What role should professionals play in policy making?

Galen Gruman, Soft News Editor

With increasing attention paid to software systems by legislators because of cost overruns and questions of safety, software professionals face the question of whether they should be involved, whether as educators or lobbyists, in software-technology policy questions.

Legislators are working on several computing-related issues, including data security (such as the threat from viruses), privacy (such as how government databases may be shared among agencies), intellectual property (such as deciding how much of software may be protected by patents and copyright), safety (such as requiring software engineers to assume responsibility for the safe operation of their programs), and, of course, research funding.

Where are the software professionals?

"I think the people [in the software community], because of the complexity of technical issues, just don't have the time nor the interest in being more politically involved," said Ken Anderson, president of the IEEE Computer Society. "These issues require people willing to be politically and technically interested. Most scientists and engineers don't have that in their orientation," he said. But establishing professional involvement is important because "the US does not have a well-articulated computer policy," Anderson said, "I'd like to see us more proactive in the social and public policy issues."

"People in computing have been very reluctant to stand up and state their needs," said Bryan Kocher, president of the Association for Computing Machinery. But "there is a general feeling throughout the community that we need to be involved in our profession," he said, "No one else is going to speak for our profession."

"I think the participation of computer scientists is very important," said Marc Rotenberg, director of the Computer Professionals for Social Responsibility's Washington, D.C. office. The community has "gone too long without providing input" to policy makers, he said, "I'm surprised that software developers don't get involved."

"Historically, not many people have been willing to come to [Washington] on a permanent or temporary basis who really understand software and what we ought to be doing to improve our ability to create it," said Peter Freeman, director from 1987 until this fall of the Computer and Computational Research Division at the NSF. "People have to be willing to give up two or three years of their regular careers to manage funding programs, plan new programs, argue for budgets, and generally provide effective representation for our community in Washington."

"It's a young field," said Richard DeMillo, Freeman's successor at the NSF, "The senior guys are in their late 50s and early 60s. They are now approaching that stage in their careers where they can start thinking about getting involved." More and more computers scientists are being inducted into the National Academy of Sciences, he said, helping create a "critical mass" of software professionals with ties to policy-making organizations. "A few years ago, there were none there," DeMillo said. The problem is not that the community avoids contributing to professional issues, he said, citing the volunteer work for conferences and journals typically taken on by the research community. The problem is that it takes time for a new profession to become part of the decision-making apparatus, he said.

"The number of people with software training in management positions in government is very small," DeMillo said. Without such people in decision-making positions, it is hard for government agencies to make fundamental changes, he

NEWS **ANALYSIS**

said. He cited Edith Martin, a deputy undersecretary of defense for advanced technologies in the early 1980s, as an example of what can happen when software professionals have management positions. "During her tenure, lots of important things happened. When she left, lots of important things went undone," he said.

"I think some of the people have been very involved," said Larry Druffel, director of the Software Engineering Institute. "I think it's more that we weren't as successful as we might have been," he said, "Perhaps the credibility wasn't there. Like many other professions, we may have made the mistake of using our own language." Furthermore, the computing community has not "had coordinated positions" like other professions have, he said.

"Over the last five to 10 years, the number of people with computer backgrounds [in government] has dramatically increased," said Fred Wood, an investigator at the congressional Office of Technology Assessment. "But there ought to be a lot more communication," he said. While Congress is aware of privacy, information access, surveillance, and security issues, areas like reliability have "not received that much attention," he said. Wood said most computing professionals work for private industry, which does not encourage political involvement. But "if you don't play in the political arena, you'll get shortchanged."

The formal-methods community has been particularly active. In Britain, it helped convince the Defence Ministry this summer to require the use of formal methods and mathematical verification in the development of safety-critical systems and to require minimum training levels for designers of safety-critical systems. Formal methodists also held a workshop that brought together researchers and government officials from the US, Britain, and Canada to make the case for formal methods' usefulness.

Professional groups. The professional organizations have little voice in Congress, although both the ACM and Computer Society are considering establishing apparatuses to handle policy issues, according to their presidents. "The main problem right now with the Computer Society profile is that we have been very passive," Anderson said. "We haven't had many people volunteer to focus on public policy," he said, but "I think we're building a cadre of people who are interested ... and can establish the right kind of communication." However, "we do have [society] leadership with high responsibility in government" who help raise the profession's concerns, he said.

The Computer Society's activities have focused on educational activities such as curriculum development.

The ACM has increased its activities, Kocher said, citing the increased coverage in *Communications of the ACM* on social policy, a $500,000 contribution to help produce the PBS documentary series *The Information Age*, and the Core of Computer Science Education report on computing curricula.

The IEEE has a lobbying office, which has promoted the institute's views on privacy, data security, and tax treatment of engineers. It also sponsors two engineers as congressional fellows to advise legislators about technical issues.

Much of the political work has been left to small, issues-oriented groups like the Computer Professionals for Social Responsibility. The CPSR's Rotenberg would like to see that change: "We don't need a lobbying office for more funding. We need a commitment to look at social issues," he said. "I'd think that the ACM and IEEE would rather be up front on the issues rather than be taken to task for not getting involved when they knew there was a problem," he said. "The professional organizations need the standing committees and mechanisms in place to examine these issues and contribute to them," Rotenberg said, "For

the CPSR, that's a full-time job, but I think the ACM and IEEE can do more."

What role should the societies play?

"I don't know," Kocher said. "That's an unresolved question. The ACM should have a leadership role in requesting things that the profession needs to address the art of software engineering and computer science. If we can't get that basic stuff, we can't have much influence in the end application of that technology," he said. ACM's tack has been to "make the profession attractive to bright kids in high school," Kocher said.

"Our involvement should be based on the commitment and interest of our members," Anderson said. "In the US, we should be involved in education and retention of the workforce. Our programs need to encompass secondary schools (precollege) through the PhD," he said. The society should also look to see what efforts would benefit members outside the US, Anderson said. These members comprise 25 percent of the society.

Federal agencies. In the federal agencies, individuals have forged relationships across the government to coordinate software policy. For example, the National Science Foundation and the Defense Dept.'s Advanced Research Projects Agency — two primary software-research funders — coordinate their research areas.

These agencies also participate in the Federal Coordinating Committee on Science, Engineering, and Technology, whose members include representatives from the National Institute of Standards and Technology, National Aeronautics and Space Administration, Energy Dept., military services, Central Intelligence Agency, National Security Agency, Strategic Defense Initiative Organization, National Institutes of Health, Health and Human Services Dept., and State Dept.

Former NSF official Freeman said he was pleased that several respected software scientists have made the move to Washington to join federal agencies. These people include William Wulf at the NSF's Computer and Information Science and Engineering Directorate, Richard DeMillo of Purdue University at the NSF's computing division, and Bill Scherlis of Carnegie Mellon University at DARPA's Information Science and Technology Office. For years, several well-known scientists have also contributed to advisory panels like the Defense Dept.'s Defense Science Board.

Changes are also taking place within federal agencies charged with promoting software researchers. The NSF, which relies on peer review for most of its research grants, is taking on a greater role in pursuing innovative programs, NSF officials like Freeman and DeMillo said. The balance in NSF research funding has changed from "90-percent reactive and 10-percent proactive to 80-percent reactive and 20-percent proactive," said Bruce Barnes, the NSF's deputy division director for information, robotics, and intelligent systems. "That's twice as much proactive effort," he said.

For example, the NSF gave its program managers the authority this fall to award small grants for "high-risk" exploratory research areas without peer-reviewer approval, DeMillo said. The peer-review process is appropriate for a basic-research funder like the NSF, he said, "but you also want a source of innovation."

Despite such programs, "there are big areas of the technology that are not being covered by anyone," DeMillo said; for example, no federal agency is funding basic research in security and privacy research.

We don't need a lobbying office for more funding. We need a commitment to look at social issues, CPSR's Rotenberg said.

Policy agenda. What should be on the agenda? There are many areas to choose from, but most seem to fit in six categories:

- Building foundations. This includes basic research, resolving software engineering's role, establishing widely accepted standards of what constitutes a software professional (perhaps certification or accreditation), and codified engineering practices (the equivalent of mathematical and architectural tables).
- Safety and reliability. This includes requiring developers to address these issues during specification and development and having developers take responsibility for their systems' performance.
- Education and training. This includes expanding the number of students studying science and engineering (including attracting more minorities and women), improving educational facilities and faculty abilities, and developing widely accepted curricula.
- Information access. This includes making government databases available to the public, restricting how enforcement agencies may trade data about individuals, giving people the right to examine and correct information stored about them, and assuring people's privacy.
- Intellectual property. This includes revising copyright and patent laws to reward innovation without creating monopolies over the underlying ideas and without discouraging standardization.
- Trade. This includes examining subsidies of local software industries, removing barriers to the trade of ideas and products, and considering the effects of wage differences between developing and developed nations as software development becomes more global.