

The Future of the Computing Profession

➔ **Neville Holmes**, *University of Tasmania*



The computing profession isn't flourishing in parallel with computing in government, business, and the community.

Ten years ago, "The Profession" began bimonthly publication. Shortly thereafter, it was promoted to its current monthly prominence at the tail end of *Computer*, where it replaced the excellent "Binary Critic" column.

There are several reasons why this column has persisted. One is the friendly skill, wisdom, patience, and tolerance of *Computer's* professional staff in Los Alamitos, California, particularly Jim Sanders and Judi Prow, who have often coped with unusual demands. A recent example is last November's essay, in which I insisted on having a saltire printed with a tilde superposed—not the simple matter it should be.

Another is the generosity of the many authors who have contributed essays over the years. Not only do they add to the column a variety and depth not available to any one author, they also save me a lot of work.

BACKGROUND

The opportunity to discuss and write about professional issues has had a special attraction to me for some time. In preparing for this anniversary essay, I sought out two papers I had written on such matters back in 1974. They were based on my professional work since I joined

IBM Australia in 1959 as a systems engineer in the field in Melbourne, as an assignee in the IBM World Trade Systems Centre in Poughkeepsie, New York, and as a systems engineer at IBM Australia's short-lived Systems Development Institute in Canberra. Given that I hadn't read them since well before I took on this column, I was pleased that they were still quite plausible in their context.

The first paper, "Data Engineering," I presented at a conference of the Institution of Engineers, Australia, in May 1974 (eprints.utas.edu.au/9851; hereinafter P1). It proposed "that a branch of engineering, perhaps best called 'data engineering,' be established by creating a College of Data Engineering in the Institution of Engineers, Australia, and that formal tertiary education in the discipline be sponsored by the Institution." Its scope and flavor were illustrated, and course topics for tertiary study outlined. The proposal wasn't adopted.

The second paper, entitled "The Social Implications of the Australian Computer Society," appeared in *The Australian Computer Journal*, the Australian Computer Society's monthly journal (eprints.utas.edu.au/9832; hereinafter P2). It assumed the existence of a data engineering or similar profession and concluded by recommending eight rather unusual

objectives. The referee declared his disagreement with them but strongly supported my right to have them published. The paper was later awarded the ACS's medal for the best paper of 1974, but I was told privately that debate within the awarding committee was heated and the result very close. The ACS didn't adopt any of the recommendations, as far as I could see.

Both these papers stressed the importance of the distinction drawn between the terms *data* and *information* by the international standard vocabulary (*The Profession*, 2001 May, pp.94-96). Both papers made observations that I've often repeated in this column.

THE PROFESSION

The inaugural essay from 10 years ago, "Fashioning a Foundation for the Computing Profession," seems rather naïve in retrospect, as it depicts an idealistic future for the column, one that hasn't yet happened. However, it made two persistent points that, interestingly enough, echo the sentiments of P2. First, it's essential to distinguish between crafts, trades, and professions.

Second, although a body of knowledge and skills defines a profession like computing, the health of a profes-

Continued on page 86

sion “depends on its members taking an interest in issues outside its body of knowledge and skills.” The three classes of such issues relate to the profession’s distinctiveness, external community constraints, and effect on the external community.

The profession’s effect on the external community comes through its products and procedures. Computing professionals are responsible for the proliferating 3D visual products and gambling websites, for example, and we should be concerned at recent reports that overdosing is hazardous (tinyurl.com/2agj858). Computing professionals are responsible for large projects using digital technology and should be concerned that so many fall far short of meeting their objectives (tinyurl.com/26y96q2).

essentially to the digital computer, which is truly distinctive since all other learned professions focus on relevant techniques and procedures and are quite happy to exploit whatever tools are available and useful from time to time. For the computing profession, this distinction is a mistake.

Second, all the other professions by now make routine use of digital computers and other digital technology, and their professional education includes training in their use. Thus the computing profession is, in effect, just an unfocused subprofession.

THE PROBLEM

The computing profession faces the problem that it’s becoming increasingly irrelevant. Specialist pro-

developed countries weren’t much different. Consider, therefore, the small number of people who have since graduated from some tertiary computing course or other compared to the number of people who belong to professional computing bodies.

This is the real problem. Computers proliferate. So do people who learn how to apply them effectively in their day-to-day work. They even learn how to apply them when they’re in school. Therefore, people who merely know a bit more than other people about how to use computers don’t properly belong to a profession, just as people who have done a first aid course don’t thereby belong to the medical profession.

THE SOLUTION

Members of learned professions are primarily people who help other people. The proliferation of computers—and the spread of skill in using them—means that most computer users don’t need professional help because their problems are small and technical rather than grand and organizational.

The people who do grand and organizational things with grand computers and networks are themselves professionals. Therefore, computing should properly be a secondary profession because computing professionals are of most help to workers in other professions. For this to come about, computing education at the professional level must change substantially. More than three years ago, I offered a suggestion for how this change could be accomplished (*The Profession*, Jan. 2007, pp. 114-116).

Stage one

Sudden substantial change is impractical. To reform the computing profession, the first change is obvious.

All other learned professions have technicians. Lawyers have clerks, mechanical engineers have machinists, architects have builders, scientists have laboratory assistants,

The computing profession faces the problem that it’s becoming increasingly irrelevant.

The external community’s effect on the computing profession comes through educational policies, government regulation, and business practices. For example, poor educational policies lead to falling literacy and numeracy levels that eventually lead to lower capabilities in newly graduated computing professionals and technicians. Changes in government regulation that are doomed to failure can be proposed because the changes in the software systems involved can’t be successfully made in the time allowed. Business practices like those involved in the recent global financial chaos hinged on the use of software by computing professionals, known as quants, to instigate financial transactions.

The most important issue, though, is the distinctiveness of the computing profession. Two important aspects of this issue were emphasized or foreshadowed in both P1 and P2.

First, the computing profession usually identifies itself as linked

professions are developing that focus on some aspect or application of digital technology. For example, digital communications is becoming the focus of a branch of engineering, as is software development

It seems that many computing graduates go into jobs where they work as technicians performing routine administrative and maintenance work on computers within organizations, or provide advice and other help to members of the public wrestling with their hardware or software. If they’re successful, they go on to become managers.

Such people don’t see themselves as computing professionals, so they quite reasonably see affiliation with a professional computing body as pointless. One citation in P2 reports a 1968 survey as finding that “between 35% and 50% of those engaged full time in systems and programming work in Australia are members of the [Australian Computer] society.” Presumably, the figures for other

doctors have pathologists, and so on. In general, professionals do the investigation, design, planning, and supervision, and technicians do the groundwork and implementation.

The computing profession needs to recognize that programmers are technicians, programming is a distinctive skill, programming techniques and tools are continually changing, and professional attention to programming details distracts from the investigation, design, and supervision that should be the professional's responsibility.

The ability to code programs well has two bases—talent and training. The talent can be tested for and was well understood in the early days of computing (*The Profession*, Nov. 2004, pp. 118-120). People who don't have the talent should be discouraged from becoming programmers.

The training should come in two stages, at the trade and craft levels. Programmers should be trained at technical training institutes, just as bricklayers, electricians, mechanics, and welders are. Effective technical training is quite different in style from academic education, and programmers properly selected for their talent and trained in this way will be much more competent on average than most academic graduates. Further, they will be more easily retrained for special fields and for the use of new operating systems, programming techniques, and other innovations.

Stage two

Once this reform is in place, computing and other professionals can focus on solving their professional problems and leave the implementation of their designs to programmers and other technicians. Professional education in computing will need some practical work in programming, however, so that graduates can properly exploit the capabilities of trained programmers, but this will be supplementary rather than central.

Schooling

A trade like programming needs a craft background. Also, because computers have become ubiquitous, some programming should be taught in primary and secondary school alongside woodwork, cooking, and the like. This would be helped if the popular operating systems replaced their menu and click facilities with a scripting and command-line facility. The menu/click style of use drives the user along paths chosen by the application or operating system developer. A scripting and command-line facility would let the learner compose and develop ways to control the computer and could lead naturally into coding programs.


When a student controls the computer, the emphasis is on constructing and achieving objectives, and the result is a realistic appreciation of the proper role and potential of the digital technology in society. All too often the computer in the classroom is used as a minder, more personal

and so more effective than television, to fill in the student's time rather than allowing learning to take place by, for example, teaching basic skills (*The Profession*, Mar. 2008, pp. 102-104; Sept. 2009, pp. 102-104).

That the preceding proposals might be adopted in toto is doubtless wishful thinking. But surely something must be done to restore buoyancy to the computing profession. What then is wrong with these proposals? What would be better? **■**

Neville Holmes is an honorary research associate at the University of Tasmania's School of Computing and Information Systems. Contact him at neville.holmes@utas.edu.au.

cn Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.



COMPUTING THEN

Learn about computing history
and the people who shaped it.

[http://computingnow.
computer.org/ct](http://computingnow.computer.org/ct)