BLOG @CACM

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

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

DOI:10.1145/3379491

<http://cacm.acm.org/blogs/blog-cacm>

Coding for Voting

Robin K. Hill explains the ethical responsibility of the computing professional with respect to voting systems.

Robin K. Hill
Voting, Coding,
and the Code
<http://bit.ly/2t5QQe5>
November 27, 2019

Our profession is to be commended for taking steps toward the establishment of computing ethics. They may be baby steps (akin to unstable toddling accompanied by incoherent babble) or perhaps tween steps (akin to headlong running accompanied by giggles, tumbles, and sobs), but steps they are. Let's consider a fundamental process critical to democracy: Voting. The author is inspired by the sesquicentennial, on December 10th, of the passage of the suffrage act in Wyoming, granting women the right to vote and to hold office. Wyoming was a territory at the time, the first known government body to pass general and unconditional (and permanent) female suffrage well before the 19th Amendment granting national suffrage, and entered the Union in 1890 as the first state where women could vote.

What is the responsibility of the computing professional with respect to voting systems? The obvious criteria are accuracy in recording and tallying, reliability in uptime, and security from malicious intervention; all of these are needed for the promotion of trust.

Let's probe deeper. This is not about voting laws, or districts, or methods,² all rich fields of inquiry in their own right. This is about voting procedures as reflected in the design and implementation of software and hardware. Of special concern is voting with electronic assistance. The scope here is the *election system* as defined by the National Academies report⁵ [page 13, footnote 5]—roughly, a technology-based system for collecting, processing, and storing election data. A special issue of this publication³ in October 2004 carried several articles on this subject still worth reading, including the rejection of the SERVE system⁴ that put a stop to the optimistic network-voting plans of the time. This discussion also will refer to sections of the ACM Code of Ethics, as a means of taking the Code out for a spin.¹

Musing on the peculiarities of voting in the abstract suggests a vote is symbolic, discrete, and devoid of connotation; not an act of communication, but an act of declaration, single-shot, unnegotiated, unilateral. Should it exist as an entity; should a vote be preserved somehow? On paper, it does exist as a tally mark. A poll worker could point to it, and even associate it with other descriptions (“the eleventh one” or “the ballot with the bent corner”). A vote may be open to

construal as a first-class artifact (existing on its own, subject to creation, destruction, examination, and modification) that lacks a description or identifier *by design*. First-class objects can be passed as parameters; votes are passed to tallying functions. First-class objects can be compared for equality; that is the salient feature of votes—sameness to or difference from other votes, a stark quality. The voter must give an all-or-nothing choice on each question, no hedging allowed. The hierarchy is flat. All votes count equally, so three votes cast in one polling place should be handled as carefully as thousands from another.

Now to take on the responsibilities of the computing professional, let's outline those at play before coding starts.

First responsibility of the computing professional: To understand why trust in voting is critical. Democracy relies on voting to reveal the collective will of the electorate. In the long view, as in the ethics of care,⁷ background matters and situations cannot be assessed in the moment, but must be viewed in a wider scope in time and place. The National Research Council published a report in 2006 remarking, “...although elections do determine in the short run who will be the next political leaders of a nation (or state or county or city), they play an even greater role in the long run in establishing the foundation for the long-term governance of a society. Absent legitimacy, democratic government, which is derived from the will of the people, has no mandate to govern.”⁶ The report goes on to make the important point that elections must, in

particular, satisfy the losers, preserving the trust that allows them to tolerate the policies of the winners. Code 2.1: “Professionals should be cognizant of any serious negative consequences affecting any stakeholder...” Under American standards, loss of faith in democratic government would be a serious negative consequence.

Second responsibility: To know the criteria for an acceptable election system. These criteria include, as examples, that voting should be easy for everyone; that ballots should present all candidates neutrally; that tallying should be computable by the average person; that audits should be possible. Privacy should be secured under all circumstances (Code 1.6: “Respect privacy,” and 1.7: “Honor Confidentiality”). The result should be dictated by all and only the exact votes cast. Other sources may give somewhat different criteria, but major standards are accepted universally. Life-support systems demand high reliability. Military systems demand high security. Financial transactions demand high accuracy. Voting demands all of those. Security looms over all of the Code, and is explicitly mentioned in 2.9: “Design and implement systems that are robustly and usably secure.” Accuracy, which must also loom over the Code, is not mentioned explicitly. Surely generating wrong answers is the worst transgression of a computing professional. References to quality of work must be intended to cover accuracy or correctness (Code 2.1, 2.2), as well as basic standards of maintainability, efficiency, and so forth, but we might ask whether correctness is a responsibility that transcends these others.

Next responsibility: To interrogate all circumstances, to appreciate the complications, and to acknowledge that unanticipated circumstances will arise. An election system involves many steps of preparation, execution, and resolution, from ballot design and training of poll workers to delivering recounts (and improving procedures for the next election). Complications are rooted in the real-world setting, and the peculiar status of a vote as anonymous but distinct artifact. Code 2.2: “Professional competence starts with technical knowledge and with awareness of the social context in which their work may be deployed.” Our county clerk’s staff will carry a ballot

outside to a car (advance notice requested) for those who cannot easily walk into the polling place. Does that affect the rest of the election system? Code 2.3: “Know and respect existing rules pertaining to professional work.” This could mean the entire local voting code and protocols. If one race is over-voted, does that invalidate the whole ballot? How should a write-in be detected? Under what circumstances is a ballot provisional? If the wind blows a ballot out the window onto a piece of charcoal that marks it, or under a car tire that punches it, after its assignment to a voter, how is it replaced? Anecdotes in electoral research describe exceptions to the notions conscientious voters mark ballots unambiguously, and error-free methods tally those votes.⁸ An election system must accommodate every non-standard circumstance. Voting is a domain where no data point can be dismissed as “in the noise.”

Thus prepared, the computing professional can perform the hardware and software design, coding, and testing. All of the Code applies. Afterward, there are other professional obligations.

Final responsibility of the computing professional: To announce and explain vulnerabilities, errors, quirks, and unknowns, and to suggest solutions. This responsibility is in service to the main one, trust. Demonstrated full disclosure is the best way to instill confidence that, in the face of no disclosure, nothing bad is happening. Code 2.5: “Computing professionals are in a position of trust, and therefore have a special responsibility to provide objective, credible evaluations and testimony to employers, employees, clients, users, and the public.” Code 3.7: “Continual monitoring of how society is using a system will allow the organization or group to remain consistent with their ethical obligations outlined in the Code.”

As a hypothetical, let’s think of a software engineer who notices the tally is incorrect by a small number of votes that exactly offset each other, an error that makes no difference to the tally, nor to the outcomes of any races. Should that flaw be debugged internally? Of course. Should the incident be made public? Yes, because any problem may result in future distortion, which brings this situation under the requirement of Code 1.2: the “obliga-

tion to report any signs of system risks that might result in harm.” It should be made public as a demonstration that votes are prioritized above tallies. The vote is primary; the tally is derivative. This may have unpleasant repercussions to the programmer, but ethical professionals sacrifice themselves before they sacrifice voters.

These responsibilities apply to all who have a hand in American voting, not just computing professionals. Everyone involved should mind Code 2.9: “In cases where misuse or harm are predictable or unavoidable, the best option may be to not implement the system.” The latest National Academies report, among several specific recommendations ranging over many aspects of election systems, recommends the Internet not be used for submitting ballots.⁵

This observer (who claims high interest but shallow expertise) concludes voting turns out to be more complicated than was thought in the early days when electronic procedures were broached. Even though it appears to be counting—the simplest computation of all—voting is a process not amenable to automation except where subordinate to the judgment of election officials. We see the ACM Code of Ethics provides broad but cogent guidance for this computing activity, although we would like to see accuracy incorporated explicitly. **C**

References

1. ACM Code 2018 Task Force. June 22, 2018. ACM Code of Ethics and Professional Conduct. Association for Computing Machinery, <https://www.acm.org/code-of-ethics>.
2. Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A., editors. 2016. Handbook of Computational Social Choice, Cambridge University Press.
3. *Commun. ACM*, 47, 10 (Oct. 2004).
4. Jefferson, D., Rubin, A.D., Simon, B., and Wagner, D. 2004. Analyzing Internet Voting Security. *Commun. ACM*, 47, 10 (Oct. 2004).
5. National Academies of Sciences, Engineering, and Medicine and others. 2018. Securing the Vote: Protecting American Democracy. National Academies Press.
6. National Research Council and others. 2006. Asking the right questions about electronic voting. National Academies Press.
7. Sander-Staudt, M. No publication date given; accessed 24 November 2019. Care Ethics. Internet Encyclopedia of Philosophy and its Authors. ISSN 2161-0002.
8. Wikipedia contributors. 2019. Spoilt vote—Wikipedia. Online; accessed 27 November 2019.

Robin K. Hill is a lecturer in the Department of Computer Science and an affiliate of both the Department of Philosophy and Religious Studies and the Wyoming Institute for Humanities Research at the University of Wyoming. She has been a member of ACM since 1978.