

# To See Ourselves as Others See Us

Neville Holmes, University of Tasmania

**B**y tradition, each year *Computer's* January issue describes the wonders that computing professionals and their industry anticipate most keenly. Doubtless our professional duty lies in designing and implementing these wonders. However, we have an equally important duty to consider the effect of these innovations on the world at large and to be aware of how our dreams and our efforts to realize them will shape others' opinions of us.

To maintain an awareness of the world outside our profession, we can choose any of several methods. For example, I subscribe to and diligently scan *The Economist* for a rational, economocratic view of the world. For a more democratic view, I read the *Guardian Weekly*, which includes sections from *Le Monde* and the *Washington Post*.

Last year, a column by editor and journalist Simon Caulkin, "Make Computers Our Servants, Not Our Masters," appeared in the *Guardian Weekly* (Sydney edition, 13 Sept., p. 16). The article gave a succinct and very critical evaluation of the role computers play in the business world.

## DIGITAL BINGES

Caulkin opened by observing of economic growth that "the great Internet binge has given us levels of improvement that haven't been seen since, er, the 1930s." According to him, the



**To mature and evolve professionally, we must listen to our critics and work closely with our clients.**

notorious dot-com bubble is not, alas, a singular phenomenon.

Caulkin reported that Paul Strassmann, at one time the Pentagon's information chief, has observed eight "build-and-scrap" cycles in computing investment since 1946, with each cycle greatly surpassing the previous one both in absolute value and percentage of overall business investment. Strassmann projects the next binge to be two and one-half times more expensive than the last, and suggests somewhat optimistically that we won't have enough money to pay for it.

Caulkin saw these binges as "sustained by some kind of arms race in that, short of any profitability justification, companies invest simply to keep up with rivals." It's easy to blame these binges on management and absolve our profession from any responsibility. On the other hand, company managers usually employ computing professionals to advise them on technical matters, which strongly implies that professionals have supplied bad advice—or none at all.

## The PC binge

I remember best the personal computer binge, which I presume qualifies as one of Strassmann's cycles. The professional failing there was twofold. First, professionals didn't cater to their end users. Second, they succumbed to the PC hyperbole.

By the time the PC binge began, computing people in the business and government worlds were organized into politically powerful data processing departments notorious for their large budgets and disdain for end users. They saw themselves as answerable only to

upper management's needs. These professionals cherished their status as big-project people. When cheap personal computers with useful full-screen generic applications like spreadsheets became available, the DP departments ignored them. However, end-user departments bought lots of PCs so that their workers could run these applications. The low cost of a single PC allowed DP departments to avoid prohibitions on large equipment purchases, and gave individual workers what they really needed.

The typical DP department then made the mistake of trying to suppress these initiatives, rather than moving to provide a comparable level of personal service with their existing equipment. When suppression proved impossible, and the benefits of desktop PCs became obvious as they quickly spread, the tradition-bound DP departments strove to bring the proliferating PCs under their control—with two quite natural consequences.

First, PC hardware and software,  
*Continued on page 142*

## The Profession

Continued from page 144

having evolved from hobbyist origins, proved much harder to use and maintain than professional mainframe software driving dumb terminals would have been. Thus, the already expensive DP departments adopted the only rarely successful Help Desk approach in a vain attempt to retain centralized computing control.

Second, as more and more end users in business and government organizations did increasing amounts of work on growing numbers of PCs, they needed to exchange and share data more frequently. This led to the widespread adoption of local area networks, with file and print servers attached, to link the PCs.

### An alternative

The time had clearly arrived for computers that provided services *directly* to end users. The PC binge could have been turned into a more gentle evolution if DP departments, or their mainframe or minicomputer suppliers, had developed generic end-user applications to run on full-screen dumb terminals. Today, thin-client computing revisits this approach—one that the profession still fails to take seriously enough.

Computing professionals must bear responsibility for missing this obvious solution. Or, if they did see it, they must bear responsibility for failing to convince their management or their suppliers that existing systems could provide the needed end-user services in an evolutionary way.

The difficulty here is that computing professionals tend to see themselves only as system engineers responsible for analyzing their client's needs and designing and implementing a system that satisfies those needs. But a professional also has a responsibility to management. All too often, the computing professional's manager remains relatively uninformed about trends in digital technology, but, being higher on the pecking order, hesitates to ask for or take advice. Professionals have an important responsibility to inform management when potential technical

problems or opportunities arise, even if their counsel might fall on deaf ears.

Too few computing professionals combine both technical and management skills. This problem can be tackled by making computing professionals knowledgeable about management sci-

**Frankenstein systems are a failure that has resulted from complexity worship—a long-standing and widespread failure.**

ence, sensitive to their responsibility to management, and interested in assuming management responsibilities themselves. Computing courses should include instruction in professional ethics and basic management. Computing students with an interest in management should be allowed to minor in the discipline, or could be advised to study management after finishing their first degree.

### FRANKENSTEIN SYSTEMS

Caulkin also observes that highly complex computer-based systems become ends in themselves, just as Frankenstein's monster assumed a life of its own. Computing professionals could think of many examples, but Caulkin gives special attention to airline reservation systems and call-center systems.

### Voracious child

A consortium of airline companies developed Sabre, the "granddaddy" of airline reservation systems. This system proved highly profitable because, rather than pricing seats according to their cost, it allowed them to be priced according to the fluctuating demand.

Sabre's success, and that of similar systems, led to spin-offs as independent businesses. According to Caulkin, Sabre now levies what is "effectively an impost of \$10-\$12 on every ticket its former parents sell." Such overhead makes the larger airlines vulnerable to low-cost flat-fare competitors, a vul-

nerability that I imagine the September 11 tragedy has only magnified. "Not surprisingly, the [major] carriers are squirming uncomfortably, trying to remove themselves from a hook of their own creation," Caulkin writes.

### Problematic "solution"

Call-center systems offer another example of complex systems developed "as a 'solution' to service cost or quality issues." They sprang from the idea that a sufficiently complex computer-based system would let low-cost, low-skilled outsourced workers respond to customers from a variety of businesses. These workers could also sell raffle tickets to the general public during slack time—the high-tech version of door-to-door salesmen.

As Caulkin sees it, "Requiring IT-scripted call-centre staff to sort out service problems often just increases the number of irate calls, leading to apparent demand for yet another information factory." My own experience with call centers makes me think they're not all bad, at least for simple inquiries. However, the financial systems that the larger banks and credit card companies use demonstrate a more blatant example of complexity's cost. These institutions' rapidly increasing charges and persistently high interest rates are attracting much criticism.

### Monstrous complexity

I view Frankenstein systems as a professional failure that has resulted from complexity worship. It's a long-standing and widespread failure. Two of the finest articles ever published in *Computer*—"A Plea for Lean Software" by Niklaus Wirth (Feb. 1995, pp. 64-68) and "Why Software Jewels Are Rare" by David Lorge Parnas (Feb. 1996, pp. 57-60)—eloquently argue the case for simplicity in software, but seem to have gone unheeded.

Often complexity seems unavoidable, even desirable. Management demands an all-encompassing system to reduce labor costs. Marketing demands continual accretion of advertisable capa-

bilities. Clients want a system that does more than their last system, and more than their competitors' systems. Professionals seek the safety of shared responsibility within a large team, and complexity justifies large teams.

Simplicity does not come merely from doing everything on the computer. Computing professionals should seek simplicity in *all* aspects of a system. Often, the software can be made simpler by designing independent machine capabilities that the user integrates into adaptable work patterns. Typically, users will be more effective if their work can be designed to give them a feeling of achievement, and if they can develop skills. Education of computing professionals would ideally include the study of human psychology and even of cognitive science generally.

### INFORMATION FACTORIES

Binges and complexities can arise from the isolation of computing professionals in a kind of organizational priesthood. Caulkin repeatedly depicts "information factories"—a phrase he attributes to Thomas Johnson—as detrimental to business. Information factories extend computing isolation by bringing specialist computer users into what is really a new kind of DP department.

### Overpaid underproducers

In manufacturing, Caulkin observes, digital technology "has greatly increased the productivity of direct labor. But these gains are generously offset by an invisible IT overhead." He also comments that information-factory workers often outnumber and earn much higher salaries than "real ones."

Caulkin's use of the term "real" conveys the view that computer professionals aren't *really* workers, that they don't produce real value. He emphasizes that "Much computerization is what environmentalists call 'end-of-pipe'—it is applied to an existing process to make it manageable. But this is a solution only in the sense that liposuction is a solution to obesity."

### An alternative

Caulkin proceeds to cite Toyota as a success story showing that industry functions better and more profitably without an information factory. At Toyota, he writes, the "work-flow is designed so that the manufacturing processes themselves carry all the infor-

**We cannot earn respect unless we are prepared to give it—to users as much as to critics.**

mation that those operating it need to know." Caulkin thus implies that Toyota's computing professionals work with the "real" workers, as part of the team on the factory floor. The traditional DP department and the modern information factory, which not only separate professionals and their clients but also the management of both groups, discourage such close cooperation.

To be most effective, computing professionals should work as closely as possible with the workers they support. Were computing professionals to be integrated into their client departments and teams, as at Toyota, they would be better able to design and support the needed systems, and to adapt them to new requirements.

### Our service profession

In some ways the computing profession can be seen as a secondary profession, providing services to other professions and occupations. Thus, the wider IEEE's general publications often have articles that might well have been published in *Computer*. For example, last September's *Proceedings* focused on soft computing, and *Spectrum* carried pieces on IBM versus Intel, computer-controlled cars, Lego Mindstorms, and Apple OS X.

Organizational dispersion of computing professionals should have benefits beyond improved systems. Skills in computer use could be built up more effectively throughout a company,

making technical advice and support more easily available. Fellow workers would more naturally share credit and responsibility for successes, and would be more concerned with ensuring that success and preventing failure.

**A** mature profession, like a mature person, values and seeks to attain a good reputation. A reputation ultimately lies not only in evaluating our own actions, but also in observing and taking seriously how other people see and value what we do. As Robert Burns once wrote, this would "frae mony a blunder free us, and foolish notion."

When critics like Caulkin find fault with the use of computers, we must look for the cause in our professional work. When Caulkin observes that the New Economy was "sustainable only as long as the illusion lasted that computers did anything genuinely new," we as professionals should not scoff, but should look instead for truth in his opinion. Truth is there to be found, and a mature profession will find and be guided by it.

We cannot earn respect unless we are prepared to give it—to users as much as to critics. When the public believes that the "New Economy was a gigantic computer-aided pyramid scheme," then the reputation of our profession suffers as much as the speculators' pockets.

The education of professionals, in the classroom or on the job, must stress that computers are only justified if they help the people who use them and the people whose lives are affected by them. It is our professional responsibility to ensure that computers are used to actually help people. ■

*Neville Holmes is an honorary research associate and a contracted lecturer at the University of Tasmania's School of Computing. Contact him at [neville.holmes@utas.edu.au](mailto:neville.holmes@utas.edu.au).*