The Computing Profession and Higher Education

Neville Holmes, University of Tasmania

The computing profession needs to focus on working with other professions.

Recently, I attended a school meeting at which participants discussed a proposal for amalgamating the computing and information systems schools. Given my strong and unconventional views on the topic, I kept silent then, but have since felt compelled to draft this essay.

Central to my views is the secondary nature of the computing profession, a concept I exposed briefly in "Jobs, Trades, Skills, and the Profession" (*Computer*, Sept. 2002, pp. 104, 102-103; *Computers and People*, Wiley, 2006, pp. 128-132), where I argued that a distinction should be made between professions and trades. Many implications for higher education spring from seeing computing as a secondary profession.

PROFESSIONS

Most professions deal with people, directly or indirectly. Doctors, lawyers, and educators mainly deal with people directly. Engineers, architects, and agronomists mostly deal with things people use. In general, professions deal with both people and the things they use in various and varying proportions.

The computing profession is different. Computing professionals deal with people and their data. Data are not things, but abstractions that represent things. Data use provides human civilization's foundation, and computers and other digital technologies have merely amplified this use.

Computers are tools

People nowadays use computers personally, just as they use cars and lawnmowers. Computing professionals use computers just as doctors use stethoscopes and sphygmomanometers, and as mechanical engineers use lathes and drills. The professionalism lies in the use of the tool rather than in the tool itself. Where then does the computing profession fit in with other professions?

Doctors now have computers on their desks. They enter data they gather from patients and use it—as well as data sent to their computer over the Internet from specialists like pathologists and tomographers—to deal with the patient's problems.

Mechanical engineers don't use machinery such as lathes and drills professionally and directly; they work with technicians who do this. But their design work thoroughly depends on computers, as does their planning and production management. Computing professionals, on the other hand, are concerned with facilitating other professionals' use of data. In this sense, the profession is secondary.

Some distinctions must be made at this point. There is, or should be, a branch of engineering that deals specifically with the design and manufacture of computers and other digital equipment. As the field is already popularly called *digital* technology, this profession would best be called *digital engineering* and is distinct from the computing profession, which concerns itself with people and their data rather than with digital equipment.

Software engineering

Concerned with the design and construction of software, software engineering has been seen as distinct from the computing profession's main body. Partly, this is because program coding is a major part of this engineering branch—which I consider a mistake. Program coding should be viewed as a craft or trade, and programming technicians should be qualified by intensive vocational training. Programmers would thus be to software engineers what electricians are to electrical engineers and mechanics are to mechanical engineers.

Program engineering then becomes more professional in the classical sense. It would also become more sparse and so less justifiable as a distinct branch of engineering. It could simply be reabsorbed into the general computing profession, but this is arguably inappropriate because software engineers focus on programs, whereas computing professionals focus on people and their data.

A better approach would be to subsume software engineering in digital engineering, somewhat as the Australian Computer Society has done in setting up a Computer Systems and Software Engineering Board alongside boards for Computer Science and Information Systems.

Continued on page 114

Continued from page 116

Simply computing

With software engineering out of the way, the distinction between computer science and information systems becomes easier to see. Computer science is about the manipulation of data. Information systems is about using data to inform people. The two fields are closely related, as indicated by their popular collapse under cover of that ugly initialism, IT, so often used to qualify terms such as *industry, profession*, and *worker*. I have even seen the phrase *IT technology* used, reminiscent of Microsoft's *NT Technology*—Google gives almost a million hits for it. Simply *computing* says it all.

Governments and the media nowadays seem to prefer the initialism ICT for Information and Communication Technologies, once a trademark of the British firm International Computers and Tabulators. The use of ICT arises because the field of telecommunications has become increasingly digital and the source of ever more consumer products.

A merger of digital engineering and communications engineering, or at least of its device-level practice, seems to be taking place informally and perhaps should be made formal. Certainly, the use of the Internet and other telecommunications industry products looms large in both computer science and information systems.

Computing has a strong secondary relationship to electrical and electronics engineering, as *IEEE Spectrum*, the IEEE's house magazine, clearly shows. For example, the front cover of *Spectrum*'s October 2006 issue features a picture of a robotic surgeon and the headline "How Electronic Medical Records Could Save Lives." Inside, it's much the same: for example, the backpage column describes what happens "When Good Clicks Go Bad" (Paul McFedries, p. 52). An article subtitled "How to cross the cultural divide when working overseas" ("Shaman, Bless This Lab," Susan Karlin, pp. 43-45) even quotes a consultant as saying, "In engineering circles, no matter where you are in the world, people are very passionate about their preferred operating systems" and so "I sometimes started a presentation by saying, 'How many people use Linux versus Microsoft?' I'd hear geeky giggles from the audience."

THE UNIVERSITY

When I studied engineering, the faculty rules required us to arrive on the first day with a slide rule, though a Curta calculator would be allowed if we could afford it. For statistics, we learned to use Odhner calculators, and for surveying we were taught to use 7-figure log tables.

The kind of coverage computing gets from the media only increases the disinclination to consider computing as an attractive profession.

Our campus had an electronic digital computer, but I didn't find out about it until after I graduated and joined the computing industry because physicists guarded it behind an impressive portal labeled *School of Natural Philosophy*. On some other campuses, the first such computer had electrical engineers for guardians. Thus, early computer science subjects and courses were taught by physicists or engineers and had a strong mathematical flavor.

When commercial computers became common, government and industry typically used them for billing, accounting, sales analysis, and similar commercial applications. The larger computer manufacturers even used separate architectures for commercial work, often with only decimal arithmetic, and for scientific work, usually with only binary arithmetic.

Eventually, university business faculties realized that their graduates would benefit from exposure to the commercial and administrative use of computers. They aimed to be—and be seen as—independent of the scientific computing academics, who were considered incompatible with business and commerce, which forced adoption of a distinctive title: "Schools of Information Systems" thus sprang up within business faculties.

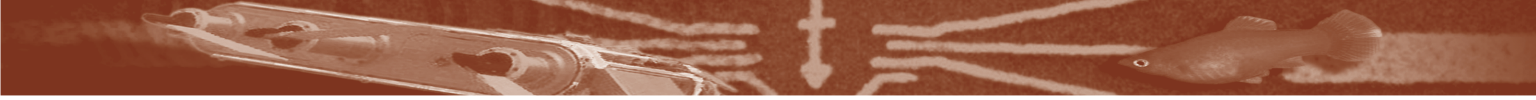
In universities, administrative computers were kept separate from their academic counterparts. University administrators saw their task as running the university, the academics' task as teaching and research—again, a thorough incompatibility. By the 1980s, many universities had three mainframe computers: one administrative, one in business, and one in engineering or physics. All remained very much at arm's length from each other.

Since then, the successive development and widespread adoption of minicomputers, desktop computers, and personal computers has dramatically increased the number of computers on campus, while their overall computing power has grown astronomically. Now, most departments, administrative or academic, even if tiny, have their own computer laboratory and accompanying technicians.

IMPLICATIONS

Academic teaching of computing suffers in two ways from personal computers' ubiquity. Because computing is taught in most major university schools, as a discipline it is losing respect. Teaching computing to undergraduates is seen as simply a matter of teaching them to use the software particular to a specific discipline, something any academic in that discipline can do quite capably. This results in both teachers and students being shackled to their software, causing both groups to lose interest in moving beyond the software, and losing the capability to do so as well.

Because computing is taught or used throughout schools, typically with an emphasis on entertainment and as a substitute for the library, students lose interest in computing as a future occupation or profession. The kind of coverage computing gets from the media—which emphasize the jargon-overloaded world of nerds and geeks



on the one hand and the fluctuating prospects for employment in a world determined by the stock market on the other—only increases the disinclination of both students and their parents to consider computing as an attractive profession. This lack of interest has led to a decline in enrollments and thus to pressure for computing school amalgamation and the focus on narrow popular applications such as Web site and videogame design and implementation.

Defining a new image

We need instead a clearly defined, stable, distinctive, and widely accepted portrayal of the computing profession, one that can only come about through the actions of higher education, supported by professional institutes. Defining the computing profession as a secondary profession that deals with data use brings clarity. Excluding the aspects of the profession that focus on digital technology and program coding brings stability. Emphasizing the role of computing professionals as partners of other professionals brings distinctiveness. But widespread acceptance of computing as a profession will rely on universities changing what they teach and how they teach it, with the support of all professional institutes.

Computing must be seen within the university as a partnership with other disciplines, in both teaching and research. The main step is to bring both computer science and information systems out of the normal faculty structure, together with mathematics. After all, mathematics is simply the subset of the theory and application of computing that deals with quantitative data. Given responsibility for all undergraduate teaching of basic computing—for example, of programming and statistics—a separate computing faculty would acquire a large enough student load to justify its existence.

Partnership for a better tomorrow

There would be great political resistance to such a move, but the advantages of uniform content, standards, and equipment throughout the university should be plain. At the same time, the danger of isolating computing from practical context must be countered by giving priority to partnerships in research and specialist teaching.

With partnerships established at the teaching level, it becomes possible to establish it for students. Those taking computing degrees need to select a major course of study in another profession with a view to specializing in computing for that discipline. The taking of double degrees should be seen as an extension of this.

Much professional education requires undertaking practical projects in the final year, if not earlier. Partnership in these projects would benefit all participants.

Bringing computing out of the cupboard at universities is a major undertaking and requires a complete change in the outlook of all concerned. There are many variations to be considered, many possibilities beyond those I've outlined, and many implications for earlier education.

What should be clear, especially to those involved in teaching computing, is that, unless such a change is undertaken, the number of computing degrees granted will decline. More importantly, the full potential of computing within the teaching and practice of all disciplines will not be realized, in either sense of the word. ■

Neville Holmes is an honorary research associate at the University of Tasmania's School of Computing. Contact him at neville.holmes@utas.edu.au. Details of citations in this essay, and links to further material, are at www.comp.utas.edu.au/users/nholmes/prfsn.