



From the Editor in Chief...

Veni, Vidi, Suffragi

I came, I saw, I voted.

Robert E. Filman • *Research Institute for Advanced Computer Science/NASA Ames* • filman@computer.org

“As long as I count the votes, what are you going to do about it?”

—William Marcy “Boss” Tweed

I write this in Menlo Park, California, on 5 October 2003, when it’s still unclear whether the citizens of this state will choose to have the American governor with the best pectoral muscles, despite his seeming to have the same trouble with the multiple meanings of “respect for women” that Bill Clinton had with “sex.” If they do, well, at least our governor will be able to beat up anyone who has been your governor, unless you’re from Minnesota.

Elections are curious, both in a scientific and political science way. Scientists, when performing an experiment that takes five million Boolean data points and finds them equally divided to one part in 10,000, declare the result to be below experimental error, yielding no meaningful result. Scientists know that really small differences don’t mean anything, unless you’re counting a sample size that gives a result like 5 to 4.

Socially (and fortunately limited by the Bill of Rights), we in the US believe in the tyranny of the majority — that, though might doesn’t make right, 50.001 percent of the electorate does. This is in the face of a system with a large experimental error — one that lost four to six million votes from the 2000 Presidential election, an election in which the difference in popular vote between the top-finishing candidates was only half a million votes.

So Where Is the Engineering in All This?

American elections have been, and continue to be, increasingly automated. In the 19th century, they were cast by paper ballot. Since then, we’ve seen mechanical voting machines, punch-card ballots

(genesis of the famous “hanging chad”), optical mark readers, and most recently, “electronic voting.” This consists of giving the voter some form of touch screen and a computer to drive the interface, record the selections, and transmit the results.

Our European readers, who still cast paper ballots, might wonder if this is evidence of an unhealthy American fascination with things artificial. In practice, it’s a reflection of the American political system’s complexity. I have a friend who tells me that when he lived in England, he voted only for his Member of Parliament and city councilor. I, on the other hand, vote for (and this is just off the top of my head): presidential electors, national senators, national representatives, state governor, state senators, state assemblypersons, various state executives like the attorney general and insurance commissioner, county boards of supervisors, various county officials like the assessor and coroner, the city council (which elects its own mayor), various judges at the state and county levels, boards of education at the elementary, high school, community college, and county levels, and commissioners for districts such as fire, sewer, water, open space, and the county harbor. That’s not to mention citizen votes on initiative laws (at the state, county, and city levels), bond issues, tax increases, constitutional amendments, and the occasional recall. And while these offices rarely attract the 135 candidates seeking to replace Gray Davis, it’s still enough to make anyone responsible for counting the votes (in my case, the San Mateo county clerk recorder — itself, an elected position) yearn for as much automation as possible.

So the design of vote-counting systems is an engineering process. The casual engineer might think it easy — after all, it’s just counting, not rocket science. You almost never have to decide if a vote is in Imperial or metric units, solve partial differential equations, or look for critical time

interactions between candidate selections.

However, the problem is harder than that in practice (though still not as hard as rocket science). There are several desirable features of a voting system. It should be open – that is, the process should be inspectable and understandable by human observers. Part of openness is the ability to audit, which implies having redundant ways of checking the results. We want secure systems, ones in which it is difficult to tamper with the process and result. Tampering includes administrators using privileges to change vote tallies, programmers obscuring incorrect behavior in tangled code, and voters employing manipulated credentials to vote repeatedly. We also want anonymity: it should not be possible to tell how any individual voted. This serves not only to prevent coercion, but also to deter against bribing voters.

It worries many people that the current generation of electronic voting devices don't meet these criteria. Most of the new generation of these machines are closed systems, built by private manufacturers that regard their internal workings as trade secrets. The outputs of these systems is vote counts; nothing else tracks the actual votes. Unfortunately, the court system has backed up the trade-secrets claims and allowed such systems. An attempt to consecrate the current poor practices as an IEEE standard has recently been beaten back (www.eff.org/Activism/E-voting/IEEE/), but we can expect to see continuing efforts by voting-machine manufacturers to avoid public scrutiny.

Kohno, Stubblefield, Rubin, and Wallach performed an analysis of one such system, using what seemed to be the leaked source code for a version of Diebold's commercial product.¹ They found an insecure system, easy to cheat by administrators, system developers, and voters, built around a complex environment (Windows CE) and displaying no understanding of cryptography or physical or network threats. To add insult to injury, the authors also criticized the

How to Reach *IC*

Articles

We welcome submissions about Internet application technologies. For detailed instructions and information on peer review, see *IEEE Internet Computing's* author guidelines are available online at www.computer.org/internet/edguide.htm.

Letters to the Editor

Please send letters, including reference to articles in question, via e-mail to internet@computer.org.

Reuse Permission

For permission to reprint an article published in *IC*, contact William J. Hagen, IEEE Copyrights and Trademarks Manager, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08855-1331; w.hagen@ieee.org. Complete information is available at computer.org/permission.htm. To purchase reprints, see computer.org/author/reprint.htm.

IEEE INTERNET COMPUTING

IEEE Computer Society Publications Office
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

EDITOR IN CHIEF

Robert E. Filman • filman@computer.org

ASSOCIATE EDITOR IN CHIEF

Li Gong • li.gong@sun.com

EDITORIAL BOARD

Jean Bacon • jean.bacon@cl.cam.ac.uk

Miroslav Benda • miro@amazon.com

Elisa Bertino • bertino@dsi.unimi.it

Scott Bradner • sob@harvard.edu

Siobhán Clarke • siobhan.clarke@cs.tcd.ie

Fred Douglass • f.douglass@computer.org

Stuart I. Feldman • sif@us.ibm.com

Ian Foster • foster@cs.uchicago.edu

Michael N. Huhns • huhns@sc.edu

Leonard Kleinrock • lk@cs.ucla.edu

G.S. Kuo • gskuo@ieee.nccu.edu.tw

(IEEE Communications Society Liaison)

Doug Lea • dl@altair.cs.oswego.edu

Frank Maurer • maurer@cpsc.ucalgary.ca

Daniel A. Menascé • menascé@cs.gmu.edu

Chris Metz • chmetz@cisco.com

Peter Norvig • peter@norvig.com

Charles J. Petrie • petrie@nrc.stanford.edu
(EIC emeritus)

Krithi Ramamritham • krithi@cse.iitb.ac.in

Ravi Sandhu • sandhu@gmu.edu

Munindar P. Singh • singh@ncsu.edu • (EIC emeritus)

Craig Thompson • cwt@uark.edu

Steve Vinoski • vinoski@ieee.org

Jim Whitehead • ejw@soe.ucsc.edu

STAFF

Lead Editor: Steve Woods

swoods@computer.org

Group Managing Editor: Gene Smarte

Staff Editors: Scott L. Andresen,

Kathy Clark-Fisher, and Jenny Ferrero

Production Editor: Monette Velasco

Magazine Assistant: Hazel Kosky

internet@computer.org

Graphic Artists: Larry Bauer, Alex Torres

Contributing Editors: David Clark,

Greg Goth, Keri Schreiner, Joan Taylor

Publisher: Angela Burgess

Assistant Publisher: Dick Price

Membership/Circulation Marketing Manager:

Georgann Carter

Business Development Manager: Sandy Brown

Advertising Supervisor: Marian Anderson

CS Magazine Operations Committee

Jean Bacon (chair), Thomas J. Bergin, Pradip Bose,

Doris L. Carver, George Cybenko, John C. Dill,

Frank E. Ferrante, Robert E. Filman, Forouzan Golshani,

Rajesh Gupta, Warren Harrison, Mahadev

Satyanarayanan, Nigel Shadbolt, Francis Sullivan

CS Publications Board

Rangachar Kasturi (chair), Jean Bacon,

Laxmi Bhuyan, Mark Christensen,

Thomas Keefe, Deependra Moitra,

Steven L. Tanimoto, Anand Tripathi



quality of the system's software engineering.

Doing Anything To Win (Except Getting the Most Votes)

While it's possible to steal an election by physically stuffing ballot boxes, it's far more efficient to do so by making minor centralized software changes. Could an unscrupulous election official or machine manufacturer produce a manipulated result? Several writers have noted that Chuck Hagel was elected Senator from Nebraska in 1996 — pulling off surprise upsets in both the primary and the election — only a few months after he resigned from the presidency of the company that manufactured the machines that recorded most of the votes.² Senator Hagel initially hid his connection to that company.

What is the right way to make an electronic voting system? After the Florida debacle, Caltech and MIT formed a project to look at the problem. Their most important conclusions were the need to have a physically readable and auditable record of the vote, countable independently of the machine that presents the choices and records the votes. (Further details are available in their report.³) This echoes (though suggests further automation of) Rebecca Mercuri's 2000 University of Pennsylvania dissertation, in which she argued for a voter-checkable paper record of the vote that could be preserved for audit.⁴

If you'd like to be more involved with election engineering, consider the Electronic Freedom Foundation site above or working on project 1583 of the IEEE Standards Coordinating Committee SCC 38 (although the manufac-

turers might persist in trying to keep the committee limited). You could also visit www.verifiedvoting.org — the home of an organization founded by David Dill (of the Stanford Computer Science Department) to press for political requirements for auditable and secure technologies.

Hasta la vista. □

References

1. T. Kohno et al., "Analysis of an Electronic Voting System," white paper, July 2003; <http://avirubin.com/vote.pdf>.
2. T. Hartmann, "If You Want to Win an Election, Just Control the Voting Machines," *The Hill*, 31 Jan 2003.
3. "Voting — What Is, What Could Be," tech. report, Caltech-MIT Voting Technology Project, July 2001; www.vote.caltech.edu/Reports/.
4. R. Mercuri, *Electronic Vote Tabulation Checks and Balances*, doctoral dissertation, Univ. Pennsylvania, 2000; www.notable-software.com/evote.html.

IEEE Internet Computing Welcomes New Editorial Board Members



Craig Thompson is professor and Acxiom Chair in the Computer Science and Computer Engineering Department at the University of Arkansas. He is also cofounder of Object Services and Consulting. Thompson received a B.S. in Mathematics from Stanford and an M.A. and Ph.D. in Computer Science from the University of Texas at Austin. He is a principal investigator on current and recent DoD contracts involving multiagent systems and distributed object middleware. He is a nationally recognized leader in software standards and coauthor of several influential reference architectures. He has a background in software architecture, databases, distributed objects, middleware, agents, hypermedia, natural language, and web technologies. He holds six software patents and has published more than 40 papers in journals and conferences. Thompson is an IEEE Senior Member. He will be writing a column for IC starting in the January/February 2004 issue.



Steve Vinoski is chief engineer of product innovation for IONA Technologies in Waltham, MA, and is an IONA Fellow. He is the creator of IONA's Adaptive Runtime Technology (ART), a fast and flexible distributed computing engine that supports CORBA, J2EE, Web Services, and general middleware integration. Steve has authored or co-authored over 40 publications about distributed computing. He writes the "Toward Integration" middleware column for IEEE Internet Computing. Michi Henning and Steve are the co-authors of "Advanced CORBA Programming with C++," widely acknowledged as "the CORBA bible." He has also co-authored the "Object Interconnections" column for the C/C++ Users Journal (and formerly for SIGS C++ Report) since 1995 with Dr. Douglas C. Schmidt. He has been a member of the IEEE since 1981, and is also an ACM member. Contact him at vinoski@ieee.org.



Jim Whitehead is an assistant professor of computer science at the University of California, Santa Cruz. He is also founder and co-chair of the Web Distributed Authoring and Versioning (WebDAV) working group of the IETF. Current research interests include collaboration support protocols, event notification, collaborative authoring, software configuration management, hypertext, and software evolution. Whitehead received his PhD in information and computer science from the University of California, Irvine, and a BS in electrical engineering from the Rensselaer Polytechnic Institute. Contact him at ejw@cs.ucsc.edu.