance is important, but do not forget to check engine parameters for possible malfunction." These interruptions were varied in their intrusiveness—they were either preceded by a warning or were not given at all if the user was engaged in diagnosis (non-intrusive) or were given with no warning regardless of user activity (intrusive). Under these conditions, as expected, diagnosis of engine malfunctions and user trust were both significantly higher in the high-reliability than in the low-reliability conditions. However, neither of these measures was significantly affected by the intrusiveness factor, in contrast to the main experiment. Apparently, less rude, non-specific interruptions were more easily ignored and did not adversely affect user-system performance or trust.

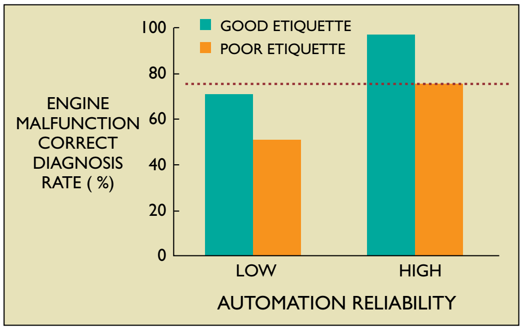


Figure 2. Effects of automation etiquette and automation reliability on the rate of correct diagnosis of engine malfunctions.

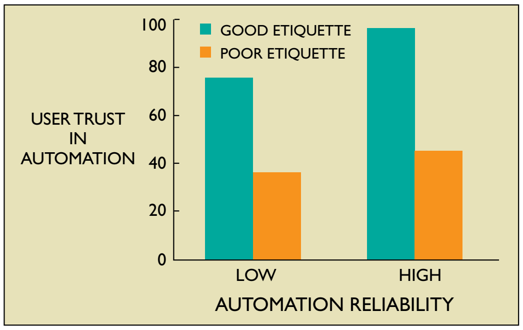


Figure 3. Effects of automation etiquette and automation reliability on subjective reports of trust in automation.

Etiquette Matters, Even for High-Criticality Automation

The results of this experiment provide what we believe is the first empirical evidence for the effects of automation etiquette in a simulated high-criticality system. Strong, if preliminary, evidence was obtained for the influence of automation etiquette on both user performance and trust in using an intelligent fault management system to diagnose engine malfunctions. The results clearly show that building reliable automation may not be enough for overall human + machine system efficiency: both user diagnostic performance and trust were lowered by poor automation etiquette even when the reliability of automation advice was high.

The results also provide support for the intriguing notion that good automation etiquette can compensate for low automation reliability. Some may find this result disturbing, since it suggests that developing robust, sensitive, and accurate algorithms for automation—a challenging task under the best of conditions—may not be necessary as long as the automation "puts on a nice face" for the user. We think not, for it was clear the best user performance (and the highest trust) was obtained in the high-reliability condition in which the automation also communicated its advice to the user in a polite and friendly manner.

References

1. Bainbridge, L. Ironies of automation. *Automatica* (1983), 775–779.
2. Cassell, J. and Bickmore, T. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User Modeling and Adaptive Interfaces 12* (2002), 1–44.
3. Horvitz, E. Principles of mixed-initiative user interfaces. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, PA, May 1999).
4. Horvitz, E., Kadie, C., Paek, T. and Hovel, D. Models of attention in computing and communication: From principles to applications. *Commun. ACM 46,* 3 (Mar 2003), 52–59.
5. Lee, J.D., and Moray, N. Trust, control strategies, and allocation of function in human-machine systems. *Ergonomics 35* (1992), 1243–1270.
6. Lee, J.D., and See, K.A. Trust in computer technology: Designing for appropriate reliance. *Human Factors* (2003).
7. Nass, C., Moon, Y., Fogg, B.J., Reeves, B., and Dryer, D.C. Can computer personalities be human personalities? *International J. Human-Computer Studies 43* (1995), 223–239.
8. Parasuraman, R., and Byrne, E.A. Automation and human performance in aviation. P*rinciples of Aviation Psychology.* P. Tsang and M. Vidulich, Eds. Erlbaum, Mahwah, NJ, 2003, 311–356.
9. Parasuraman, R., Molloy, R., and Singh, I. L. Performance consequences of automation-induced "complacency." *International J. Aviation Psychology 3* (1993), 1–23.
10. Parasuraman, R., and Riley, V.A. Humans and automation: Use, misuse, disuse, abuse. *Human Factors 39* (1997), 230–253.
11. Reeves, B., and Nass, C. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places.* Cambridge University Press/CSLI, NY, 1996.
12. Sheridan, T.B. *Humans and Automation.* Wiley, NY, 2002.

Raja Parasuraman (parasuraman@cua.edu) is a professor of psychology and director of the Cognitive Science Laboratory at The Catholic University of America, Washington, D.C.
Christopher A. Miller (cmiller@sift.info) is chief scientist at Smart Information Flow Technologies (SIFT) Inc., Minneapolis, MN.

ETIQUETTE ONLINE: *From* NICE *to* NECESSARY

In the early days of the Internet, an occasional sarcastic or confrontational remark was considered part of its "charm." Times have changed.

By Jenny Preece

Exasperated, Mary shrieked:

" ... I'm tired of these [nasty] comments. Why can't people be more civil?"

Tom, a member of another community, reported:

"... it took only one aggressive, insulting person to ruin the whole community for everyone else."

As Internet settlers form cyber communities, the importance of etiquette grows. Indeed, the lack of it is weakening sociability and even destroying online communities. Etiquette online is not just *nice* to have, it is *necessary.*

Like the pioneers of the Wild West, early Internet adopters were a rough and tumble gang. An occasional sarcastic comment, expletive, or confrontational challenge was part of the fun. But times have changed. Today's settlers flock online in the millions. A single word aptly summarizes these settlers: "diverse." These Internet users come from many cultures and walks of life. They arrive with a mix of expectations using a variety of technologies, which they access in different ways.

The new settlers include children and adults, healthy and infirm, eager and reluctant. English speakers dominate but other languages and cultures are gaining prominence. The number of women matches the number of men. Even, low-income families and those with limited education are beginning to appear [6]. This diversity can have its charm, but also can lead to unpredictable encounters, misunderstandings, and frustrated expectations. One person's clever joke is another person's offensive insult.

Connecting with others tops the "to-do" lists for new settlers. Internet shopping, gaming, and searching for information, particularly health information, are also near the top. The new settlers are a social bunch compared with the early pioneers, who focused on programming and information-oriented tasks.

Different types of technology require different forms of etiquette [4]. Textmessaging via a mobile phone is different from instant messaging and worlds apart from the asynchronous experience of email. A short abrupt comment that is acceptable in instant messaging may not be in email where some people expect to be addressed by name. Emotional affordances, syntax, and semantics vary across technology, too. Furthermore, new technologies may challenge previously accepted norms. Who would have guessed that having a stranger edit one's fastidiously composed prose without first asking permission would be acceptable? Yet this is exactly what happens in Wikis, designed for developing collaborative Web pages. Therefore, rules of etiquette are needed to preserve this spirit of flexibility while supporting reasonable behavior and good will. With such a wide range of communications software now available to users, etiquette is challenged when users move from one type to another. It's particularly easy to forget more subtle differences between the technologies.

Access conditions may also cause poor etiquette. Hundreds of miles and many time zones separate some participants; most cannot rely on face-to-face meetings to learn each other's norms. In addition, access to and experience with technology differs. Unwanted messages and large attachments that slightly annoy high-bandwidth users can be distressing for users with unreliable, expensive, dial-up facilities in remote locations.

The complexity of this rich mélange of users, goals, technology, and access conditions presents new challenges to etiquette online, particularly for the growing number of support communities where kindness, help, and empathy is anticipated. Good approaches for fostering etiquette online are therefore needed.

LISA HANEY

How Does Etiquette Develop?

Widely accepted ways of behaving reflect the attitudes and values of a community or society at large; indeed, they are its norms [5]. Social norms are people's beliefs about behaviors that are normal, acceptable, or even expected in a particular social context [7].

Norms, including rules of etiquette, are learned through experience in a community. For example, children observe how adults and other children behave, absorb these norms, and learn their community's etiquette at an early age. This role-modeling process continues throughout life. Other community members correct those who do not conform to expectations. Problems arise when people go into other cultures with different norms, particularly when the differences are subtle.

Gift giving provides a good example of an etiquette norm. If I receive a gift from American friends, I open it, thank them, and comment enthusiastically to show my pleasure. In Japan, I would instead thank the person and carefully put my gift aside to be opened later; opening the gift there and then would not be polite. My behavior would contravene the norm in Japanese society and be interpreted as poor etiquette.

In each culture, norms preserve or enforce comfort and empathy in the community. Consequently, when norms of etiquette are broken, discomfort, confusion, annoyance, embarrassment, and even fear may ensue.

What Problems Occur Online?

To understand how to create and support etiquette norms online, we must first understand what challenges them. Ask Internet users what online practice offends them most and you will get a slew of comments. For example, two friends offered the following:

"Receiving notes with inappropriate automated signatures or not addressing me by my name and ending without a farewell greeting and the sender's name—that's rude and unfriendly."

Another said: *"One-word answers and comments that don't refer to our conversation. How am I supposed to remember what 'Yes' refers to when I get 70–100 emails a day?"*

"Attaching large files that take a long time to download is thoughtless."

Surveys also provide examples of annoying behavior. A recent Internet survey used convenience sampling to collect opinions from 4,155 participants (istudio.vantagenet.com/cgi-bin/pollresults/002). The survey asked: "Which Netiquette issues aggravate you most?" The following problems were identified and listed in order of most to least mentioned: sending spam, forwarding bogus virus warnings; sending dumb jokes; typing in all caps or all lower case; lack of basic grammar and punctuation; including my email address in the CC: or TO: with a list of other email addresses; not editing email; including no hello or thank you; and poor use of formatting styles.

These responses suggest different reasons for the underlying behavior: unintentionally annoying behavior due either to poorly developed skills (for example, not mixing caps and lower case, not editing email, poor grammar) or failing to appreciate what others care about (for example, wasting time by sending dumb jokes); potentially malicious behavior (for example, sending spam); and absence of courtesy (not including hello and thank you). Surprisingly, aggressive comments—or flames—are not mentioned. Flames are comparatively infrequent but when they do occur, they can be devastating to an individual or community.

What are the Solutions?

Many researchers have reported breakdowns in etiquette online over the years (for example, [10]). We also know much about the causes of poor etiquette in textual communication; for example, absence of non-verbal feedback and reduced sense of responsibility between people who may never have to address each other face-to-face (for example, [2]). This knowledge has given rise to new interfaces that support identity and social interaction online (for example, [8]).

Two well-known approaches that specifically address etiquette solutions are setting rules (often called Netiquette) and moderating discussions. These can be effective but often prove inadequate. New approaches are called for that combine human judgment with technical efficiency. The following discussion reviews current processes and suggests some other approaches.

Netiquette. The usual approach is to develop written lists of rules for online behavior. Typing "Netiquette" in Google produces a stream of links to rules for online etiquette. Recent examples like Nokia's "Don't b a txt msg abuser" offer the following guidance for text-messaging with cell phones (www.thefeature.com):

- *Common courtesy still rules.* Contrary to popular belief, composing a SMS while you're in a face-to-face conversation is just about as rude as ...
- *Leave the slang to the kids.* Don't expect your stodgy superiors at work to be hip to the lingo of the SMS streets. And don't expect to win points with your kids by trying to be cool.
- *Remember your phone has an off button!*

One **person's clever joke is another person's offensive insult.**

These basic, commonsense rules can be effective, but they are often read and forgotten. When upheld by moderators, community leaders, and participants they are more likely to be successful; particularly for setting standards in the early days of technology adoption or early in a community's life.

Moderators. Although a well-known practice for preventing impolite behavior online, moderating can be demanding and time-consuming, particularly in active communities. Knowing when to control groups, when to let go, and how to bring in those who do not participate is not as easy as it might seem.

Role models. It seems obvious that role models could be used more online. After all, it's the way children learn etiquette [5]. Online moderators and early adopters tend to be role models for those that follow. By greeting, acknowledging and praising participants they encourage a climate of appreciation and respect that fosters etiquette.

Watching what others do is also a common strategy for newcomers to an online community. It enables them to judge the tone of the community before launching in, and so avoid causing offense, being ridiculed or putdown [9]. Even established community participants adopt this wise strategy when joining new communities because each community has its own standards and ways of behaving. What is acceptable in one may not be in another.

Evidence of constructive discussions, information exchanges, and empathic support are the trademarks of most successful communities. But there are other models for community. Some, like the Wild West saloons, are places that thrive on sarcasm, bawdiness, and punchy comments. Problems only arise when expectations are not met, especially in communities that are normally helpful and supportive.

Mentors. Mentoring could be helpful in technical, specialist, and cross-cultural communities. This approach has been used successfully in education environments. Based on principles advocated by the well-known psychotherapist Carl Rogers, students were given templates to teach them polite critiquing skills [12]. These templates promote addressing a person politely by name and being careful to check that you understand what she is trying to say before jumping in with questions, suggestions, or criticism; for example:

"[Name] What I think you mean is ... My own experience differs in this way ..."

"[Name] Please tell me what you want to do here, so that I can understand your point of view."

This approach encourages empathy and shared understanding, which is important online where nonverbal cues are reduced or missing.

Citizen regulation. Communities in which homegrown etiquette norms develop with little outside influence tend to be particularly successful. These citizen-regulated communities moderate their own behavior. If someone steps out of line, community members remind the offender about what is expected in a gentle or stern manner, as the situation warrants. This behavior is particularly noticeable in patient support communities. For example, in a health support community one participant told another who was miserable and in pain to *"tough it out"* and not be a wimp [3]. Fellow supporters quickly came to the rescue with comments like:

"Nonsense! A macho approach ... cannot avoid all problems, you were just lucky."

How does a community become self-regulating? Participants who take on specific roles in the community help to establish the community's norms and express its values [7]. Good community management and skillful moderating can help to make this happen by ensuring people treat each other politely and with respect. Even communities with a high turnover can develop strong etiquette norms and become self-regulating. Mary, a member of a vibrant patient support community that has existed for eight years, describes how this happens:

"Folks with different values eventually weed themselves out or are invited to leave. There is definitely respect granted to members of the [discussion] board. I am constantly amazed at how well members of the [discussion] board get on on a day-to-day basis."

Technically oriented processes. Some tools exist

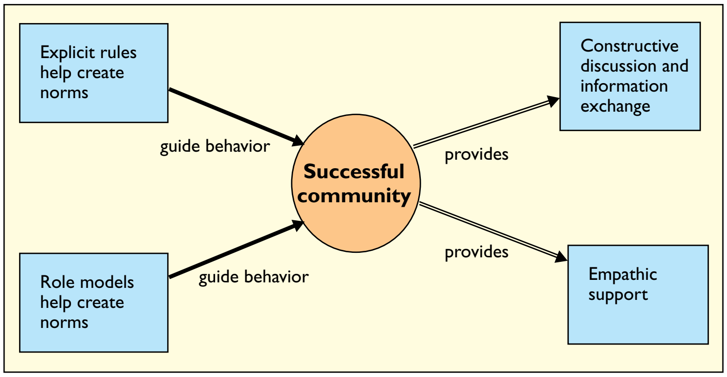
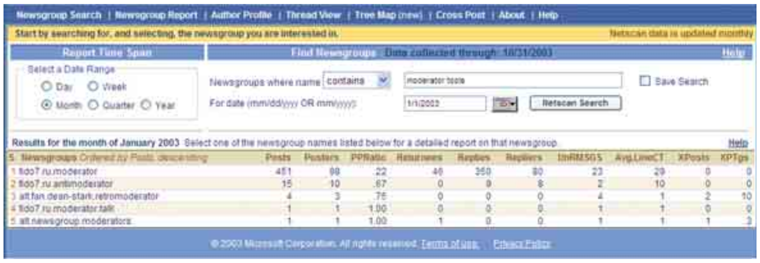

Explicit rules help create norms
guide behavior
Successful community
provides
Constructive discussion and information exchange
provides
Empathic support
guide behavior
Role models help create norms

Figure 1. The checkbox interface for selecting filters in ePrism.

to support etiquette online but more are needed.

Filters. Obvious obscenities and unwanted spam messages can be detected and eliminated using filters. Email systems enable users to set their own filters. Spam-detecting software like SpamKiller and SpamAssassin offer increasingly sophisticated filtering capabilities. The software comes with a large list of preset filters that check a sender's email address, subject line, message body, and embedded URLs for unwanted material. These lists are updated regularly with revised ones downloadable free of charge. In addition users can set their own word lists, email addresses and URLs for filtering. For example, ePrism (www.stbernard.com/products/eprism/products_eprism-spam.asp) provides a checkbox interface so that users can specify which filters they want in operation (Figure 1). Some text filters also substitute URLs, obscenities and other unwanted words with a message such as "Hi there!," "Cool it!," or nonsense words like "Hubbubb!"


Newsgroup Search | Newsgroup Report | Author Profile | Thread View | Tree Map (new) | Cross Post | About | Help
Netscan data is updated monthly
Report Time Span
Find Newsgroups
Help
Select a Date Range
○ Day ○ Week
● Month ○ Quarter ○ Year
□ Save Search
Netscan Search

Start by searching for, and selecting, the newsgroup you are interested in.

Data collected through: 10/31/2003

Newsgroups where name contains: moderator tools

For date (mm/dd/yyyy OR mm/yyyy): 1/1/2003

Results for the month of January 2003 Select one of the newsgroup names listed below for a detailed report on that newsgroup

5 Newsgroups Ordered by Posts descending	Posts	Posters	PPRatio	Returnees	Replies	Repliers	UnRMSGS	AvgLineCT	XPosts	XPTgs
1 fido7.ru.moderator	451	98	.22	46	350	80	23	29	0	0
2 fido7.ru.antimoderator	15	10	.67	0	9	8	2	10	0	0
3 alt.fan.dean-stark.retromoderator	4	3	.75	0	0	0	4	1	2	10
4 fido7.ru.moderator.talk	1	1	1.00	0	0	0	1	1	0	0
5 alt.newsgroup.moderators	1	1	1.00	1	0	0	1	1	1	3

© 2003 Microsoft Corporation. All rights reserved. Terms of use. Privacy Policy

Figure 2. Tabular display generated by Netscan showing the results of a search to identify UseNet News groups for discussing tools for moderators.

Community tools. Many community-building software applications provide tools for moderators to identify, approve, reject, delete, and edit messages, or to request the sender to edit his or her messages. Moderators can also delete whole threads or lock a topic so that no further discussion can occur and send automatic replies. Some systems also make it easy for participants to request help directly from a moderator.

Search and visualization tools. Tools are available for identifying, rating, and rewarding contributors to online discussions. Some offer data mining and visualization, such as Microsoft's Netscan (netscan.research.microsoft.com) where users can search for the most active UseNet Newsgroups, or the most active contributors within a group or groups, and the most valued members [11]. Results for different types of searches can be displayed in ranked tabular form (see Figure 2), graphically as a treemap for showing hierarchical relationships (see Figure 3), or in clusters, as appropriate. A similar approach could be adopted for etiquette if dictionaries of good and bad etiquette were added.

Rating and reward schemes. Led by Amazon and eBay, several e-commerce sites have rating schemes for customers to evaluate their products and services. Often a five-point scale is used in which "1" represents "poor" and "5" represents "excellent." Individual and average scores are then displayed for future customers to see.

Slashdot.org, a large, technical discussion community that receives several thousand messages per day, has a more sophisticated scheme known as "karma" for evaluating participants' contributions. Moderators award karma points to each contributor for the messages and stories they submit to the board. Karma can be rated as: "Terrible," "Bad," "Neutral," "Positive," "Good," and "Excellent." Karma ratings are also influenced by meta-moderating in which one moderator evaluates the ratings of another moderator ensuring that moderation is done fairly and can adjust karma points if it is not. This clever scheme therefore also checks moderator performance and enables moderation to be an open process involving community members. Schemes like this provide a basis for self-government in which communities determine and maintain their own etiquette standards. They encourage commitment and community service. Becoming a moderator is an honor.

What is the Way Forward?

Just as today's Internet settlers have different needs from the early pioneers, tomorrow's settlers will be different, too. More people from different cultures will come online. Ensuring good etiquette online will challenge moderators and community leaders unless better processes and tools for supporting good

Ensuring good etiquette online will challenge moderators and community leaders unless better processes and tools for supporting good etiquette are developed.

etiquette are developed. This should be a welcome challenge for researchers and developers. It is an opportunity to build bridges between people from different cultures, religions, genders, ages, and educational achievements.

The way forward is to develop processes that bring together the best human-oriented approaches with good technical support. New processes must include rich social structures as well as be labor-saving and scalable. Processes that encourage communities to develop self-governing etiquette standards are promising. Creative new sociotechnical processes are needed that address knowledge about: cross-cultural communication; counseling; group facilitation strategies; conflict management; community development; personal and group identity online; as well as strong technical know-how. In addition to preventing obvious breaches of etiquette, processes are needed for dealing with subtle etiquette problems such as clever pranks designed to incite reactions and inadvertent impoliteness due to cultural misunderstandings. A deeper knowledge of semiotics is needed to build these kinds of applications [1].

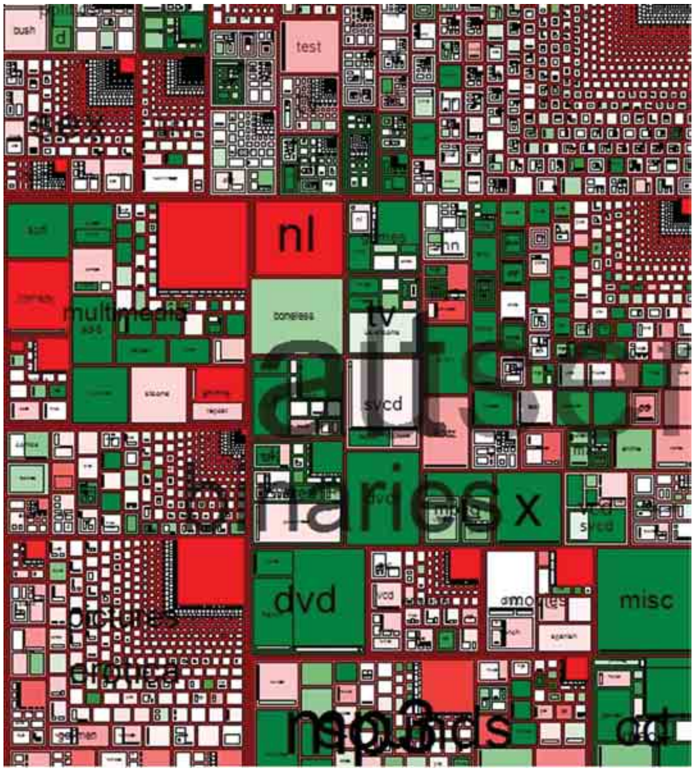


Figure 3. An excerpt from a graphical treemap display generated by Netscan showing the hierarchical relationships of Newsgroups in the Usenet.

As the Internet population continues to grow and diversify, etiquette will become increasingly important. Strong etiquette online is no longer just nice to have, it is necessary. **C**

References

1. De Souza, C.S. *The Semiotic Engineering of Human-Computer Interaction.* The MIT Press, Cambridge, MA, 2004, in press.
2. Katz, J.E. and Rice, R.E. *Social Consequences of Internet Use: Access, Involvement, and Interaction.* The MIT Press, Cambridge, MA, 2002.
3. Maloney-Krichmar, D., Preece, J. *Which Factors Facilitate Effective and Meaningful Support of Members of an Online Health Community? A Multilevel Analysis of Sociability, Usability, and Community Dynamics.* 2004, in press (Draft available from author).
4. Marx, G.T. New telecommunications technologies require new manners. *Telecommunications Policy 18,* 7 (1994) 538–551.
5. Morton, L.W. Civic structure. *Encyclopedia of Community: From Village to the Virtual World (Vol. 1).* K. Christensen and D. Levinson, Eds. Sage Pubications, Thousand Oaks, CA, 2003, 179–182.
6. Pew Research Center. The ever shifting Internet populations: A new look at Internet access and the digital divide. *Internet and American Life Survey* (Apr. 16, 2003); www.pewinternet.org/reports/ toc.asp?Report=88.
7. Postmes, T., Spears, R., Lea, M. The formation of group norms in computer-mediated communication. *Human Communication Research 26,* 3 (2000), 341–371.
8. Preece, J. (Ed.) Supporting community and building social capital. *Commun. ACM 45,* 4 (Special section, Apr. 2002), 36–73.
9. Preece, J., Nonnecke, B., Andrews, D. The top 5 reasons for lurking: Improving community experiences for everyone. *Computers and Human Behavior* (2004, in press).
10. Shapiro, N.Z. and Anderson, R.H. (1985) *Toward an Ethics and Etiquette for Electronic Mail;* www.rand.org/publications/MR/R3283/.
11. Smith, M. and Fiore, A. Visualization components for persistent conversations. In *Proceedings of ACM SIGCHI Conference* (Minneapolis, MN, Apr 2001).
12. Zimmer, B. and Alexander, G. The Rogerian interface: For open, warm empathy in computer-mediated collaborative learning. *Innovations in Education and Training International 33*, 1 (1996), 13–21.

Jenny Preece (preece@umbc.edu) is a professor of information systems at the University of Maryland, Baltimore County.

Exploring research-derived best practices for effective management of global software teams.

MANAGING CROSS-CULTURAL ISSUES IN GLOBAL SOFTWARE OUTSOURCING

By S. Krishna, Sundeep Sahay, and Geoff Walsham

IT outsourcing continues to be a booming business. The reasons why companies choose to outsource have been well-documented, including reduced cost, improved performance, and access to wider labor markets [1, 4]. One aspect of IT outsourcing is the outsourcing of software production. An important trend that started in the 1990s and continues to increase today is to outsource software production globally [9]. Much of the software development takes place at offshore locations, where costs are low and labor is often plentiful. Software suppliers normally maintain small bridgehead teams in the client countries for sales and customer liaison purposes. Outsourcers in turn often locate executives in the supplier countries to, for example, oversee large projects.

Illustration By Tomasz Walenta

All of this makes good economic sense for both sides of a cross-border outsourcing relationship, but it raises the question of how best to manage the process. In particular, cross-cultural issues are likely to become an important factor, as they have in the management of international joint ventures [2]. We have been investigating such issues over a five-year period, primarily through in-depth case studies, and with a particular empirical focus on outsourcers in North America, Western Europe, and Japan to software suppliers in India.[1] The primary conclusion from our research is that working across cultures when outsourcing software production is not a trouble-free process [10]. Particular societies tend to have distinct ways of working, and they can prove problematic when attempting cross-border collaboration. For example, Indian software companies have found they need to approach communication with U.S. and Japanese clients in very different ways. U.S. client companies normally work with extensive written agreements and explicit documentation, reinforced with frequent and informal telephone and email contact. In contrast, Japanese clients tend to prefer verbal communication, more tacit and continuously negotiated agreements, and less frequent but more formal use of electronic media.

Minimize cross-cultural issues through project choice of 'culturally-neutral' software	• Embedded software • Middleware
Use relationship to learn about leading-edge business systems, particular business sectors or higher-level software work	• For example, in telecommunications or e-business systems (outsourcer) • To gain domain expertise/move up the value chain (supplier)
Choose applications software only when good cross-cultural working feasible	• Cross-cultural match • Or major effort through staffing/ training

Table 1. Choice of software projects for cross-cultural outsourcing.

A second area where problems can arise in cross-border outsourcing is in the cultural adaptation of the bridgehead teams working in the client countries. Challenges not only concern the need to adapt to different ways of working but to cultural norms of social behavior, attitudes toward authority, and language issues. For example, some Norwegian outsourcers express a preference for Russian software suppliers rather than Asian companies. They explain this in terms of physical proximity, the similarity of the so-called European mindset, and the relative ease with which Russians learn the Norwegian language.

How can the cross-cultural difficulties of global software outsourcing relationships be addressed? This is the focus of the remainder of this article.[2]

Strategic Choice of Projects

One approach to handling the difficulties of cross-cultural working is through the appropriate choice of projects to be outsourced. For example, software that is to be embedded in operating systems or consumer products can often be specified in a relatively culturally neutral way, so less cross-cultural understanding is needed. Similarly, middleware is a layer of software between the network and the applications that performs the function of enabling different end-user systems to communicate more effectively with one another in advanced network applications. This can often be specified in a way that does not depend on continuous cross-cultural contact between outsourcer and supplier.

A second strategic approach to the choice of appropriate projects concerns the value in learning that can be gained through the particular projects. Many Indian software suppliers have acquired knowledge in the telecommunications and e-business domains through projects carried out for North American and European companies. This has resulted in some cases in Japanese outsourcers becoming interested in learning from the Indian software suppliers about leading-edge business systems in these domains. Software suppliers in developing countries, such as China, often focus on particular outsourced projects that offer the opportunity to gain domain expertise (in the banking sector, for example) or to move up the value chain, from tasks such as simple maintenance to higher levels of project involvement and ownership.

The development of application software is only a good strategic choice for cross-border outsourcing where conditions are such that effective in-depth working relationships can be achieved throughout the project. This exists where, for example, there is a good cultural match, such as that between Japan and China. This match relates not merely to linguistic closeness but also to compatible ways of working and understanding user attitudes. Similarly, Indian software developers speak English and often have extensive educational and cultural contact with the U.K., so there is normally a good cultural match here also. In contrast, Germany has not been very successful in attracting Asian software developers to work there, reflecting cross-cultural barriers of language and culture. The more difficult way to achieve effective cross-cultural working, where the cultural match is not close, is through careful attention to such issues as rela-

[1] We have also conducted more limited empirical work involving software suppliers in China and Eastern Europe and have studied the literature and interacted with other researchers on all aspects of cross-border outsourcing. On the basis of this, we believe the analysis and conclusions of our article apply generally to cross-cultural software outsourcing relationships.

[2] Our conclusions, as summarized in the tables here, apply to the relationship between outsourcers and software suppliers and thus have relevance for both sides of that relationship. In cases where the conclusion relates specifically to the outsourcer or supplier, we have indicated this in the table.

tionship management, staffing, and training, as shown in Table 2 and discussed in the next two sections.

Managing the Relationship

In all cases of cross-border outsourcing, active management of the client-supplier relationship on both sides is of key importance. The use of common systems is one way the relationship can be facilitated [7]. Such systems include agreed coordination and control mechanisms, for example, to report on and monitor project progress. Further harmonization can be achieved through common processes, such as systems development methodologies [6], and common compatible technologies in terms of computers, software systems, and telecommunications links.

Use systems to harmonize between outsourcer and supplier	• Coordination/control systems • Processes • Technology
Understand differences in norms and values	• Hierarchy/power • Business practices
Encourage 'negotiated culture' of cross-cultural teams developing compromise 'working culture'	• Bridgeheads and exchange mechanisms • Staffing and training

Table 2. Approaches to managing the cross-cultural relationship.

Although much can be achieved through the use of compatible technology and systems, it is important to recognize the limits of this approach. Major differences in norms and values cannot be harmonized, since they derive from deep-seated differences in cultural background, education, and working life. Examples include attitudes toward hierarchy and power and different business practices. For example, British managers in an outsourcing relationship with a particular Indian software supplier found that Indian programmers, in deference to authority, would not voice criticism in face-to-face meetings but would sometimes send their opinions in email messages after the meetings had disbanded. The British managers, used to intense interaction and the development of ideas through meetings, felt frustrated at this "polite" behavior [11]. Such difficulties can, however, be recognized and understood, but it requires substantial effort by both sides in the cross-border collaboration.

An attempt to understand and move some way toward the other partner in a cross-cultural collaboration has been called a negotiated culture perspective [2]. This perspective focuses on attempts to form and develop cross-cultural teams so a compromise working culture is achieved in which both sides of the partnership modify their work behaviors to take account of the cultural norms of their partners. For example, Germans and Japanese typically have very different attitudes toward "after-hours" working. However, it was noted in a particular German-Japanese international joint venture that some of the German managers began to stay later at work while many of the Japanese worked fewer hours than they were accustomed to in Japan [2]. Negotiated culture of this type is not something that can be achieved easily, and normally occurs only over a significant time period. Approaches to its achievement include the use of bridgehead teams that spend significant periods in client premises, exchange of staff on a long-term basis between cross-cultural partners, and staffing and training issues, as shown in Table 3a and discussed in the next section.

Staffing Issues

Although some movement toward other cultures is possible, it is unrealistic to expect expatriates in any country to be able to think and act like locals. This can create serious problems in areas such as application software development, where in-depth client contact is needed. To resolve this problem, successful outsourcing relationships often involve people who bridge cultures. For example, people originally from India, but with higher education and long-term residence in North America, have been reposted to India as expatriate managers for outsourcing projects. Such managers are often effective in overseeing complex outsourcing projects.

Recognize limits to cultural adaptation	• Foreigners cannot 'become' locals
Use 'cultural bridging' staff	• People rooted in both cultures • Locals as on-site workers (supplier)
Use locally-relevant recruitment and retention incentives	• Salary • But also status/expertise acquisition

Table 3a. Choice of staff and incentives.

A complementary solution to the problem of cultural bridging is for the software supplier to maintain a mixed cultural team in the client country. Locals in this country can then be used to perform a range of tasks, including being members of the sales force and of bridgehead teams in client premises or sometimes as senior staff dealing with the corresponding level in the client company. One reservation that so-called third-world software suppliers have about employing "first-world" staff is cost. However, such staff should be regarded as an essential overhead for major projects.

Give pro-posting cultural training for supplier employees working in outsourcer companies (supplier)	• Language • Cultural practices, norms, and values
Develop systematic on-the-job cross-cultural training	• To reflect on ongoing experience • To share knowledge with colleagues
Recognize that training needs are two-way (outsourcer)	• Not just for supplier staff

Table 3b. Training needs and processes.

How can people who can effectively bridge cultures be recruited and retained? Salary is one means, but there are cultural differences in the weight ascribed to this factor. In many Western economies, such as the U.S., salary is arguably the most important incentive for many people. In Japan, for example, it is less so, with many Japanese being very concerned about the status of the employing company rather than merely the salary. This is one of the problems that Indian companies, for example, have in recruiting employees in Japan. Similarly, German companies often find it difficult to locate managers with the appropriate personality profile for directing their development centers in Asia. A suitable person, apart from technical competence, must be open and adaptable to the different living and work environments. Recruitment and retention packages for staff must be tailored to the realities of these issues in specific contexts and labor markets.

Training

Many organizations, including those in the cross-border software business, offer pre-posting cultural training for employees (see Table 3b), varying from basic orientation courses to more substantive programs on language and cultural practices [5]. For difficult software outsourcing situations, such as Indian companies working in Japan [8], the fuller type of program is necessary.

Systematic on-the-job cross-cultural training is less common in our experience. Staff involved in cross-border relationships learn ways to achieve better cross-cultural collaboration, but there tends to be no structured opportunity in which this experience can be reflected upon and shared with colleagues in a formal way. Informal sharing of experience is important. However, we would argue that formal cross-cultural training should not stop when the staff member has arrived at the foreign employment location.

Cultural training is often perceived as necessary only in one direction, namely for the staff from the software supplier to learn about the culture of the countries of their client organizations. Regardless of any ethical concerns about such a culturally-blind attitude, it is also surely bad business practice. Training for cross-border software outsourcing should be seen as a two-way learning process. Then, all aspects of the relationship, from contract negotiation to the delivery of the final software product, can take place on the basis of a well-informed understanding of the culture and business practices of one's customer or supplier.

Conclusion

We have suggested ways to address and resolve problems and challenges in cross-border software outsourcing relationships. These involved the initial strategic choice of appropriate projects, ways of managing the relationship, and approaches to staffing and training. Although focused on software outsourcing, many of our conclusions are also relevant to global software teams operating within a specific company [3]. Some increased convergence of attitudes and approaches can be expected in such a context when compared to outsourcing to a different company, but the challenges of working in multicultural teams still apply.

We live in a more globalized world, in which there is increasing interconnection between different societies, and cross-border software outsourcing provides one example of this. But globalization does not imply homogeneity of culture [12]. In working in the contemporary world, we need to make extra efforts to tackle cross-cultural issues. This should lead not only to more effective business practices in areas such as cross-border software outsourcing but also to a world of increased cross-cultural understanding. C

References

1. Barthélemy, J. The hidden costs of IT outsourcing. *MIT Sloan Management Review 42*, 3 (Spring 2001), 60–69.
2. Brannen, J.V. and Salk, J.E. Partnering across borders: Negotiating organizational culture in a German-Japan joint venture. *Human Relations 53*, 4 (2000), 451–487.
3. Carmel, E. *Global Software Teams*. Prentice-Hall, New Jersey, 1999.
4. DiRomualdo, A., and Gurbaxani, V. Strategic intent for IT outsourcing. *Sloan Management Review 39*, 4 (Summer 1998), 67–80.
5. Forster, N. Expatriates and the impact of cross-cultural training. *Human Resource Management Journal 10*, 3 (2000), 63–78.
6. Gopal, A. Mukhopadhyay, T., and Krishnan, M.S. The role of software processes and communication in offshore software development. *Commun. ACM 45*, 4 (Apr. 2002), 193–200.
7. Heeks, R.B., Krishna, S., Nicholson, B., and Sahay, S. Synching or sinking: Global software outsourcing relationships. *IEEE Software 18*, 2 (Mar./Apr. 2001), 54–61.
8. Khare, A. Japanese and Indian work patterns: A study of contrasts. Chapter 7 in *Management and Cultural Values: The Indigenization of Organizations in Asia*, H.S.R. Kao, D. Sinha, and B. Wilpert, Eds. Sage Publications, New Delhi, 1999.
9. Lacity, M.C. and Willcocks, L.P. *Global Information Technology Outsourcing*. Wiley, Chichester, 2001.
10. Nicholson, B. and Sahay, S. Some political and cultural issues in the globalisation of software development: Case experience from Britain and India. *Information and Organization 11* (2001), 25–43.
11. Nicholson, B., Sahay, S. and Krishna, S. Work practices and local improvisations with global software teams: A case study of a UK subsidiary in India. In *Proceedings of the IFIP Working Group 9.4 Conference on Information Flows, Local Improvisations and Work Practices* (Cape Town, May 2000).
12. Walsham, G. *Making a World of Difference: IT in a Global Context*. Wiley, Chichester, 2001.

S. Krishna (skrishna@iimb.ernet.in) is a professor and chair of the Software Enterprise Management Program at the Indian Institute of Management in Bangalore, India.
Sundeep Sahay (sundeeps@ifi.uio.no) is a professor in the Department of Informatics at the University of Oslo, Norway.
Geoff Walsham (g.walsham@jims.cam.ac.uk) is a professor at the Judge Institute of Management at the University of Cambridge, U.K.

By Linda Wallace and Mark Keil

SOFTWARE PROJECT RISKS AND THEIR EFFECT ON OUTCOMES

How to identify the risks that interact to pose the most significant threats to successful project outcomes.

While many software projects deliver the functionality and performance promised by their developers on time and within budget, some result in systems that fail to deliver as promised. The Standish Group, an IT consulting firm, reports $275 billion is spent on software development projects each year in the U.S. alone [7]. More than 70% of these projects suffer total failure, cost overruns, schedule overruns, or deliver fewer functions than promised [6]. Examples of software project failures have been described in several books devoted to the subject [3, 4], and reports of troubled projects appear regularly in the business media. Failing to understand and manage the related risks can lead to project failure [1, 2], a costly problem that hasn't been completely addressed in the almost 30 years since such outcomes were first described in the literature. Software project managers would thus benefit from a better understanding of how software project risks affect project outcomes, leading to fewer project failures.

Illustration By Oreste Zevola

Risks are factors that can, when present, adversely affect a project, unless project managers take appropriate countermeasures. While a number of risk checklists have been developed, few firms have effectively incorporated them into their risk-management strategies. Managers need a simple approach for categorizing software project risks and providing insight into the relationship between different types of risk and project outcomes. At least one software-project-risk framework has classified individual risk factors according to their perceived importance and whether project managers view them as controllable [8]. Here, we explore the results and implications of its use in a multi-industry study of more than 500 software development projects managed by members of the Project Management Institute (see the sidebar "How the Study Was Done").

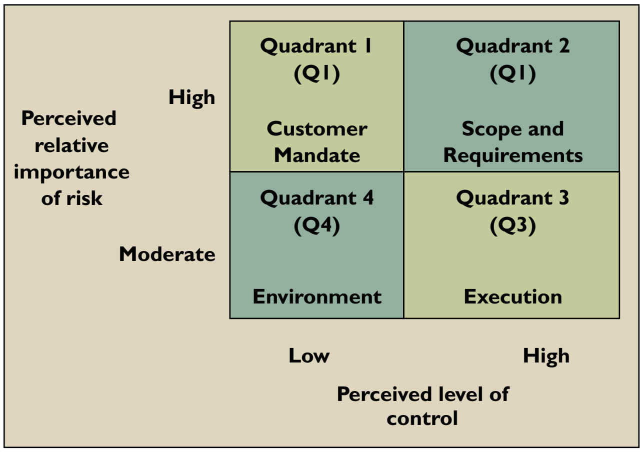


Figure 1. A risk categorization framework.

Identifying the risks complicating software development projects and incorporating them into a coherent risk management strategy is clearly a challenge [9]. While the various risk checklists have been proposed [11], relatively little effort has gone into organizing the risks. Moreover, only a handful of studies have examined the effects risk factors might have on the outcomes of projects [5, 12]. As a result, software project managers have few formal procedures to guide themselves in identifying the relative effects of the various risk factors and the trade-offs needed to manage them.

The framework in [8] for identifying software project risks represents one of the earliest attempts to create a useful risk management tool for software development managers (see Figure 1). Based on an international Delphi study of software project managers published in 1998, the framework organizes software project risks into four categories based on perceived importance (in the project manager's view) of the risk and perceived level of control project managers are likely to have in managing each one.

Customer mandate (quadrant 1, or Q1, in Figure 1) focuses on risk factors relating to customers and users, including lack of top management commitment and inadequate user involvement; though important to the success of a project, such factors are often beyond the project manager's control. Scope and requirements (Q2) focuses on risk factors associated with a project manager's inability to judge a system's scope. It also includes the risks associated with required functionality. Project managers should be able to control many of the risks associated with Q2. Execution (Q3) focuses on such risk factors as inadequate project staffing, inappropriate development methodology, failure to define roles and responsibilities, and poor project planning and control. Because most project managers are confident they can control these risks, they regard them as producing moderate rather than strong effects [8]. Finally, environment (Q4) focuses on risk factors in both internal and external environments, including changes in organizational management, that might affect a project.

Although the framework proposed in [8] is intuitively appealing, it remains untested, prompting project managers to ask: Do the risks embodied in each of the four quadrants in Figure 1 affect project outcomes? And do the risks in one quadrant interact with or offset the risks associated with any other quadrant? The framework's practical value could be established more clearly by exploring such questions. In addressing the first, we hope to give managers a better understanding of the relationships among different types of risk and project outcomes. In addressing the second, we hope to provide insights into the interactions that may exist among different types of risk and how to manage affected projects.

How different risks affect process outcomes. We first analyzed how the different types of risk affect process outcome—whether a project is completed on schedule and within budget. For process outcome, we found only scope/requirements risk (Q2) and execution risk (Q3) were significant and no interactions among quadrants. These results make sense intuitively, as poorly executed projects or projects involving unstable scope or requirements generally exceeding their budgets and schedules. We also found that execution risk is twice as important as scope and requirements risks in explaining process outcome. Because execution risk embodies factors associated with project teams, project complexity, and project planning and control, management must focus on budget and schedule concerns. Risks associated with customer mandate (Q1) and environment (Q4) do not significantly affect process outcome. In other words, it may be possible to deliver software on schedule and within budget without much regard for risks associated with customer buy-in or the organizational environment. However, these risks may still affect project outcome.

How different risks affect product outcomes. Our statistical analysis showed that customer mandate (Q1), scope and requirements (Q2), and execution risks (Q3) have a significant relationship with product outcome. Environment risks (Q4) did not significantly affect product outcome. As pointed out in [8], although environment risks, including changes in organizational management during a project, are relatively rare, they are almost always unpredictable. Although the project managers in the study reported they experienced some environment risks, these risks might not have been significant enough that they couldn't be handled by well-defined requirements and superior project execution. Perhaps because environment risks occurred less frequently, they were not identified as significant in our sample.

We found that several interactions among these first three quadrants also influence product outcomes (see Figures 2 and 3), prompting several observations. The most notable is that when execution risk is low, high levels of customer mandate or scope/requirements risk have little effect on project outcome. On the other hand, when execution risk is high, the effect of customer mandate risk and/or scope/requirements risk on project outcome is significantly greater—almost two times if customer mandate risk is high and nine times if scope/requirements risk is high. Practically speaking, this means that project managers who know execution risk is high and are unable to lower it must develop a risk-mitigation strategy focusing on minimizing the risks associated with scope/requirements and customer mandate. On the other hand, if execution risk is low, the effect of a high level of customer mandate or scope/requirements risk is minimal. This relationship suggests that if execution risks are minimized, the effect from the other types of project risk will likewise be minimized.

These results seem reasonable given that execution risk deals with such issues as project-team experience, project complexity, and project planning and control. Managing them effectively may compensate for, prevent, or neutralize the risks associated with scope/requirements, including scope creep, volatile requirements, and customer mandate. If execution risk factors are out of control (high risk), then the addition of high levels of risk in customer mandate and/or scope and requirements are sure to interact in ways that increase the project's execution complexity and difficulty.

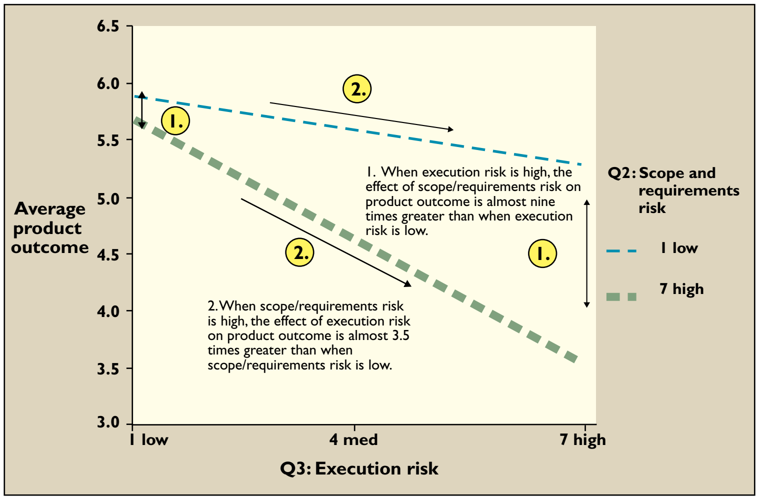


Figure 2. Interaction between quadrant 2 and quadrant 3 in Figure 1 and effect on product outcome.

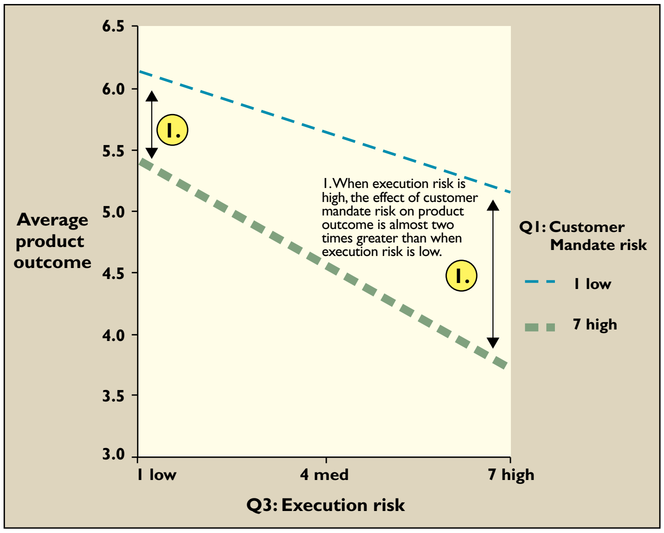


Figure 3. Interaction between quadrant 1 and quadrant 3 in Figure 1 and effect on product outcome.

The interactions also show that a low level of scope/requirements risk helps compensate for high levels of execution risk. If scope/requirements risk is kept low, then execution risk will have only a minimal effect on product outcome. On the other hand, if scope/requirements risk is high, execution risks must be managed to prevent a less than desirable product outcome—up to 3.5 times worse than when scope/requirements risk is low. If scope and requirements are under control, execution problems will likely have less of an effect on the product. Requirements define what the product should be and how it should perform. Thus, a clear understanding of these requirements could be responsible for producing a good product outcome, even as problems with process outcome persist.

These inevitable effects on product outcome suggest that in order to produce a successful application, software project managers must learn to control execution risks. If execution risks are managed effectively, then the effect of any level of customer mandate and scope/requirements risks will be minimal. If execution risks are not managed effectively, project managers must adjust their risk-mitigation strategies to minimize the risks associated with both scope/requirements and customer mandate.

How the Study Was Done

In order to test the framework in Figure 1, we compiled a list of software project risk factors known to affect development efforts. We used previous research and interviews with practicing project managers in the U.S. to write and refine 53 statements reflecting the range of software project risks project managers are likely to encounter [12]. We mapped the 53 risk factors into the four quadrants in Figure 1 (see the table here). The two researchers engaged in the project performed this mapping independently; while they generally agreed on the placement of most risk factors in the quadrants, the occasional disagreement prompted them to discuss factor placement until they reached a consensus.

We included the 53 items in the Web-based survey, and 507 software project managers—all members of the Project Management Institute's Special Interest Group on Information Systems—indicated the extent to which each risk factor was present during their most recently completed projects. We used a seven-point Likert scale; higher numbers represent a higher presence of a risk factor, lower numbers a lower presence. We also asked about two types of project outcomes:

Product. Measuring the success of the application produced through the development effort [10]; and

Process. Focusing on the success of the development process, or whether the project was delivered on schedule and within budget.

For each project, we calculated a score for each quadrant using an average of the values (from 1 to 7, indicating the degree to which a risk factor was present) of the risk factors belonging to the quadrant. We then used multiple regression analysis to determine the relationships among the different types of risk and project outcome (measured in terms of product and process), as well as among the interactions among the different types of risks.

Quadrant	Factor Description
1	Lack of user participation
1	Users resistant to change
1	Conflict between users
1	Users with negative attitudes toward the project
1	Users not committed to the project
1	Lack of cooperation from users
1	Lack of top management support for the project
1	Lack or loss of organizational commitment to the project
2	Undefined project success criteria
2	Conflicting system requirements
2	Continually changing system requirements
2	Continually changing project scope/objectives
2	System requirements not adequately identified
2	Unclear system requirements
2	Incorrect system requirements
2	Ill-defined project goals
2	Users lack understanding of system capabilities and limitations
2	Difficulty in defining the inputs and outputs of the system
3	Inadequately trained development team members
3	Lack of commitment to the project among development team members
3	Inexperienced team members
3	Frequent conflicts among development team members
3	Frequent turnover within the project team
3	Development team unfamiliar with selected development tools
3	Team members not familiar with the task(s) being automated
3	Negative attitudes by development team
3	Team members lack specialized skills required by the project
3	Project involves the use of new technology
3	High level of technical complexity
3	Highly complex task being automated
3	Project affects a large number of user departments or units
3	One of the largest projects attempted by the organization
3	Large number of links to other systems required
3	Immature technology
3	Project involves use of technology that has no been used in prior projects
3	Lack of an effective project management methodology
3	Inadequate estimation of project schedule
3	Lack of people skills in project leadership
3	Project progress not monitored closely enough
3	Inadequate estimation of required resources
3	Poor project planning
3	Project milestones not clearly defined
3	Inadequate estimation of project budget
3	Ineffective project manager
3	Inexperienced project manager
3	Ineffective communication
4	Resources shifted from the project due to changes in organizational priorities
4	Change in organizational management during the project
4	Corporate politics with negative effect on project
4	Unstable organizational environment
4	Organization undergoing restructuring during the project
4	Dependency on outside suppliers
4	Many external suppliers involved in the development project

The 53 project risk factors mapped to the quadrants in Figure 1.

Conclusion

We have explored how different types of risk influence both process and product outcomes in software development projects by analyzing input from more than 500 software project managers representing multiple industries. Our results reflect the importance of three of the four types of risk identified in the framework in [8]; the only one missing was environment (Q4). From the perspective of process outcome, managing the risks associated with Q2 and Q3 is critical. Thus, managers chiefly concerned with meeting schedule deadlines and budget limitations must find ways to reduce the risks associated with project execution, as well as the risks associated with scope and requirements.

The good news is that achieving successful process outcomes hinges on managing the risks associated with the two quadrants—Q2 and Q3—over which project managers feel they have the most control. However, every project manager knows that process isn't everything.

From the perspective of product outcomes, managing the risks associated with Q1, Q2, and Q3 is critical. Moreover, significant interactions take place among the different types of risk associated with these quadrants and influence product outcomes. As with process outcomes, the factor with the greatest influence on product outcomes is how the project is executed. Managing the risks associated with project execution requires project managers enlist experienced project team members who work well together and use proven project planning and control techniques. Scope and requirements, as well as customer mandate, also affect product outcomes.

The interaction effects we observed also suggest that good project execution can, to some extent, compensate for shortcomings in other areas, including customer mandate and the tactics employed for managing project scope and requirements. Similarly, if scope and requirements are identified, then execution problems are less important, though they could still affect the process and the likelihood of the project being completed on time and within budget.

Dealing with the triple constraints—scope, cost, and schedule—of project management almost always requires trade-offs. Our results suggest that projects emphasizing cost and schedule, or process goals, must be managed differently from projects emphasizing scope, or product goals. Ideally, both product and process outcomes should result in successful projects. Producing a project on schedule and within budget is of little use if the resulting product lacks the features and functions users thought they were paying for. However, in situations where budget and schedule are the top priorities, Q2 and Q3 must be the most closely managed. In other situations, product may be the most important outcome, so problems with budget and schedule may be overlooked if the resulting system is highly functional. Thus, Q3 risk must be minimized; to a lesser extent Q1 and Q2 risks must also be minimized.

Perhaps the most important conclusion to be drawn from the study is that project execution matters more than any other type of risk in terms of shaping both process and product outcomes. We cannot overemphasize the importance of employing experienced team members who work well together, managing project complexity, and exercising good project planning and control methods. Though other types of risk are important, managing the risks associated with project execution must be management's main focus. C

References

1. Boehm, B. Software risk management: Principles and practices. *IEEE Software 8,* 1 (Jan. 1991), 32–41.
2. Charette, R. *Software Engineering Risk Analysis and Management.* Intertext Publications, New York, 1989.
3. Ewusi-Mensah, K. *Software Development Failures.* MIT Press, Cambridge, MA, 2003.
4. Glass, R. *Software Runaways.* Prentice-Hall, Inc., Upper Saddle River, NJ, 1998.
5. Houston, D., Mackulak, G., and Collofello, J. Stochastic simulation of risk factor potential effects for software development risk management. *J. Syst. Soft. 59,* 3 (Dec. 2001), 247–257.
6. Johnson, J. *Chaos in the New Millennium: The Ghost of Christmas Future.* The Standish Group, West Yarmouth, MA, 2000.
7. Johnson, J. Turning chaos into success. *Software Mag. 19,* 3 (Dec. 1999), 30.
8. Keil, M., Cule, P., Lyytinen, K., and Schmidt, R. A framework for identifying software project risks. *Commun. ACM 41,* 11 (Nov. 1998), 76–83.
9. Longstaff, T., Chittister, C., Pethia, R., and Haimes, Y. Are we forgetting the risks of information technology? *Comput. 33,* 12 (Dec. 2000), 43–51.
10. Nidumolu, S. The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable. *Inform. Syst. Res. 6,* 3 (Sept. 1995), 191–219.
11. Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. Identifying software project risks: An international Delphi study. *J. Mgmt. Inform. Syst. 17,* 4, (Spring 2001), 5–36.
12. Wallace, L. *The Development of an Instrument to Measure Software Project Risk.* Doctoral Dissertation, Georgia State University, 1999; see www.cob.vt.edu/wallace/dissertation.pdf.

Linda Wallace (wallacel@vt.edu) is an assistant professor in the Department of Accounting and Information Systems at Virginia Polytechnic Institute and State University in Blacksburg, VA.

Mark Keil (mkeil@gsu.edu) is a professor in the Department of Computer Information Systems in the J. Mack Robinson College of Business at Georgia State University in Atlanta.

By Blake Ives, Kenneth R. Walsh, and Helmut Schneider

The Domino Effect of Password Reuse

One weak spot is all it takes to open secured digital doors and online accounts causing untold damage and consequences.

Password security is an essential form of user authentication both on the Internet and for internal organizational computing systems. Password protection schemes are used to protect relatively low-sensitivity systems such as access to online archives as well as highly sensitive corporate intranets or personal bank accounts. "Unfortunately the system of user name and password works less well than people believe," according to Bruce Schneier [10]. Moreover, the problem is escalating. The FBI found in a recent survey that detected system penetrations from outside the organization were reported by 40% of the organizations surveyed, up from 25% a year earlier [3].

Extensive literature on password security has evolved over the past 20 years. Much of it has been prescriptive, offering, for instance, advice for creating passwords and safeguarding them [7]. However, password vulnerabilities remain significant. Between 1989 and 1995, 22% of incidents reported to Carnegie Mellon's CERT Coordination Center (CERT/CC) involved password breaches [4]. The most common breach involved copying password files. It is now common to read news stories on password breaks such as the *Wall Street Journal's* report of NASA's loss of 6,000 passwords, or the Associated Press' account of the man who "pleaded guilty to federal charges of infiltrating sensitive computer systems, including those at Stanford University and NASA's Jet Propulsion Labora-

tory,"[1] or the use of an unauthentic request for bank customers to log onto a false Web site so the perpetrator could record passwords associated with online banking accounts [8].

In some more notable cases a security analyst in South Korea used an apparently stolen password from a rival to make a $22 million illegal trade; an admissions officer at Princeton was charged with using Social Security number passwords to pry into admissions decisions at Yale; and at least 21 students at Hofstra University were suspended and employees fired because of grade changes made with stolen passwords. Public access—now a common method for computer access—can be particularly susceptible to password theft due to potential weaknesses in the security of the workstation. For examples, a Kinko's facility was found to have keystroke-capture software installed, sending over 450 user names and passwords to a thief who subsequently used them for bank fraud [5]. Password vulnerability is not restricted to computer applications or operating systems. Indeed, a bug that released user passwords was found in the Netopia 650-T ISDB router firmware [11].

While password theft is a threat to the system from which the passwords were stolen, the network password vulnerability also threatens other systems. If users have many password-protected accounts and they reuse a password across more than one account, a hacker gaining access to one account may be able to gain access to others. If, for example, a hacker gains access to a weakly defended departmental file server and those passwords are stolen, those passwords could be used to gain access to a more secure corporate system. The hacker will reasonably anticipate that some users keep the same password on both systems. As e-commerce grows, the likelihood increases that a hacker who obtains access to passwords at a popular site might be able to use those user-IDs and passwords at another site. For example, there is an obvious and probably sizeable overlap between AOL and Citibank or BankOne and Amazon.com customers. A domino effect can result as one site's password file falls prey to a hacker who then uses it to infiltrate other systems, potentially revealing additional password files that could lead to the failure of other systems.

This is not just speculation. An intrusion was documented by the CERT/CC a few years ago in which a hacker was storing password files collected from several other sites on an intruded site. The hacker collected passwords from 186,126 accounts and had decrypted 47,642 of them. CERT/CC contends the passwords were then loaded into a password cracking tool [2].

Hacking tools can give more people the capability of launching more sophisticated attacks. Some hacking tools can coordinate attacks from several intruded systems onto targeted systems. If hackers used this automation and coordination with the untold number of stolen password files (though not necessarily reported or even recognized as having been stolen), a strong new hacking tool could exploit a huge database of known user-ID/password combinations. A variation of the now common denial-of-service attacks could be applied, greatly enhancing the effectiveness of password attacks.

Such attacks are difficult to defend against. In many systems, if a hacker tool repeatedly attempts to use incorrect passwords to access a particular account, the system will, usually after several tries, shut down the account in defense. In this scenario where variations on both the user-ID and password are involved, no such defense will be effective and the hacker will be free to run long lists of such pairs. Once the user-ID/password pairs have been tested, a hacker could add database entries capturing identities of systems accessible by a particular pair. This database would then be a powerful tool for systems infiltration, with coordinated attacks arranged to maximize the damage before the problem is discovered.

The rapid increase of e-commerce has fostered a proliferation of password protected sites. Forrester Research [6], for instance, reports that active Web users manage an average of 15 passwords on a daily basis. Unfortunately, Adams and Sasse [1] report that four or five passwords are the most a typical user can be expected to use effectively. Users are thus poorly equipped on a cognitive level to deal with today's need for multiple passwords, thus leading to password reuse on different systems. One source speculated that reusing a password is akin to revealing a password, thus potentially shifting legal liability for misuse to the user.

Users who reuse passwords often fail to realize their most well-defended account is no more secure than the most poorly defended account for which they use that same password. Unfortunately, the latter accounts may be quite weak as sites fail to implement the kinds of secure authentication methods now standard at the operating systems level. For instance, a journal editor given top-level access to a reviewing system was startled to discover he could see the passwords of the several hundred users of the system.

Risks will grow exponentially as password-protected systems proliferate, particularly among small e-commerce sites. Indeed, this problem inspired the title of our article—for with the falling of the weakest domino, other systems will follow, yielding new password information from which still more systems will fall. Today,

[1] www.nytimes.com/2000/11/08/technology/08HACK.html

We believe it is imperative that e-commerce security systems move expeditiously to either augmented password security systems or alternative security schemes such as smart cards or biometrics.

in an increasingly connected world, this recognition is also essential for, but largely ignored by, everyday users. Sasse et al. [9] trace many password weaknesses to the way in which passwords systems are implemented and call for greater collaboration between security administrators and users. Since issues such as poor password choice have been recognized for more than 20 years without major changes in user behavior, this does not appear to be an adequate general solution, although organizations should heed their advice to make the best use of current password systems.

Despite the apparent high costs and the given limitations of password security systems, we believe it is imperative that e-commerce security systems move expeditiously to either augmented password security systems or alternative security schemes such as smart cards or biometrics. Here, we describe alternatives to password security schemes, discussing both the strengths and weaknesses of those systems and including recommendations for practice and research for what should be done given the identified vulnerabilities.

Alternative Security Schemes

While there are numerous alternatives to password security systems, each involves trade-offs. Among the considerations are the cost to implement, the time required to use, any special considerations regarding place of use (for example, must it be from a particular computer), ability to change the scheme if it is compromised, physical limitations, health considerations (for example, a fingerprint reader on a public site), non-transferability, time stamped, and so on. It is beyond the scope of this article to thoroughly explore each of these alternatives and their limitations. However, the main classes of alternative technology are discussed to demonstrate their potential.

In public-key encryption (PKE) the user is authenticated by the private key used to encrypt a message to the server. While similar to a password, the private key has two features that increase its security. First, the private key is stored on a client computer or smart card and can be of considerable length, thus eliminating the need for the user to memorize the code while also avoiding the possibility of the user generating an easy to guess code. Second, the server verifies the code by correctly decrypting certain information sent by the client rather than comparing to a password file thus eliminating any server-side storage of passwords, encrypted or not. However, a user's private keys must be protected on the client side, thus changing the location of a potential theft.

Public-key infrastructure (PKI) uses PKE to authenticate users across a number of different applications or systems. A PKI can be set up by an organization to be used across its various systems or it can be set up by a third-party vendor to provide authentication services to many vendors. PKI can allow a user to have a single private key that can be used across some or all of the user's needs, simplifying key management for the user. In this situation, loss of the user's private key can make several or all of the user's systems vulnerable in a similar way as when a user chooses the same password to enter for multiple systems. However, in this case, greater emphasis may be placed on making sure the system is difficult to penetrate with responsibility for the key remaining in the hands of the user. Further, a method of centrally revoking a key can be put in place so that a stolen key can be quickly disabled for all systems. Despite some positive attributes, PKI systems are so difficult to use and so poorly implemented, they are usually viewed as ineffective. Indeed, according to one observer, "digital certificates provide no actual security for electronic commerce; it's a complete sham" [10]. Education and improvements in standards and technology may make this approach more effective in the future.

Biometrics involve some form of data obtained about the user's physiology such as scanning an eye, fingertip, or face, or capturing patterns in voices or motions made while signing a document. Again, a biometric can be seen as another form of password, in this case generated by the interaction of a human and a scanning device. While convenient, the digital scan or pattern is vulnerable to network analyzers and, unlike a password, cannot be changed once stolen. This generally limits the use of biometrics to scenarios where the network and biometric capture device are secure.

Smart cards can be used in a number of ways, from simply storing a password to performing complex encryption such as PKE on the card. In fact, a PC can be used as a smart card when electronic purse software is used to store passwords. While the PC itself is vulnerable to network attack, threats to the server side and client side are drastically reduced if a tamper-resistant card employs internal PKE and an algorithm that does not require the user's private key be transmitted from the card or stored on a server. The private key, which replaces the password, is only used within the card. This technology appears promising if the client-side device remains in friendly hands. Tokens that generate changing codes over time are resistant to network analyzers and, if implemented in a convenient manner, may offer some promise for effective e-commerce security. So too, in the long run, might systems involving trusted intermediaries. As solutions arise, however, so do new problems. For example, if smart card technology were well developed, privacy concerns might increase.

Recommended Courses of Action

While there is no silver bullet solution to the user authentication problem, it is still important to work toward improvements in password usage, security systems, and understanding threats. Improvements to authentication systems can be made by both practitioners and researchers and must be addressed from many points of view. Here is a summary of recommendations for improved practice and further research.

Recommendations for practice:

- Improve password guidelines to include a recommendation to limit the repetition of passwords across sensitive systems.
- Improve audit methods to audit current use of passwords and other identification schemes with emphasis on weaknesses at the user, client, network, and server interfaces.
- Improve security education to train and educate the general public as well as universities, e-commerce merchants, governments, and other organizations. Further, users should be trained in more advanced methods such as biometrics, PKE, and smart cards to be better prepared to use such technology as needed.
- Improve security education for application developers on systems design security issues.

Recommendation for research:

- Conduct research into the practices of security management at multiple levels in the organization including planning, implementation, and maintenance.
- Conduct research into alternative technologies including research that considers the user aspects of its implementation.
- Continue research into what influences user security behavior including policies, education, and incentives.
- Continue research into user password behavior including password reuse behavior and the potential impact of training schemes.
- Conduct research to better understand what security weaknesses are most useful to the hacker and what might be future hacking trends.
- Conduct research into trends in system break-ins. Analyze the extent to which security measures are becoming stronger or weaker over time.

Conclusion

Password security today is the perception of security rather than the reality. Indeed, the Emperor really does have very few clothes. The growth of e-commerce has led to a huge increase in the numbers of passwords required by individual users—very often duplicated over and over throughout the Web. In such an environment, a password, and all the accounts it provides access to, are no more secure than the weakest system using that password. Like dominos, when a weak system falls prey to hackers, information will be revealed that will aid the hackers in infiltrating other systems, potentially leading to the fall of many other systems, including systems with far better security than the first. Until these problems are addressed, there remains a very real threat to the fabric of our increasingly electronic society.

References

1. Adams, A., and Sasse, M.A. User are not the enemy. *Commun. ACM 42,* 12 (Dec. 1999), 40–46.
2. CERT. CERT Incident Note IN-98.03, Password Cracking Activity. CERT Coordination Center, Carnegie Mellon University, 1998; www.cert.org/incident_notes/IN-98.03.html.
3. Harreld, H. Security: An uneasy alliance. *Infoworld 23*, 13 (Mar. 26, 2001), 42–44.
4. Howard, J.D. An analysis of security incidents on the Internet, 1989-1995. Dissertation, Engineering and Public Policy, Carnegie Mellon University, 1997.
5. Jesdanan, A. Spy case shows public risk: Names, passwords stolen at Kinko's Internet terminals. *South Florida Sun-Sentinel* (July 23, 2003).
6. Kanaley, R. Login error trouble keeping track of all Your sign-ons? Here's a place to keep your electronic keys, but you'd better remember the password. *San Jose Mercury News* (Feb. 4, 2001).
7. Morris, R. and Thompson, K. Password security: A case history. *Commun. ACM 22,* 11 (Nov. 1979), 594–597.
8. Needham, K. Internet banking passwords stolen. *The Sydney Morning Herald* (Mar. 19, 2003).
9. Sasse, M.A., Brostoff, S., and Weirich D. Transforming the "weakest link"—A human/computer interaction approach to usable and effective security. *BT Technology Journal 19*, 3 (July, 2001), 122–131.
10. Schneier, B. *Secrets and Lies: Digital Security in a Networked World*. Wiley, New York, 2000.
11. Security Focus. Netopia 650-T ISDN Router Username/Password Disclosure Vulnerability. bugtraq id 1952, 2000; www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fvid%3D1952.

Blake Ives (bives@acm.org) is a professor at C.T. Bauer College of Business, University of Houston, TX.

Kenneth R. Walsh (kwalsh@uno.edu) is an associate professor in the College of Business Administration, University of New Orleans, LA.

Helmut Schneider (hschnei@lsu.edu) is a professor at Ourso College of Business Administration, Louisiana State University, Baton Rouge, LA.

By Anat Hovav and Paul Gray

Managing Academic E-Journals

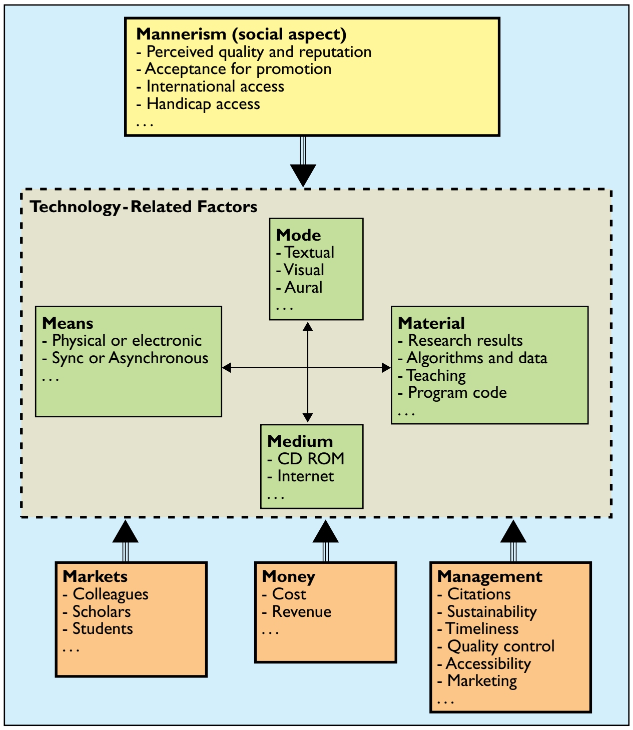
Though e-publishing is relatively inexpensive, e-publisher survival still depends on the age-old virtues of content quality and author credibility.

You are thinking of publishing an electronic journal. The advantages seem obvious: no backlog, quicker editing and production cycles, minimal initial capital (volunteer labor and access to the Web), and you can publish new types of material. But e-publishing also involves risks, hidden costs, and trade-offs. Here, we present a framework for implementing and managing academic e-journals.

The seven factors introduced in [3] influence publishing in the information age. We've adapted this framework to academic e-publishing, adding an eighth factor—Mannerism (see the figure here). The four technology-related factors are driven by the Medium, or the tools used to store and display the Material, or the published content, which can vary depending on the Medium being used. The Mode includes the symbols and language used to present text, as well as visual or audio material. The Means of distribution describes how publications are delivered: physically (CD-ROM); electronically (Internet); periodically (synchronized); or as soon as it's ready (asynchronous).

In academic publishing, the Market is generally limited to the scholars within a particular discipline. The number of subscribers is thus limited and shrinking, except in a few growing fields. Journals that market to both academia and practitioners retain a larger subscriber base. For example, the circulation of

MIS Quarterly is approximately 3,000, compared to *Communications*, with approximately 85,000. Money refers to both the costs of and the revenues from publishing. Universities and research institutions absorb the costs of creating, submitting, and refereeing articles, regardless of the Medium used to distribute them. The main cost reduction for e-journals is in editing, production, and distribution. Some costs, including storage and bandwidth, are transferred to the reader. Studies of the economics of e-journals [7] emphasize the benefits to the publisher and ignore the increased costs to the consumer. The management of academic journals extends beyond the preparation and distribution of material. Publishers and editors must ensure that e-journals are sustainable, accessible, timely, reputable, and of high quality.



Academic e-journal publishing framework.

Mannerism refers to the social characteristics of academic publishing. For example, though perceived quality is difficult to measure and control, it determines a journal's success [9] and the level of participation by reputable authors [2]. Other social factors include promotion and tenure, handicap access, and international access.

Implications

E-journals involve new opportunities and challenges. Some issues are journal-dependent and addressed by individual editors; others are universal and affect the overall academic milieu.

Technology factors (such as Medium) increase available space at minimal additional cost. Material (such as computer programs and algorithms) can be added; content can be expanded beyond traditional research; and richer Modes (such as audio and video) can be included. The editor decides how to balance the benefits of technology and the risk of information overload. E-journals are designed to work with current technologies. When the technology becomes obsolete, and unless backward integration is maintained, journal content becomes unreadable. What makes this problem universal in scope is the complexity and cost of conversion and the required infrastructure.

In applied fields, including information systems, academic journals try to expand their Market to practitioners who seek relevant, accurate, and timely information, rather than rigorous theoretical development. E-journals offer ways to increase circulation. Lower production costs allow publishers to create dual outlets, each targeting a particular market; examples are *Communications of AIS* and the *Journal of AIS*. The dynamic structure of e-journals provides an individual journal with multiple entry points and levels of theoretical rigor; an example is the *Journal of Information Law and Technology*. Reference [5] is an example of a multi-level article from which readers can select desired levels of detail for each section, depending on their interest and orientation.

The three Money characteristics [7] unique to academic publishing are:

Conflicting stakeholder incentives. Author incentives, which are based on journal ratings, differ from library incentives, which are based on cost and constrained by budgets;

Little price competition. Prices and revenue are unrelated to journal ranking. The revenue to the publisher from a 20-page article in a computer science journal can range from $1,000 to $8,000 notwithstanding the journal's prestige; and

No pay for scholars, unlike for commercial authors. Moreover, article refereeing and much of the editing is often free to the publisher.

The cost of producing an e-journal is lower than for producing a print journal (p-journal) because it requires no printing, binding, shipping, or storage. E-journals, however, introduce the issue of copyright payment structures. Payment structures are evolving toward a collective licensing system whereby subscribers access everything published in a particular system (such as the ACM Digital Library.) Whereas in

IT IS UNCLEAR TO WHAT EXTENT INTRODUCING ADVANCED TECHNOLOGIES SUPPORTS THE ULTIMATE OBJECTIVE OF RESEARCH—CREATING KNOWLEDGE.

1999, 70% percent of e-journals were free [6], in 2003, only 25% were free, supporting our assertion that free e-journals are not viable in the long term.[1] Publishers offer a variety of fee structures for e-journals, including combined (paper and electronic) subscription, pay-per-view, and bundled with association membership dues. However, distributed articles with hyperlinks to external Web resources lose the hyperlinking advantage with any charging scheme.

Each Management task introduces new challenges. For example, long-term sustainability helps maintain collective knowledge. However, the infrastructure does not exist for maintaining archives, as with, say, the Library of Congress. Individual publishers can archive their own e-journals. Each e-journal's sustainability thus depends on its publisher's survival. If a particular manuscript resides on a single server, archiving it is easier than archiving distributed articles [1].

Marketing a new e-journal is especially difficult. Established publishers might use a brand name (such as ACM), combining their e-journal marketing efforts with that of more traditional publications. Independently published e-journals can use announcement services, listservs, and Internet-based search agents to market their e-journals. Researchers traditionally find articles via secondary indexes, such as abstract services. Few e-journals are indexed today, causing some scholars to overlook relevant e-articles, thus reducing the benefits of potential anytime-anywhere e-journal access.[2]

E-journals also reduce publication cycle time. The time needed for researching, writing, refereeing, and editing remains the same. But production time is shortened by eliminating printing, binding, and shipping and by publishing articles when they're ready, thus eliminating backlogs of accepted articles. Publishers must balance quicker publication cycles with irregular article flow and the reader's potential information overload.

Citation records are often used to measure publication quality, influencing author reputations, promotion, and compensation. E-journals that allow "published" articles to be modified over time must publish each version to ensure a proper chain of citation is maintained.[3]

To maintain quality and ensure space limitations are not exceeded, gatekeepers tend to accept studies on topics within a discipline's established paradigms. Unorthodox work is difficult to publish and often rejected by reviewers [9], fostering a philosophy maintaining that most manuscripts should be rejected. The relatively low cost of establishing an e-journal allows scholars other than gatekeepers to start publications (such as *Philosophical Foundations of IS*) that accept new ideas and values and work with authors to improve promising papers to high-quality publishable work. This philosophy requires a shift in academic principles, the approval of the gatekeepers, and modification of promotion guidelines, and are beyond the capabilities of individual e-journals [10].

The academic social environment is the main hurdle for academic e-journals. While p-journals are perceived as elitist products (as journal costs increase, they become available only to well-endowed universities), the acceptance of e-journals is a relatively slow process. Like p-journals, e-journals need to maintain standards for quality and accuracy. E-journals offer a relatively high acceptance rate and minimal backlog, conflicting with the notion that quality requires a low acceptance rate and large backlog. E-journal editors contend with technical instability, lack of control, and few definitive standards on the Internet [4].[4] In 1999, only a third of e-journals were peer-reviewed [6], decreasing acceptance for all e-journals.

[1] Calculated based on the Colorado Alliance of Research Libraries index, including both pure e-journals and combined (paper and electronic) journals.

[2] Primarily true for independent, pure e-journals. Electronic versions of existing p-journals are indexed with their paper counterparts.

[3] If article B cites article A and article A is subsequently changed, the citation may no longer be valid.

[4] Occasional downtime can hurt the reputation of an otherwise well-run e-journal.

PERHAPS THE MOST CHALLENGING TASK AHEAD FOR E-JOURNALS IS THE MANAGEMENT OF ACADEMIC SOCIAL ACCEPTANCE AND PERCEPTIONS.

Acceptance for promotion[5] is a major success factor [2] and instrumental in distributing rewards to researchers. Scholars with greater awareness of e-publishing perceive e-journals as superior [8]. Public relations efforts can increase awareness. An e-article that can be reviewed by promotion committee members in traditional paper format is more likely to be accepted than a distributed e-article that uses various modes and that is not printable. Therefore, some publishers provide a conventional printable version of their e-articles, and others format their e-journals as PDF files to look like p-journals.

Unlike print, e-journals offer only limited support for visually and aurally impaired users, though they can do so if properly designed. E-journals do, however, provide greater access to universities in poor countries by eliminating distribution delays and reducing distribution costs. Technological limits can be overcome by providing multiple formats, putting multimedia into appendices, and providing text descriptions of complex images.

Conclusion

Our eight-factor model provides a systematic framework for thinking about managing academic e-journals. Some changes, such as reduced cycle time and publication costs, are clearly beneficial. Others, such as the need for backward integration and universal sustainability, must still be resolved.

It is unclear to what extent introducing advanced technologies supports the ultimate objective of research—creating knowledge. Properly designed e-journals increase visual appeal without causing information overload, support living scholarship while retaining traceability, and create products perceived as high quality.

Perhaps the most challenging task ahead for e-journals is the management of academic social acceptance and perceptions. Academic p-journals are established and reliable institutions. They may be less efficient in terms of cost or time, but they are entrenched. In the interim, some publishers use electronic media only as a delivery mechanism for paper-like articles.

We expect e-journals to become an integral part of academic publishing. However, they represent a systemic change and will, like the automobile at the turn of the 20th century, require a fundamental change in both publishing infrastructure and academic culture.

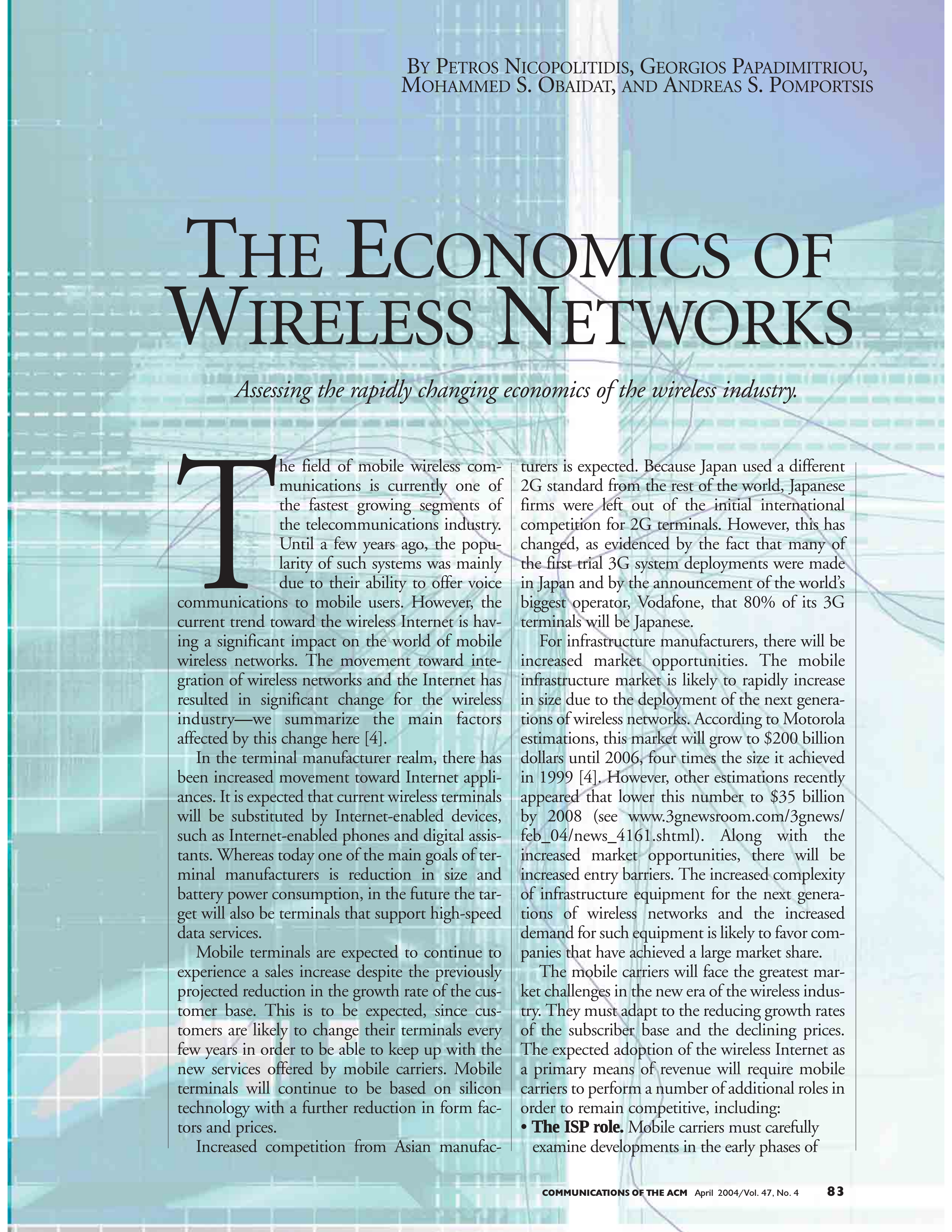
References

1. Baudoin, P. Uppity bits: Coming to terms with archiving dynamic electronic journals. *Serials Librarian 43,* 4 (June 2003), 63–72.
2. Berge, Z. and Collins, M. IPCT journal readership survey. *J. Amer. Soc. Inform. Sci. 47,* 9 (Sept. 1996), 701–710.
3. Eisenhart, D. *Publishing in the Information Age: A New Management Framework for the Digital Era.* Quorum Books, Westport, CT, 1994.
4. Harnad, S. Implementing peer review on the Net: Scientific quality control in scholarly electronic journals. In *Scholarly Publishing: The Electronic Frontier,* R. Peek and G. Newby, Eds. MIT Press, Cambridge, MA, 1996, 103–118.
5. Hars, A. Web-based knowledge infrastructures for the sciences: An adaptive document. *Commun. Assoc. Inform. Syst. 4,* 1 (July 2000), 1–34.
6. Mogge, D. Seven years of tracking electronic publishing: The ARL directory of electronic journals, newsletters, and academic discussion lists. *Library Hi Tech 17,* 1 (1999), 17–25.
7. Odlyzko, A. The economics of electronic journals. *First Monday 2,* 8 (Aug. 1997).
8. Palmer, J., Speier, C., Wren, D., and Hahn, S. Electronic journals in business schools: Legitimacy, acceptance, and use. *J. Assoc. Inform. Syst. 1,* 2 (Mar. 2000), 1–29.
9. Schauder, D. Electronic publishing of professional articles: Attitudes of academics and implications for the scholarly communication industry. *J. Amer. Soc. Inform. Sci. 45,* 2 (Mar. 1994), 73–100.
10. Weber, R. The journal review process: A manifesto for change. *Commun. Assoc. Information Syst. 2,* 12 (Aug. 1999), 1–22.

Anat Hovav (anat.hovav@temple.edu) is an assistant professor in the Management Information Systems Department at Temple University in Philadelphia.

Paul Gray (paul.gray@cgu.edu) is a professor emeritus in the School of Information Science at Claremont Graduate University in Claremont, CA.

[5] Unlike the gatekeepers, tenure-and-promotion committee members evaluate articles outside their own disciplines and are thus less familiar with intradisciplinary nuance.



By Petros Nicopolitidis, Georgios Papadimitriou,
Mohammed S. Obaidat, and Andreas S. Pomportsis

The Economics of Wireless Networks

Assessing the rapidly changing economics of the wireless industry.

The field of mobile wireless communications is currently one of the fastest growing segments of the telecommunications industry. Until a few years ago, the popularity of such systems was mainly due to their ability to offer voice communications to mobile users. However, the current trend toward the wireless Internet is having a significant impact on the world of mobile wireless networks. The movement toward integration of wireless networks and the Internet has resulted in significant change for the wireless industry—we summarize the main factors affected by this change here [4].

In the terminal manufacturer realm, there has been increased movement toward Internet appliances. It is expected that current wireless terminals will be substituted by Internet-enabled devices, such as Internet-enabled phones and digital assistants. Whereas today one of the main goals of terminal manufacturers is reduction in size and battery power consumption, in the future the target will also be terminals that support high-speed data services.

Mobile terminals are expected to continue to experience a sales increase despite the previously projected reduction in the growth rate of the customer base. This is to be expected, since customers are likely to change their terminals every few years in order to be able to keep up with the new services offered by mobile carriers. Mobile terminals will continue to be based on silicon technology with a further reduction in form factors and prices.

Increased competition from Asian manufacturers is expected. Because Japan used a different 2G standard from the rest of the world, Japanese firms were left out of the initial international competition for 2G terminals. However, this has changed, as evidenced by the fact that many of the first trial 3G system deployments were made in Japan and by the announcement of the world's biggest operator, Vodafone, that 80% of its 3G terminals will be Japanese.

For infrastructure manufacturers, there will be increased market opportunities. The mobile infrastructure market is likely to rapidly increase in size due to the deployment of the next generations of wireless networks. According to Motorola estimations, this market will grow to $200 billion dollars until 2006, four times the size it achieved in 1999 [4]. However, other estimations recently appeared that lower this number to $35 billion by 2008 (see www.3gnewsroom.com/3gnews/feb_04/news_4161.shtml). Along with the increased market opportunities, there will be increased entry barriers. The increased complexity of infrastructure equipment for the next generations of wireless networks and the increased demand for such equipment is likely to favor companies that have achieved a large market share.

The mobile carriers will face the greatest market challenges in the new era of the wireless industry. They must adapt to the reducing growth rates of the subscriber base and the declining prices. The expected adoption of the wireless Internet as a primary means of revenue will require mobile carriers to perform a number of additional roles in order to remain competitive, including:

- **The ISP role.** Mobile carriers must carefully examine developments in the early phases of consumer use of the Internet. Considering how local telephone companies in the U.S. missed the opportunity to become major ISPs when AOL emerged as the initial dominant provider, mobile carriers will want to ensure a similar situation does not occur with the wireless Internet. This means reduction of wireless Internet prices. However, it will be difficult to match or reduce the costs to equal those of the wired Internet because wireless bandwidth is a scarce and expensive resource.

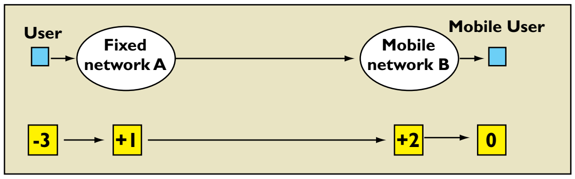


Figure 1. The calling party pays for usage of both the fixed and the mobile networks.

- **The portal role**. Mobile carriers will also have to run their own portals to the wireless Internet world. In this case, it is likely that portals already flourishing on the wired Internet will have an advantage over those of mobile carriers.
- **The application service provider role**. Many new services will appear in the 3G and beyond generations of wireless networks, and the mobile carriers are potential providers of these new services, which may constitute a significant portion of revenue.
- **The content provider role**. Mimicking the fixed Internet, mobile carriers will also have to prepare content for their portals.

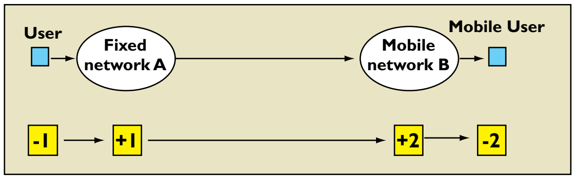


Figure 2. The receiving party pays for usage of the mobile network.

Because the cost of the equipment for the rollout of new services is estimated to be two to four times more than the cost of 2G equipment, a reduced number of carriers is likely to characterize each market. This number is estimated between two and four carriers for each country's market—it has been proved through game theory that the maximum number of carriers that does not impede profitability is four [4].

Carriers associated with telecom operators, especially for data services, will have a relative advantage because in most cases consumers appear to prefer bundled products. Changing traffic patterns are another factor affecting mobile carriers. Increased intra-country mobility, especially within the European Union where a common standard—GSM—is used, increases traffic related to roaming between countries. In some small countries, it is probable that traffic due to roaming will constitute more than half of the exchanged traffic.

Wireless Data Forecast

A somewhat similar situation with that of the early phases of the Internet characterizes today's wireless data scene: low data rates, abbreviated user interfaces, such as those of the Short Message Service (SMS) and the Wireless Application Protocol (WAP), text-centric output, and low-resolution graphics. As the capabilities of wireless networks to deliver data as well as the number of subscribers rise, growth similar to that of the fixed Internet will occur for the wireless Internet as well. The growth in the number of wireless subscribers and data use is becoming evident in recent years (see www.wow-com.com). In addition, data applications, albeit in the primitive form of text messages, are experiencing a similar increase: according to studies, one billion text messages were being exchanged in Europe alone over the wireless medium (see www.sims.berkeley.edu/research/projects/how-much-info-2003/telecom.htm). When carriers deploy higher-speed networks, the usability of wireless Internet will become much more obvious due to the use of data-rich applications. The iMode system in Japan is one example, which we will briefly review here.

A number of capacity-demanding data applications are expected to increase wireless data traffic [5, 6], including video telephony and videoconferencing, Internet browsing, mobile commerce, multimedia messaging and geolocation applications.

Wireless Internet Success Case Study: iMode. Although in its relatively early stages, the wireless Internet already has shown signs of its potential. A good example is the success of NTT DoCoMo's iMode system in Japan that enables users to access Internet services via their cell phones. The iMode system had 29 million subscribers by 2002, increasing by approximately 37,000 subscribers per day (see pressreleasenetwork. com/pr-2002/jan/mainpr1020.htm). The success of iMode has helped DoCoMo to become one of the largest mobile phone carriers worldwide. The iMode system is about to penetrate other markets as well, such as those of Germany, Netherlands, and Spain.

Technological Alternatives and their Economics. There exist a number of candidate technologies for offering data transfer in wireless networks, including: cdma2000, High Data Rate (HDR), Wideband CDMA (WCDMA), and General Packet Radio Service (GPRS). In [6], based on a cost-per-megabyte scenario, it is estimated that CDMA-based technologies have an economic advantage over GPRS due to the limited capacity of the latter. From the CDMA-based technologies, HDR is the most advantageous alternative for supporting data traffic, as it has a two- to three-times cost advantage over cdma2000 1X and WCDMA. This advantage of HDR is due to its optimization for data traffic.

Charging Issues

Mobility Charges. In most cases the price for placing a call through a mobile carrier is significantly higher than that through a fixed telephone carrier. This is because mobile carriers have paid a significant amount of money to acquire spectrum licenses and frequently spend large amounts of money installing new infrastructures. The actual price for a mobile telephone call is not constant but rather depends on factors including the policy of the carrier, the time at which the call is placed, or the user's contract. However, despite the fact that mobile calls cost more than fixed ones, these prices generally follow a declining rate due to the competition between carriers and the concerted effort to make mobile telephony a direct competitor of the traditional fixed telephone carrier.

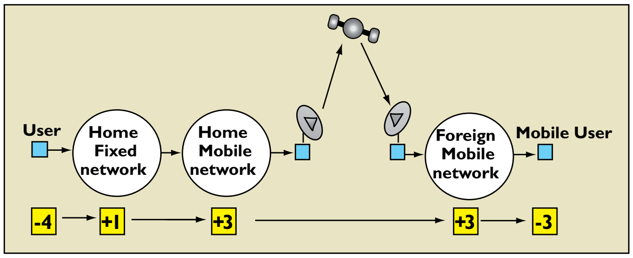


Figure 3. Charges for a call placed to a roaming user.

Another interesting issue regarding the charges for the case of a user who places a call that ends at the network of a mobile carrier. In this situation, there are two approaches:

- *Calling Party Pays (CPP).* This approach, shown in Figure 1, is mostly used in European countries. The caller pays for usage of both the fixed and the mobile networks. Thus, calling a mobile phone from a fixed one is more expensive than a call placed between two fixed telephones. In order to provide fairness to the callers, mobile numbers are preceded by special codes, which let the caller know that the charge for such a call will be higher than that for a call to a fixed telephone.
- *Receiving (called) Party Pays (RPP).* This approach, shown in Figure 2, is mostly used in the U.S. and Canada. The called party pays for usage of the mobile network. Thus, calling a mobile phone from a fixed phone costs the calling party the same amount of money as when the call is placed between two fixed telephones. This approach is driven by the fact that in the U.S. consumers are accustomed to the situation in which local calls are free, thus paying for a call to a mobile phone in the same area would seem incongruous.

Roaming Charges. Figure 3 shows the case of a call placed from a fixed telephone to a user of a mobile carrier, who has moved to the operating area of mobile carrier located in a different country. This situation is known as roaming and imposes relatively high charges to the receiving party. As shown in the figure, an RPP/CPP combination is in effect in roaming situations. This is because it would be unfair to charge the caller for usage of the foreign mobile network since he or she has no way of knowing the called party is roaming to a foreign network. Thus, the cost of the call for the calling party is just the sum of the cost of using the fixed network and the cost of using the home mobile network, meaning the charge for the calling party is what it would be if the called party wasn't roaming. The extra cost of using the foreign mobile network is charged to the called party. This charge is usually much higher than the amount of money is charged to customers of the foreign network, a fact that may make roaming an expensive service.

Billing: Contracts vs. Pre-paid Time. Once the charges for utilizing network resources are summed up, mobile carriers must bill their customers. There exist two main approaches here: contracts and pre-paid billing. A contract is essentially leasing of a connection to the network of the carrier. Users that sign such contracts usually get the mobile handset for free. The mobile operators of course eventually get back the cost of the handset, since the contract forces users to pay a monthly rental charge for their connection irrespective of the fact that they might not use the connection at all. Of course the user is also charged for both calls.

Contracts have the disadvantage of limiting the user to a specific carrier for a certain amount of time. Thus, another approach appeared; that of "pre-paid" time. This approach, first applied by Telecom Portugal (TMN) in 1995, requires users pay in advance for both their handsets and the calls they make. Handsets can be bought from electronics stores and usually include a certain amount of credits, which translate into speaking time (and obviously credits for using other network services, such as SMS). Once the user of the phone has exhausted all the credits, the phone can be recharged via a simple procedure. The pre-paid approach has found wide acceptance in Europe and developing countries [1].

Charging Methods. Here, we describe some methods for charging in mobile networks [2, 3, 7]. Most of these methods have already been proposed for the Internet, but are equally applicable to mobile networks.

Metered Charging. The model charges the subscriber with a monthly fee irrespective of the time spent using the network services. However, most of the time this fee also includes some “free” time of network use. When users have spent this time, they are charged for the extra time using the network. This method is used in 2G networks for charging voice traffic. The way to charge voice calls is quite straightforward: The duration of the call is proportional to the call’s cost. Nevertheless, sometimes charges decrease for increased network usage. Metered charging is well suited to voice calls, which are typically circuit-switched, since the user pays for the period of time the circuit is used. Furthermore, it adds little network overhead and is transparent to customers since it does not require configuration in their devices. However, this model is not suitable for charging the data services expected to be offered by the wireless Internet.

Packet Charging. This method is used for charging in packet-switching networks. It is more suitable for data than metered charging. This is because the user is not charged based on time but rather on the number of packets exchanged with the network. Thus, this method obviously calls for a system able to efficiently count the number of packets belonging to a specific user and produce bills based on these measurements. The disadvantage of packet charging is the fact that its implementation might be difficult and thus costly, since the cost of counting packets for each user might increase the complexity to the network. However, the overhead to subscribers remains minimal as the method is transparent to them.

Expected Capacity Charging. This method involves an agreement between the user and the carrier regarding the amount of network capacity that will be received by the user in case of network congestion; and a charge for that level of service. However, users are not necessarily restricted to the agreed capacity. In cases of low network congestion, a user might receive a higher capacity than the agreed one without additional charge. Nevertheless, the network monitors each user’s excess traffic and when congestion is experienced, this traffic is either rejected or charged for. The advantage of this method is that it enables mobile carriers to achieve more stable long-term capacity planning for their networks. Expected capacity charging is less complex than packet charging both in terms of network and subscriber overhead.

Paris-Metro Charging. In this method, the network provides different traffic classes, with each class being characterized by different capabilities (such as capacity) and hence a different charge. Thus, users can assign traffic classes to their different applications based on the desired performance/cost ratio. Switching between traffic classes might also be initiated by the network itself in order to provide self-adaptivity. Paris-Metro charging is useful for providing network traffic prioritization in wireless data networks. Another advantage of the method is that it provides customers with the ability to control the cost of their network connections. The disadvantages of this method are an increase in the mathematical complexity of the network’s behavior and thus cost of implementation and the fact that users must be familiar with the process of assigning traffic classes to their connections, which introduces some overhead for them.

Content-based Charging. A different approach to the problem of how to charge a customer for utilizing the network is content-based charging. The novelty of this approach is that users are not charged based on usage, but rather on the type of content they access.

Conclusion

Wireless networks constitute an important part of the telecommunications market. The wireless Internet is expected to significantly increase the demand for wireless data services and provide an important new revenue source for wireless telecommunication companies. C

References

1. Beaubrun, R. and Pierre, S. Technological developments and socio-economic issues of wireless mobile communications. *Telematics and Informatics 18* (2001), 143–158.
2. Cushnie, J., Hutchison, D. and Oliver, H. Evolution of Charging and Billing Models for GSM and Future Mobile Internet Services. In Proceedings of QofIS 2000 Symposium (Berlin-Germany, Sept. 2000), 313–323.
3. Franzen, H. *Charging and Pricing in Multi-Service Wireless Networks*. Master Thesis, Department of Microelectronics and Information Technology Royal Institute of Technology of Sweden, 2001.
4. Hugh, M.A., Down, K., Clements, J. and McCarron, M. *Global Wireless Industry Report: Part 1: The Changing Economics of the Wireless Industry*; www.totaltele.com/whitepaper/docs/wireless111600.pdf.
5. Nicopolitidis, P., Papadimitriou, G.I., Obaidat, M.S. and Pomportsis, A.S. Third generation and beyond wireless systems. *Commun. ACM 46*, 8 (Aug. 2003), 120–124.
6. *The Economics of Wireless Mobile Data*. Qualcomm whitepaper; www.qualcomm.com/main/whitepapers/WirelessMobileData.pdf.
7. *Value-Based Billing for Wireless Internet Services, Portal Overview*; www.asiatele.com/internet/wireless.pdf.

Petros Nicopolitidis (petros@csd.auth.gr) is co-author of *Wireless Networks*, Wiley, 2003.
Georgios Papadimitriou (gp@csd.auth.gr) is an assistant professor in the Department of Informatics at Aristotle University of Thessaloniki, Greece.
Mohammad S. Obaidat (Obaidat@monmouth.edu) is a professor of Computer Science at Monmouth University, NJ and the chief editor of the Wiley Publishing’s *International Journal of Communication Systems*.
Andreas S. Pomportsis (apombo@csd.auth.gr) is a professor in the Department of Informatics at Aristotle University of Thessaloniki, Greece.

By Kevin C. Desouza and J. Roberto Evaristo

Managing Knowledge in Distributed Projects

A hybrid approach to knowledge management helps maximize the benefits of the centralized and P2P approaches.

In today's organizations the common unit of work is the project. Turner defines a project as "an endeavor in which human, material, and financial resources are organized in a novel way, to undertake a unique scope of work, for a given specification, within constraints of cost and time, so as to achieve beneficial changes defined by quantitative and qualitative objectives" [6]. Projects have moved from being simple phenomena to manage to more complex entities spanning geographical locations, multiple occurrences, and different organizational affiliations, with IT being the key enabler for the transformation. For instance, a co-located program involves multiple projects running at one location, whereas a distributed project is a single endeavor conducted from multiple locations. Finally, the most complicated scenario is multiple projects conducted at multiple locations. Complexities can be attributed to managing multiple interdependencies across time, space, and projects [4].

In the realm of project management, much of the effort in incorporating technology has involved fostering ubiquitous communication among members while facilitating knowledge exchange. Knowledge generated by projects can be categorized as knowledge in projects, knowledge about projects, and knowledge from projects [1]. Knowledge in projects calls for a close look at insights generated within each individual project, such as schedules, milestones, meeting minutes, and training manuals. Individual project members need to know when, what, how, where, and why something is being done and by whom, with the goal being to promote efficient and effective coordination of activities. From the macro perspective, an organization must have an inventory of all projects under way at any given time, or knowledge about projects. This aids in the planning and controlling of resources to maximize utilities. Knowledge includes employee assignments to projects, return on investment, cost and benefit analysis, deadlines, and customer commitments and expectations. It is common for such knowledge to be generated at regular intervals, such as in weekly, monthly, or bi-monthly reports. Knowledge from projects is a post hoc analysis and audit of key insights generated from carrying out projects. This knowledge is a key determinant of future project success, as it aids organizational learning. These three categories call for distinct roles by IT to enable effective and efficient knowledge management.

Hansen and colleagues divide knowledge management approaches into two categories: codification and personalization [5]. In codification, individual knowledge is amalgamated, put in a cohesive context, and made centrally available to members of the organization via databases and data warehouses. Here, we use a document-to-person approach on the premise that knowledge can be effectively extracted and codified. Personalization is the exact opposite; it recognizes the tacit dimension of knowledge and assumes that knowledge is shared mainly through direct person-to-person contacts. The role of IT is to facilitate communication among members through such tools as email and group support systems.

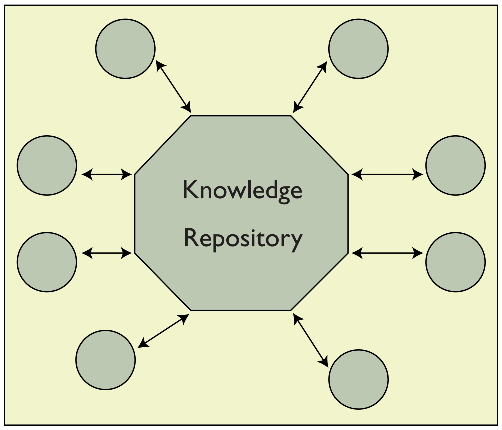


Figure 1. Centralized approach.

A close examination of the codification and personalization approaches led us to draw parallels to two popular models of computing: client-server and peer-to-peer (P2P). The client-server paradigm, wherein a centrally located resource is used by multiple clients to request services for task accomplishment, is common in most distributed computing environments. P2P is a rather recent computing paradigm in which all nodes can take the role of either client or server. A node can request information from any other node, or peer, on the network and also serve content. Due to the centralization of the main resource provider, client-server computing is similar to the codification strategy, whereas the distributed nature of P2P, in which each node owns and makes its resources available to the network, can be viewed as parallel to the personalization strategy. Hence, we term codification and personalization as the centralized and the P2P approaches, respectively, to knowledge management.

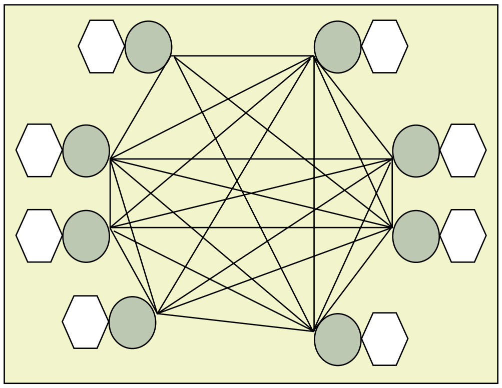


Figure 2. Peer-to-peer approach.

Here, we look at the implications of these approaches on the aggregation, transfer, and sense-making of knowledge in non-collocated work environments. Drawing on the strengths and limitations of each technique, we propose a hybrid model. We begin by comparing the approaches, focusing on three dimensions: sharing, control of, and structuring of knowledge.

Sharing. Many studies report that members of organizations fear that sharing their knowledge with the community at large makes them less valuable to the organization. As such, the idea of contributing to a central repository does not jibe well [2]. In the centralized approach, there are inherent delays between the moment the knowledge is created in the minds of individuals and when it is posted to the repository. Individuals may delay posting not only for gatekeeping purposes but also to allow for confirmation of events, sometimes to the point of irrelevance. This defeats the concept of real-time availability of knowledge, as insights not captured immediately are lost. As individuals are more likely to store the draft notes and working documents of insights on their local repositories than on the main server, this concern is minimized in the P2P approach. Moreover, the P2P approach fosters dialogue among the various agents of the team and develops a spirit of community, as each agent interacts with peers to gain knowledge. Hence socialization and externalization is mandated, which is pivotal for tacit knowledge exchange [3].

Control. The centralized approach detaches the contributor from his or her knowledge. Once posted centrally, the author loses control over knowledge access and usage. In the P2P approach, each member of the organization retains his or her knowledge, as well as explicit control over its visibility. Members are connected to their peers in the organization and can choose what knowledge to share. Since individuals have control over their own knowledge repositories, they are less likely to view sharing of knowledge as a threat to their value.

Structuring. Knowledge contained in the central repository is structured on such dimensions as teams, products, and divisions, enabling faster access times to required elements. This facilitates the use of filtering and categorizing mechanisms for sifting. However, the nature of centrality calls for global filtering and categorizing schemas, which are not optimal in all cases. Setting global thresholds for relevance, accuracy, and other attributes for knowledge may lead to loss of knowledge, as insights considered important for one project may be lost due to filters. The significant costs in categorizing information by appending appropriate key words and metadata to knowledge prior to posting it are borne by everyone, whereas the benefits of better retrieval times are selectively reaped only by the users of knowledge. This asymmetry creates a particularly perverse motivation conundrum: those with potentially the least to gain (knowledge providers) pay the most for it, decreasing the attractiveness of the whole schema.

On the other hand, in the P2P approach, each agent may choose to invoke his or her unique coding and categorization scheme for inputting knowledge. While this allows for flexibility, it makes availability of shared context impossible. Such a system will become ill-structured over time, resulting in cumbersome search and seek times and irrelevant knowledge search results. The ease of effort through which knowledge can be made available to the network can act as a double-edged sword. While it ensures more real-time capture and dissemination of knowledge, it also leads to quality and validation issues. Moreover, when individuals add to the knowledge they download, multiple and inconsistent versions can result.

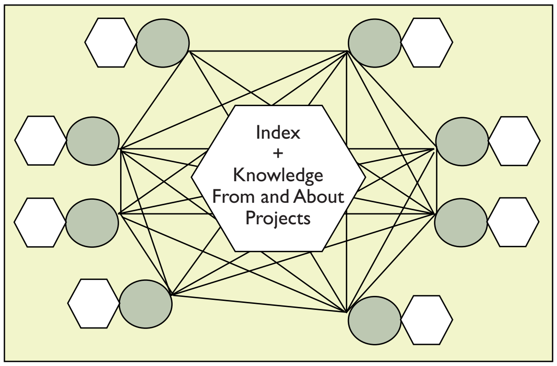


Figure 3. Hybrid approach.

In the centralized approach, members of the organization know where the knowledge objects reside and have the prerequisite tools and knowledge to access them, thus making for ease of use. These characteristics also make the centralized approach useful for storing structured knowledge about and from projects. Requirements for knowledge about projects do not change frequently. As such, having structured approaches for retrieval is facilitated via a centralized approach. But only a small percentage of the organization uses knowledge about projects for budget preparation, staff allocation, and other control purposes. Hence, storing such knowledge in a central repository is of minimal value to the remaining employees, or the majority of the organization. However, the P2P approach is not advisable here due to difficulties in filtering, categorization, and coordination of disparate knowledge sources.

Using the centralized approach to store knowledge from projects helps make lessons learned from past endeavors available to organizational members at large, but centralization is no panacea. Once again it may be difficult to contribute knowledge if appropriate categories do not exist or the repository is not amenable to customization. As the repository is built on the premise that all members of the organization need to access the knowledge base, care is taken to provide a shared context. This entails knowledge contributors making extra effort to ensure their thoughts and insights can be understood by their peers once entered into the knowledge repository. Developing a ranking mechanism to indicate the relevance of results becomes easier owing to the structured categorization of knowledge and the shared context. Thus, transfer of knowledge from the provider to the consumer is improved by the centralized approach.

The U.S. Army is a prime example of an organization taking a centralized approach to knowledge management. The Army Publishing Agency serves as one of the largest publishing houses in the U.S., producing thousands of publications. Today, the Army Knowledge Online (AKO) portal site contains most of these knowledge objects in digital format, giving Army personnel worldwide access to a unified collection of Army knowledge. AKO provides a single entry point into a growing knowledge management system that enables greater sharing among Army communities and enhanced communication between Army personnel and their civilian friends. AKO serves as a virtual home away from home for soldiers stationed around the world. With bases and personnel located worldwide, keeping in touch can be difficult. For security purposes, the Army relies on its own telecommunications systems rather than the local systems running in host nations like Germany, Italy, and the U.K. Most knowledge from and about past endeavors can be found on the AKO and is available to servicemen and women based on their degree of security clearance. The AKO has been highly successful in training new recruits and enabling faster access to lessons learned and protocols.

The solution in the U.S. Army example is unlikely to work in all situations. The recommended option for knowledge in projects is to use the P2P system. Adopting the centralized approach is inefficient, as much of the knowledge in projects is local to one site or to one project and as such, storing such information on a central repository is meaningless and irrelevant to nonmembers of the project. Since knowledge in projects is updated often, in many cases daily, to record details, such as project schedules, milestones, minutes of meetings, and training manuals, will result in overfilling the knowledge repository, high network traffic, and irrelevant search results.

John Deere, a global producer of tractors and other equipment, uses the personalization approach to knowledge management. Deere has established communities of practice (CoPs) for facilitating knowledge exchange. It has recognized hundreds of CoPs, approximately 65 of which use JD MindShare as a technology solution. Knowledge is exchanged within these CoPs via videoconferencing, email, and discussion groups. Using the personalization approach, Deere has experienced the challenges discussed in this article, including the difficulty of sharing knowledge among CoPs due to lack of shared context and varying schemas of knowledge representation.

A Hybrid Approach

To gain the benefits of the centralized and P2P approaches and overcome their limitations, we propose a hybrid model with two components. At the core of the system is the first component: a central repository holding popular knowledge (knowledge about and from projects). This repository serves as an index to the second component: knowledge available by peers (knowledge in projects). By storing knowledge about and from projects in a central repository, we ensure the following:

- Maintenance of a shared context, thus improving means of exploration of knowledge;
- Ease of access, as knowledge about projects is well structured and stored in a central repository;
- Ease of transfer of knowledge from projects throughout the organization;
- Enhanced validity of knowledge from projects, since only validated knowledge makes its way to central storage. Members whose knowledge is stored centrally can also be rewarded for its high value; and
- Easy identification of the source of knowledge about and from projects.

A centrally located index for knowledge in projects, the second component of the central repository, helps foster an efficient coordination mechanism, as this index contains the sources of knowledge in projects. It can serve as the knowledge dictionary to integrate individual knowledge. Common terms and categories can be assigned, along with facilities to serve as an organizational thesaurus.

We assert that knowledge in projects must be exchanged via P2P approaches. Knowledge capture becomes efficient and effective, as each project team can set up its own protocols, build categories, and develop filtering mechanisms. As each project is unique and each team is different, allowing for the establishment of flexible knowledge creation and exchange protocols is pivotal. Such an approach prevents the loss of knowledge considered relevant to one project, site, or stakeholder, since local parameters will not affect the global knowledge of the organization. Moreover, it promotes efficient sharing of knowledge among team members, as the fear of making one's insights available throughout the organization is absent. If an employee is interested in a knowledge object dealing with an ongoing project, he or she can refer to the index and obtain the source. The employee can then request that knowledge directly from the source.

Furthermore, maintenance of local repositories becomes simple, as each team is given the right to purge their local repositories periodically, therefore deciding which knowledge is relevant and useful and which is not. This not only prevents irrelevant and outdated search results, but also helps improve access times. It also helps circumvent the version control problem by constantly updating the repository while purging old and irrelevant knowledge.

Motorola, a major provider of wireless communications, semiconductors, and advanced electronic components, systems, and services, takes an approach to knowledge management that conforms to the hybrid model. Motorola's major business segments/sectors are the personal communication segment (PCS), network systems segment (NSS), commercial government and industrial systems segment (CGISS), semiconductor products segment (SPS), and other product segment (OPS). Motorola has an internal portal where knowledge objects (from and about projects) are stored in a central location for employee access and use across sectors. Examples of these documents include white papers, feature requirements documents, project test reports, and data reports. These documents are posted for the software/hardware features for a given product in a sector. Each employee can customize a portion of the portal, thus fostering the P2P approach. Functionality exists for storing their documents on the central repository with password-protected locations for confidentiality purposes. Passwords are issued by the owner (knowledge provider and document author) of the site, so if an individual first accesses the site he or she will be instructed to "contact the site owner/administrator for password." This guarantees that the material is not openly available, and people trying to access it have a conversation with the original source or owner of the material knowledge, just like a P2P exchange. Search functionalities allow employees to locate knowledge sources; however, a full-fledged indexing system is not present. Thus, Motorola follows a hybrid approach to knowledge management much like the one proposed here.

Conclusion

We have addressed the issues of knowledge management systems in non-collocated environments. We have specifically analyzed two common approaches to knowledge management and made a case for a hybrid model. Our insights have implications for practitioners, as one can consciously choose the right scheme for managing knowledge in projects, about projects, and from projects. For researchers, we have laid the foundation for inquiry into some key issues in distributed knowledge management. Questions of interest include empirically testing the theoretical framework using the different systems approaches. The role of context can be examined both from the global and the local perspectives. Our unique contribution is the analysis of how various system architectures affect knowledge exchange in, about, and from projects. C

References

1. Damm, D. and Schindler, M. Security issues of a knowledge medium for distributed project work. *Intern. J. of Project Management 20* (2002), 37–47.
2. Desouza, K.C. Barriers to effective use of knowledge management systems in software engineering. *Commun. ACM 46,* 1 (Jan. 2003), 99–101.
3. Desouza, K.C. Facilitating tacit knowledge exchange: A humanistic and entertainment approach. *Commun. ACM 46*, 6 (June 2003), 85–88
4. Evaristo, J.R. and Fenema, P. A typology of project management: Emergence and evolution of new forms. *Intern. J. of Project Management 17*, 5 (Oct. 1999), 275–281.
5. Hansen, M.T., Nohira, N., and Tierney, T. What's your strategy for managing knowledge? *Harvard Business Review 77*, 2 (Mar./Apr. 1999), 106–116.
6. Turner, J.R. *The Handbook of Project-Based Management.* McGraw Hill, Maidenhead, 1993.

Kevin C. Desouza (kdesou1@uic.edu) is a doctoral candidate in the Department of Information and Decision Sciences at the University of Illinois at Chicago.
J. Roberto Evaristo (evaristo@uic.edu) is an assistant professor in the Department of Information and Decision Sciences at the University of Illinois at Chicago.

Informational Cascades in IT Adoption

Before adopting new systems and procedures, IT managers should ensure they are not unwittingly following the herd.

As unprecedented IT development continues to produce many investment opportunities, imperfectly informed IT managers keep trying to acquire credible external signals to update their knowledge about each new technology. Such learning processes usually help them reach better IT adoption decisions. However, in some cases, independent of their private information, the majority of IT managers quickly come to the same adoption decision. IT managers who want to think dynamically and strategically must understand the causes of herding and its implications for IT adoption and diffusion.

Information economists have suggested that herd behavior may arise because of informational cascades [1–3], which occur when rational individuals ignore their private information and instead mimic the actions of previous decision makers. Basic cascade models have been tested in laboratory experiments. Empirical evidence of informational cascades has been documented in the realms of financial investment, emerging technology adoption, animal mating behavior, and television programming. In the world of IT, such behavior results in adoption herding. In the uncertain business world, IT managers must independently make technology adoption decisions with incomplete information. In many cases, such decisions are difficult to reverse because of significant technology switching costs. These costs could easily exceed the price of the technology itself when technology adopters have such sunk costs as learning, file creation, and the development or purchase of extrinsic complementary systems.

To avoid being locked into an inferior technology, decision makers may wait to let uncertainties be resolved over time. For example, by taking advantage of the option to wait, they can learn about potentially superior alternative technologies [9]. Also, they can obtain more information about the technology and the IT project as a whole by observing the adoption decisions of those who move earlier. So, with asymmetric information, later IT adopters can benefit from information externality—the spillover of nonpublic information from others' adoption decisions. But information cascading and herding behaviors can occur among later adopters as

By Xiaotong Li

well. Although observing previous adopters' decisions facilitates information conveyance, the information revealed through others' actions may overwhelm decision makers' private information and cause them to make adoption decisions independent of their private information. Then their actions become uninformative to others, and information is poorly aggregated thereafter. If many IT adopters blindly follow others based on the same inference, an informational cascade is triggered and everyone may make the same adoption decision in a very short period.

As a simple example of IT adoption cascade, suppose you and other IT managers are considering adopting one of two competing IT platform technologies. One is a potentially winning technology, the other is inferior, but no one knows for sure which one will win. Each person has a private imperfect signal, or judgment about which platform is better, and each signal has a different precision, or accuracy of judgment. We assume that no one knows the others' private signals, and delaying adoption is costly for everyone. Also assume everyone knows that your signal has the greatest precision. (This assumption is not essential but is merely used, as it is in [12], to simplify the argument.) Since the person with the most precise signal gains least by waiting, you will be the first one to make the adoption decision. Once you choose the platform, all others will adopt what you adopt because their signals are less precise than yours. Because other people's imitation actions become uninformative, further delay only incurs costs, and an IT adoption cascade is triggered.

Now compare this scenario with a full information scenario where everyone shares information freely. Supposing your judgment is correct with a probability of 3/4, and the probability that all persons choose the wrong platform is $1-3/4 = 1/4$. Under a full information scenario, the aggregation of everyone's independent signals may significantly reduce this probability. Thus, an IT adoption cascade resulting from incomplete and asymmetric information may trigger a string of incorrect adoption decisions. This problem is exacerbated if there are strong network effects, and in such cases the superior IT platform may not survive.

	Network Externality	*Information Externality*
What is it?	A type of payoff externality, usually positive for technology market. The value of a technology increases as the number of its users increases. The idea behind it is similar to Metcalfe's Law with which many IT professionals are familiar.	Private information revealed by the actions of previous decision makers. Those who make decisions later can take advantage of the information spillover and update their information accordingly.
How does it cause herding in IT adoption?	Via positive network feedback that makes technology with a larger user base more attractive to potential adopters.	It could cause informational cascades that usually generate massive imitation in IT adoption. Most adopters ignore their private information and blindly follow previous adopters' decisions, which might have been incorrect.
What is its relation to informational cascades?	It does not directly cause informational cascades, but positive network effects make informational cascades less fragile and much more difficult to stop.	It is the primary reason why informational cascades sometimes occur when many imperfectly informed IT managers make technology adoption decisions.
What is its effect on IT adoption and diffusion?	It can create "tippy" markets in which one technology gains overwhelming market share in a very short period.	It is an important information conveyance mechanism in IT diffusion, but if information cascades are triggered, suboptimal IT adoption dynamics and IT overbuilding can occur.
What are its business implications for IT adopters?	It provides rich soil for natural technology monopolies. Unless your careful assessment indicates that a less popular technology will bring you more benefits in the long run, select the technology most people adopted.	You might adopt a wait-and-see strategy to reap its benefits, but be cautious about the possibility of informational cascades. Read the signals carefully.
What are its business implications for IT vendors?	The winner-take-all situation it creates makes IT vendors' choice of compatibility level essential. If war between incompatible technologies is justified, make every effort to achieve the critical mass of customer base, and roll the tide against your rivals.	If your IT product is superior but unknown to potential adopters due to asymmetric information, send them credible signals to separate your product from your competitors' products. Influential adopters' early commitment to your technology could motivate others to follow their decisions.

Table 1. Roles of network externality and information externality in IT adoption.

Informational Cascades

Informational cascading is not the only mechanism that causes herding behavior in IT adoption. Another is that many technology markets are subject to positive network feedback that makes the leading technology grow more dominant [4, 6–8]. The dual mechanisms of informational cascading and positive network feedback are mutually reinforcing in many IT markets. Network effects stem from the efficiency of a compatible product user base and the presence of significant technology switching costs. They usually lead to positive payoff externalities that make an IT adopter's return positively correlated with the number of adopters who have already committed themselves to the same technology.[1] In this sense, those network externalities[2] reward herding by increasing the payoffs of those IT adopters who associate themselves with the majority. Although network externalities give IT adopters more incentives to follow the herd, they do not directly cause informational cas-

[1] Network benefits also include extrinsic benefits. Contributors external to a network sometimes provide these benefits to the network. For example, a sizable network usually leads to larger (potentially more competitive) markets for complementary goods or services.

[2] Some people use network effects and network externalities interchangeably. But technically speaking, network externalities are those network effects not internalized through some mechanisms like contracting [6].

IT ADOPTERS MAY FIND THAT FOLLOWING THE MAJORITY IS THE BEST STRATEGY IF THE BENEFITS OF JOINING THE HERD DOMINATE THE BENEFITS OF LEARNING.

cades. Instead they create “tippy” technology markets in which IT adoption processes are more prone to cascades. Positive network feedback also makes informational cascades that have already formed much more difficult to stop. Many information economists claim informational cascades are usually fragile because information is not efficiently aggregated, and a little bit of new information can quickly reverse the tide [2, 3]. However, in technology markets with network externalities, informational cascades are reinforced by later IT adopters who intentionally agglomerate to reap the benefits of network effects. Table 1 compares the roles of network externalities and information externalities in IT adoption.

Does word-of-mouth learning play an important role?	Yes, the later IT adopters can learn from previous adopters’ actions and the outcome of their actions		No, the later IT adopters can only infer nonpublic information from the actions of previous adopters. Cheap talk is not credible.	
Are the payoff externalities positive or negative?	Positive Payoff Externalities (e.g., Positive network externalities)	Negative Payoff Externalities (e.g., An IT adopter’s return is lowered by others’ adoptions)	Positive Payoff Externalities (e.g., Positive network externalities)	Negative Payoff Externalities (e.g., An IT adopter’s return is lowered by others’ adoptions)
	Scenario I	**Scenario II**	**Scenario III**	**Scenario IV**
How easy is it to form an IT adoption herd?	Relatively easy. Late adopters make more informed decisions, but network effects reward herding.	Not easy. Late adopters make more-informed decisions, and negative payoff externalities work against herding.	Easy. Massive imitation happens more frequently due to information asymmetry and positive feedback.	Not easy. Negative payoff externalities make herding less desirable.
Once formed, is an informational cascade fragile to external shock?	No. Positive network effects make cascades resilient to new information.	Yes. New information could easily overturn an informational cascade.	No. Positive network effects make cascades resilient to new information.	Yes. New information could easily overturn an informational cascade.
Give a type of IT adoption that fits this scenario.	Adopt relatively mature IT to facilitate cooperation with your partners who have adopted the same technology.	Adopt relatively mature IT to compete with rivals.	Adopt emerging IT, anticipating that others will adopt it later to benefit you.	Adopt an emerging IT to preempt your competitors and take the first mover’s advantage.
Common consequence of adoption cascades	Cascades are more likely to lead to the adoption of the right technology.	Cascades are difficult to form.	Cascades occasionally lead to the adoption of the wrong technology.	Cascades could lead to IT overbuilding, but it is usually temporary.

Table 2. Four scenarios of IT adoption.

Information externality gives IT adopters the incentive to learn from previous adopters’ decisions. However, when they adopt a wait-and-see strategy, they should be aware of the possibility and the consequences of informational cascades. It is a tough but important task for them to differentiate among informative signals and signals generated through blind imitation. The situation is more challenging when imperfectly informed IT adopters face potential cascades in technology markets with positive network externalities. IT adopters should understand that technology adoption cascades are likely to happen under this scenario, and they may find that following the majority is the best strategy if the benefits of joining the herd dominate the benefits of learning. The presence of network externalities in some technology markets also changes the risk/return ratio of herding. The rationale for each IT adopter to join a herd is simple: network externalities generously reward agglomeration while significantly reducing the possibility of cascade reversals. Unfortunately, such rational herding at an individual level may not be socially desirable. The socially optimal scenario is for everyone to share private information freely to make the adoption decision with the greatest precision. In an informational cascade, significant private information is lost due to uninformative imitations, which reduces overall decision quality and ultimately negatively affects social welfare. IT adopters may end up selecting the inferior technology/standard because some valuable information is lost during the adoption cascade.

IT vendors should design their business strategies to allow information externalities and network externalities to work for them, not against them. Since competition among incompatible technologies often results in a winner-take-all situation when network effects are strong, IT vendors’ choices of compatibility level become essential. If a standards war among incompatible technologies is justified, they need to make every effort to achieve a critical mass[3] of adopters

[3] Traditional technology diffusion theory suggests that critical mass occurs at the point where the user base of a technology is large enough to self-sustain the technology’s future diffusion. Positive network effects make critical mass even more critical in technology competition.

in the early stages of the war [11]. Information asymmetry, from another perspective, also underscores the importance of technology commitments from early (especially influential) adopters. Thanks to information externality, IT vendors can convey more positive information about their products to potential adopters by securing the key technology commitments of early adopters. Both technology diffusion theory and general equilibrium economic analysis suggest that early adopters tend to be opinion leaders who exert a strong influence on followers [10, 12].

Like IT adopters, vendors also need to be cautious about the possibility of informational cascades. For vendors, an informational cascade is clearly a double-edged sword. IT vendors can happily ride the tide if the cascade is in their favor, but they may be swamped if they go against the tide. They are definitely in an uphill battle if positive network effects are also present. However, there are some effective tools to stop or even reverse the tide if vendors believe their technology is superior when its superiority is unknown to potential adopters due to information asymmetry. In the early stages of a cascade, they can send potential adopters credible signals to separate their products from those of their competitors and to convince adopters that following uninformative herd behavior is inefficient. Some examples of this signaling strategy include advertising campaigns, strategic alliances, and various pricing strategies and product pre-announcements. How to use costly signals to reach separating equilibria has long been discussed in the information economics and marketing literature. Under certain circumstances, ordinary informal conversation (cheap talk) may also achieve the same goal.

Word of Mouth

Until now, our discussion assumes that later IT adopters can only infer nonpublic information from the actions of previous adopters. This assumption, often used in informational cascade research, implicitly suggests that later adopters cannot learn more information from previous adopters' experience through conversation or other social learning, mechanisms. It is consistent with the belief that information obtained through subjective learning, like conversation, usually lacks credibility, and actions speak louder than words. However, both innovation diffusion research and recent information economics research suggest that word-of-mouth learning can significantly affect technology diffusion processes [5, 10]. The conversational learning among these IT adopters can be greatly facilitated by the Internet and other telecommunication networks. (But the Internet may not necessarily reduce the problems of informational cascades; see [3].) Intuitively, inefficient informational cascades, or those leading to the wrong IT platform choices, are less likely to occur when additional information revelation channels exist. Even if an inefficient cascade occurs, it is more likely to be overturned by later adopters who observe the payoffs of previous adopters and gain private information through conversation. Unfortunately, learning by observing previous adopters' payoffs is difficult under many IT adoption circumstances. Precise quantification of IT investment payoffs is notoriously complicated, and the outcomes of many IT adoption projects can be judged only in the long run, while the competitive business environment usually forces managers to make IT adoption decisions in relatively short time frames.

Conversational learning is relatively common in IT adoption, but is such talk credible? Later IT adopters definitely benefit from their conversations with those who have made the adoption, but before taking these early birds' words seriously they should always ask themselves: Is it in the early adopters' best interests to tell me the truth? In a cooperative environment, such as a project development team, people are more than willing to share their IT adoption information and experience. However, managers in most cases face an uncooperative (competitive) environment when they make their IT adoption decisions. To facilitate our discussion of such cases, we describe four IT adoption scenarios in Table 2. We use the credibility of conversation and the sign of payoff externalities as two criteria for classification. Conversational learning plays an important role in scenarios I and II, but its role is downplayed in scenarios III and IV. The technology markets in scenarios I and III exhibit positive payoff externalities, such as network externalities, and the markets in scenarios II and IV exhibit negative payoff externalities, such as pecuniary externalities.

Negative payoff externalities are common in many competitive business environments where a company's return from adopting a technology decreases as more companies adopt the same technology. They usually arise because of downward sloping demand curves, and they punish IT adoption herding and make informational cascades less likely. The recent fiber cable network glut in the U.S. and Europe exemplifies the potential damage caused by negative payoff externalities. Driven by skyrocketing demand for network bandwidth and global connectivity, Qwest, Global Crossing, WorldCom, Williams Communications, and many others raced to build their nationwide backbone fiber optic networks during the late 1990s. This investment herding quickly led to global fiber network overbuilding that eventually resulted in a gigantic supply and demand mismatch. These companies paid a hefty price to learn that herding could be extremely harmful when strong negative payoff externalities exist.

As discussed earlier, network externalities make informational cascades difficult to overturn. Negative payoff externalities do just the opposite; by raising the price tag for blind imitation they can significantly increase the fragility of informational cascades. Table 2 also gives a type of IT adoption that likely fits each scenario. Worth noting is that potential adopters of emerging IT are less likely to learn from others' experience simply because no one has much experience with a new technology; technology vendors might be exceptions, but their opinions are clearly biased. Conversational learning is much more helpful for potential adopters of relatively mature IT, as prior experience is abundant. IT adoption cascades arising under different scenarios also lead to different business implications. Network effects make the IT diffusion process prone to cascades that may occasionally lead to incorrect technology selection. However, if adopters convey nonpublic information through multiple channels, including conversation, they are more likely to adopt the correct technology in an adoption cascade. On the other hand, negative payoff externalities make potential IT adopters reluctant to follow a herd. Even if they imitate others' adoption decisions, the resulting adoption cascade is often fragile because negative payoff externalities will sooner or later manifest themselves.

Conclusion

Although herd behavior is commonly exhibited and observed in IT adoption, IT managers do not have a systematic framework to fully understand its implications. Recent developments in information economics suggest that informational cascades may be one important force behind IT adoption herding. As various degrees of information asymmetry exist in most IT adoption processes, informational cascades may profoundly change the dynamics of IT competition and diffusion. Informational cascades are enabled by information externalities but reinforced by network externalities. Faced with adoption cascades, potential IT adopters should differentiate informative signals from the ones generated by blind imitation. IT vendors need to make every effort to ignite favorable adoption cascades when network effects are strong. If an IT vendor with a superior product falls victim to adoption cascades, it can send potential adopters credible signals to convey positive information about its product.

Besides direct observation of other adopters' actions, other communication channels, like conversation, can convey useful information under some IT adoption scenarios. Such channels, combined with negative payoff externalities, might mitigate the propensity toward IT adoption cascades. Although we discuss IT adoption cascades under four scenarios, we should point out that many real-world IT adoption situations are hybrids. Many IT investments involve both positive network externalities and negative payoff externalities. For example, a compatible technology user base may enhance each adopter's welfare, but downstream market competition may negatively affect each adopter's payoff. With a better understanding of the forces behind adoption cascades, IT managers should feel more comfortable when dealing with complex IT adoption situations under competitive pressure and information asymmetries. C

References

1. Banerjee, A. A simple model of herd behavior. *Q. J. of Economics 107*, 3 (1992), 797–818.
2. Bikhchandani, S., Hirshleifer, D., and Welch, I. A theory of fads, fashion, custom, and cultural change as informational cascades. *J. of Political Economy 100*, 5 (1992), 992–1026.
3. Bikhchandani, S., Hirshleifer, D., and Welch, I. Learning from the behavior of others: Conformity, fads, and informational cascades. *J. of Economic Perspectives 12*, 3 (1998), 151–170.
4. Brynjolfsson, E. and Kemerer, C. Network externalities in microcomputer software: An econometric analysis of the spreadsheet market. *Management Science 42*, 12 (1996), 1627–47.
5. Ellison, G. and Fudenberg, D. Word-of-mouth communication and social learning. *Q. J. of Economics 110*, 1 (1995), 93–125.
6. Farrell, J. and Klemperer, P. Coordination and lock-in: Competition with switching costs and network effects. In *Handbook of Industrial Organization, Volume 3*, M. Armstrong and R.H. Porter, Eds. North Holland, Amsterdam, The Netherlands, 2004.
7. Katz, M. and Shapiro, C. System competition and network effects. *J. of Economic Perspectives 8*, 2 (1994), 93–115.
8. Kauffman, R., McAndrews, J., and Wang, Y. Opening the 'black box' of network externalities in network adoption. *Information Systems Research 11*, 1 (2000), 61–82.
9. Li, X. and Johnson, J. Evaluate IT investment opportunities using real options theory. *Information Resource Management J. 15*, 3 (2002), 32–47.
10. Rogers, E. *Diffusion of Innovations, 4E*. The Free Press, New York, 1995.
11. Shapiro, C. and Varian, H. *Information Rules: A Strategic Guide to Network Economy.* Harvard Business School Press, Boston, MA, 1999.
12. Zhang, J. Strategic delay and the onset of investment cascades. *Rand J. of Economics 28*, 1 (1997), 188–205.

Xiaotong Li (lixi@uah.edu) is an assistant professor of MIS in the College of Administrative Science at the University of Alabama in Huntsville.

TECHNICAL OPINION

John Gerdes, Jr.

The Viability of Supporting Anonymous Employees

Identifying contractual mechanisms to support anonymity in the employment environment.

Anonymity has been used effectively to provide privacy, confidentiality, secrecy, security, neutrality, and to reduce inhibitions. Well-known food critics and bidders at high-end auctions will at times effectively use anonymity to their advantage. Some marketers de-emphasize the association between their products to promote brand separation (consider Toyota's ownership of Lexus). But can anonymity be beneficial in an employment context? This may be the case, as certain positions may be difficult to fill because they impose indirect costs that prospective employees are unwilling to bear. These costs could be due to some stigma associated with the position or employer (for example, accepting an accounting job at an adult entertainment club). Accepting a controversial position or one opposed by some radical group may place the individual, his or her family, and property at risk. It could be that accepting the job may jeopardize future hiring and earning potential. Consequently, the support of anonymous employment could increase the applicant pool, potentially reducing costs and improving the process outcome. Unfortunately, traditional contractual mechanisms do not support anonymity.

Significant obstacles exist for the support of contractual anonymity. Since current business processes do not support anonymous employment, new processes are needed to:

- Verify credentials of unknown, anonymous individuals;
- Confirm anonymous compensation (proving all contractual payments are made);
- Comply with governmental reporting requirements (tax payments, required SEC disclosures), which currently require Social Security numbers;
- Support social equity initiatives (affirmative action, contract set-asides/quotas); and
- Provide for employee accountability/warranty.

PAUL WATSON

Mechanisms to Support Anonymous Employment

Mechanisms needed to support anonymous contractual agreements draw heavily on well-known public-key encryption (PKE) technologies. PKE uses a dual-key encryption scheme—messages encrypted using one key can only be decrypted with the corresponding key. Common uses of PKE include digital signatures, which verify text authorship, and digital certificates, which permit a certificating authority (CA) to certify that a public key belongs to a specific individual (much like a driver's license links a specific DMV number to a particular individual). While these procedures do not support anonymity, blinded versions do. Blind digital signatures hide the contents of what is being signed from the signer—a capability that some digital money schemes use to make digital transactions untraceable. Blind digital certificates do not identify the certificate holder. Rather they indicate that a certain skill has been earned by the owner of the certificate's embedded public key (for example, that the certificate owner has passed the driving test).

Consider a case in which an individual wants to prove she received a degree from Cal Tech, but wants to maintain her anonymity. She supplies the certifying agent (Cal Tech) a blinded (encrypted) token. After validating the claim, the agent digitally signs this blinded token without ever seeing the underlying, unblinded token. Assuming a transitive encryption scheme, the validation applied to the blinded token (the CA's digital signature) is automatically transferred to the cleartext token when it is unblinded by the individual. The resulting signature is identical to the result had the CA signed the unblinded token directly. Combining this signature with the cleartext token generates an untraceable, yet verifiable certificate. Anyone can verify the signature by comparing the cleartext token and digitally signed token using the CA's public key. Unfortunately, this does not provide verifiable ownership of the certificate. Ownership can be proven by incorporating a challenge/response mechanism into the certificate. Rather than a random token, the individual could provide his or her blinded public key. Certificate ownership is proven by demonstrating access to the private key corresponding to this public key (by properly decrypting a message encrypted with the public key).

In the situation concerning the Cal Tech graduate, she receives a digitally signed version of her blinded public key. Cal Tech indicates the degree earned by the key set used to sign the token. Different key sets would be used for each degree offered. Such a certificate has three important characteristics. Anyone can validate it using Cal Tech's public key; ownership can be proven by demonstrating access to the appropriate private key; and the certificate cannot be traced back to the certificate holder, since not even Cal Tech has access to the unblinded token that it signed.

This approach can certify all manner of personal traits and accomplishment. Utilizing multiple key sets, the user can link attributes from different CAs, or keep them unlinked. If the same key set is submitted to multiple CAs, the resulting certificates are tied together through the common embedded token (common public key). Alternatively, the user can submit different public keys to each CA, and selectively combine the certificates to present a limited profile to an employer. This also limits the ability of the third party to build a dossier on the individual. Currently, common information such as Social Security numbers are used to link information from different sources to build a more complete picture of an individual. Using blind certificates eliminates this common identifier, preventing the cross-referencing across information sources.

When it may be necessary to invalidate a certificate, fair blind signatures can be used. In this case a trusted third party is introduced into the credentialing process. This trustee has access to additional information that allows anonymity to be broken if necessary. This ability to associate a certificate with its owner provides the ability to invalidate the blind certificate, but unfortunately, also negates the guarantee of absolute anonymity.

Anonymous Payments. To support an anonymous employment environment means the compensation system must also support anonymity. There are two requirements: it must be able to verify that contractual payments are made, and it must prevent others from masquerading as the worker and claiming payments. Both goals can be accomplished using a modified digital money scheme incorporating blind signatures. To verify payment, the non-traceable digital money can be

Using existing public-key encryption technologies, it is possible to support anonymous employment, an environment that has not been considered in the literature.

blinded with the anonymous employee's public key. Since only the employee knows the corresponding private key, misappropriation of funds is prevented.

Governmental Reporting Requirements. Governmental reporting requirements must be addressed if anonymous employment is to be realized. Two alternatives exist, each supporting different degrees of anonymity. The government could issue Social Security certificates (SSC)—encrypted versions of Social Security numbers (SSN) coupled with appropriate digital signatures. The SSC's validity can be verified by the employer. Incorporating the individual's public key into the SSC would even allow the verification that an individual owns the SSC, currently not possible with SSNs. Required government reports would then use this SSC instead of the SSN. Multiple SSC versions would prevent it from becoming a pseudo-identifier. This approach hides the employee's identity from the employer but not from the government. If the government is unwilling to incur the administrative overhead, an outsourcing model could be employed. A firm could contract with a service bureau or employment agency, which then contracts with the individual doing the work. The employer now has an identifiable entity (the service bureau) to which payments are made. The service bureau is responsible for reporting payments to the government (the IRS) for the anonymous worker, and can shield the worker's identity from the employer. However, under this scenario the employee's anonymity is not guaranteed. The service bureau knows, and subsequently could disclose, the employee's identity.

Anonymous Employees Accountability. Finally, we look to traditional environments to see how employee accountability has been addressed. Workers/employees may guarantee work quality, with payment being withheld or put in escrow until satisfactory performance is demonstrated. Partial payments may be made based on verifiable deliverables. Alternatively, the employee could post a performance bond. Under each scenario, contract negotiations establish obligations and penalties for both parties. A digitally signed contract demonstrates knowledge of, and agreement with, the contract's terms. In the most straightforward case, objective, observable metrics would be used to verify compliance with the contract terms.

Note that anonymous employment reduces "agency costs"—the costs of observing and monitoring work done by an employee while the employee is doing it. Since only outputs can be observed and measured, the employer is forced to precisely define in advance both the deliverables and criteria for accepting the produced output. This does not prevent intermediate deliverables as long as they are well defined beforehand.

Conclusion

Mechanisms exist that mirror traditional employment/contractual mechanisms yet allow individuals to remain anonymous, as has been illustrated here. Using blind certificates it is possible to verify applicant's credentials, while avoiding the creation of an alternate, pseudo-identifier—a problem with the Social Security number today. Fair blind signatures would address cases in which it may be necessary to revoke credentials. To address governmental reporting requirements, the traditional Social Security number could be replaced with a digital Social Security certificate. Such a scheme potentially supports employment anonymity and improves protection against identity theft while providing the government ready access to the individual's identity.

Thus, using existing public-key encryption technologies, it is possible to support anonymous employment, an environment that has not been considered in the literature. It is hoped this work will stimulate future research into the implications of an anonymous work force.

John Gerdes, Jr. (john.gerdes@ucr.edu) is an assistant professor with the A. Gary Anderson Graduate School of Management at the University of California, Riverside.

Coincidental Risks

The story of the Aceville elections has received some attention in the national press, but it is worth considering from a Risks perspective. This column is based on reports by AP (Affiliated Press, “Unusual Election Results in Ohio Town,” 2/30/04) and Rueters (“Losers Question Ohio Election,” 2/30/04). The Aceville, OH, municipal elections last February—the city’s first time using the SWERVE electronic voting system—led to the election of the alphabetically first candidate in all 19 races. This is an astonishing coincidence. Furthermore, every winning candidate, and Measure A, garnered 100% of the votes counted.

“I am extremely gratified by this mandate,” said mayor-elect Neuman E. Alfred, who received 7,215 votes in a town with 7,213 registered voters.

Byron Augusta, CEO of Advanced Automatic Voting Machines (AAVM), which supplied the SWERVE system, denied there was anything suspicious about the coincidence that Alfred was also the AAVM technician in charge of the new voting machines. “We are confident of the integrity of our employees, which is reflected in their unblemished record of electoral success. Reports that Alfred installed undocumented ‘software patches’ the day before the election are completely unfounded. We could prove this to you, except that the machines now contain the software upgrade that Alfred installed the day after the election. Anyhow, our software was once certified tamper-proof by the Federal Election Commission. Any suggestion of hanky-panky is scurrilous and un-American. We were unquestionably the low-cost bidder.”

Ohio Supervisor of Elections Ava Anheuser expressed no surprise that the alphabetically first candidate won every race. “Don’t you believe in coincidence?” she asked. “This is an example of Adam Murphy’s Law: ‘If it’s logically possible, sooner or later it’s bound to happen.’ AAVM downloaded the totals from the voting machines three times. There’s nothing else to recount.”

Rueters reported several voters claimed to have voted for losing candidates, including mayoral candidate Zeke Zebronski, who said, “I know this election was crooked. I voted for myself three times, and still got no votes.” However, the *Aceville Advertiser* conducted an investigation and concluded the complaints were the work of “a small group of out-of-town academic Luddites with a paper fetish,” and “an even smaller group of agitators for ‘alphabetic equality’.” “They should remember that ‘America’ starts and ends with A,” chided *Advertiser* Editor-in-Chief Ada Augusta.

PAUL WATSON

Pundits are divided on whether this election was a statistical fluke, or is the harbinger of a statewide, or even national, trend. But many politicians are taking no chances. The Democratic Party is scrambling to find an A presidential candidate. “We just don’t see how Dean or Kerry can beat Bush in this environment,” said party spokeswoman April Brown. The newly renamed All American Party’s entire Ohio slate has filed to legally change their names, to Aaron Aaren, Abigail Aaren, and so on. “It’s like one big family,” said party secretary Absalom Aaren, “and we expect to do very well in the next election.”

The American Association of Non-Critical Thinkers has pressed for national adoption of the SWERVE system. Spokeswoman Ada Augusta stressed “This is the only available system that guarantees that your vote will be counted, whether you can cast it or not. And it will bring jobs to Aceville.” Measure A provided tax-exempt bond funding for the Aceville Automation Park, which will house new headquarters for both AAVM and the *Advertiser*.

On a lighter note, the American Automobile Association was elected Dog Catcher, even though it wasn’t on the ballot. “This seems to be the first time a write-in candidate has been elected without any write-ins,” said an AAA insider, who spoke on condition of anonymity.

Regular readers of “Inside Risks” realize there is an important distinction between coincidence and causality. The fact that A preceded B does not mean that A caused B. The order of the candidates probably didn’t influence enough voters to change Aceville’s landslide results. However, “out of an abundance of caution,” election officials should have followed the advice of People for Randomized Alphabetic Ballots (PRAY4Ps). Putting names on the ballot in random order preserves faith in the fairness of the election. Of course, it is still possible for a random permutation to leave names in alphabetical order. Wouldn’t *that* be a coincidence? I’d be happy to risk it.

Jim Horning (horning@acm.org) is a member of the American Association for April Foolishness.