Professionalism in the Computing Field

Aaron Finerman
SUNY at Stony Brook

The term professional means different things to different people; nevertheless, there are certain general technical and social standards normally associated with a professional. Further, the term is more generally applied to the practitioner rather than to the researcher. But within the rather broad definition specified, the computing practitioner is, as yet, not regarded as a professional.

Each of the four types of institutions—academic, industry, government, and the professional society—that educate, employ, regulate, and mold the practitioner contributes to the "nonprofessional" status of the computing practitioner. The roles of these institutions are examined, various shortcomings are noted, and recommended changes are suggested.

In the last analysis, professional status is not bestowed; it is earned. However, universities and industry, specifically, can make certain improvements to help the computing practitioner achieve professional status.

Key Words and Phrases: professional aspects, educational programs, industry attitudes, professional societies, licensing and certification

CR Categories: 1.50, 2.10, 2.2

Adapted from an invited address, ACM Southeast Region Conference, Nashville, Tennessee, April 18, 1974. Author's address: Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11790.

There is an immediate difficulty in initiating a discussion on "professionalism" since the very term is not well defined and has different meaning to different people.

To some, a professional is one who gets paid for performing services; this holds especially in sports where we speak of the professional tennis player, for example, versus the amateur.

To some, a professional is one who has established a reputation a client can rely upon with some confidence; this holds especially in medicine and law.

To some, a professional is one who is supercompetent; Stone [1] states, ". . . professionalism now means supercompetent. Jack Nicklaus, Peggy Fleming, Henry Aaron, Billie Jean King are professionals. Charlie Brown is not."

Let me attempt a more detailed definition of a professional. Broad as the term "professional" has become in common parlance, I intend this definition more specifically for those in the scientific, engineering, administrative, and other computer-related fields. (Much of the following material has been synthesized from [1] and [2].)

Technical Standards

—A professional field has a definable body of knowledge with an attendant broad level of education, training, and experience necessary to acquire the requisite knowledge.

—The field has established standards for professional competence and behavior to qualify the individual for admission to the professional group.

—Society recognizes that the professional performs a valuable service, one which has a tangible effect on society; the form of recognition usually is payment for services rendered.

—The professional has a high degree of individual responsibility, a willingness to take initiatives, and a sense of obligation to identify client (and employer) needs as well as client (and employer) wants.

—The professional has a sense of responsibility for the quality of the work performed, a high self-imposed standard of workmanship to maintain that quality, and joy and pride in performing that work.

Social Standards

—The profession is aware of the effects that services performed have on society and has a sense of responsibility for serving the public good.

—The professional has an understanding of the interaction and relationship between facts and values (or technology and values).

The last point is especially difficult to fulfill because values are not always well defined; technology causes revolutionary changes, while aggregate social values change slowly with time. Indeed, I would hope that some values never change: justice, truth, love, and the value of the life of a human being.

The above definition may still be fuzzy, but in general, the first set establishes certain technical standards dealing with quality, competence, behavior, and the like, while the second set goes beyond this to assert that the professional has a primary orientation to serve the community interest rather than his individual interest. In recent years, scientists and technologists have been held increasingly accountable for the effects of their science and technology on society. This may not be fair; another viewpoint, one held by many scientists, is that it is up to society to protect the public good, that scientists are more interested in the nature of their work than in the ultimate end to which their contributions are put. But fair or not, events in our time have made clear that the technologist is held responsible not only for the weapons of war but for the tools of peace—not only for the anti-ballistic missile but for the computerized data bank (see [3]). A professional must understand and accept this responsibility.

In the above discussion, I have used the terms "technology" and "science" quite interchangeably. However, let me now differentiate between the two (see [2] and [4]).

Science concerns the search for fundamental knowledge about the nature of our environment and our functioning in that environment, using mechanisms for formulating and testing hypotheses and for discovering new truths and theories.

Technology concerns the application of scientific knowledge to affect the human and social condition, and undergoes substantial changes with time, as new principles, new materials, and new processes are invented.

Usually, we speak of "applied science," a bridge between technology and science. Most fields are a continuum from science to applied science to technology. However, the generally accepted viewpoint (at least the one I adopt) of the term "professional" is more often applied to the technologist—the practitioner—rather than to the scientist—the researcher. The practitioner affects society more immediately than the scientist; the airline reservation system is much more visible than the research on time sharing which made possible that system. Accordingly, society requires and demands protection to assure that the practitioner is serving the public good. We now regard such fields as teaching, law, medicine, engineering, architecture, and accounting as professional, and we enforce certain standards to protect the public. Although the work of both the doctor and the researcher is built on a body of scientific knowledge, the practicing doctor faces the malpractice suit; the researcher in the background does not.

If you can accept these background thoughts, let me now try to look at professionalism in the computing field. The point I shall make is that the computing practitioner is not yet regarded as a professional. (Carlson [5] expresses the point succinctly: "The overall impression is that computing technology does not yet provide a foundation for professional practice. The experience to date indicates that only limited areas of subject matter have reached a stage where two or more experts can refer to, and agree upon, a first set of principles in their development or evauation of practical designs, applications, or operations.")

This attitude is largely determined by four types of institutions—*academic, industry, government, and the professional society*—that all play dominant roles in educating, employing, regulating, and molding the practitioner. Each is responsible, to a greater or lesser degree, as to whether the practitioner is regarded and accepted as a professional.

Academic

Just as I have excluded scientists at one end of the professional spectrum, I shall exclude technicians at the other end. This automatically eliminates two-year colleges from the subsequent discussion. Two-year community colleges turn out a most useful product, but not a professional, at least not according to the previous definition. Just as engineering requires the draftsman, computing requires the computer operator and the data expeditor—the technician or paraprofessional.

My main concern with this exclusion is that some two-year colleges, in fact, do believe that their role is to produce the professional—the inexperienced but "intern" programmer or systems designer. To my mind, the professional, whether in engineering or computing, requires the intellectual orientation and the broad education normally associated with the bachelor degree. It is a fact of life that career paths in computing, as in other fields, rarely lead from the paraprofessional to the professional. Within this framework, two-year colleges serve two very useful functions: producing the well trained technician or serving as the academic bridge between secondary schools and universities.

But, if I have been gentle with community colleges, let me compensate by being somewhat less gentle with universities. In my opinion, most university programs in computing are inherently and incestuously fashioned to yield a product in the academic's own image—the scientist, the researcher, the student who will continue for the doctorate. The very name of most computing programs—"computer science," rather than "technology" or "engineering" (with some exceptions)—indicates primary emphasis on the scientific aspects of computing, in contrast to the engineering emphasis on the pragmatic, the intuitive, and the empirical. (Many European universities have adopted the term "informatics" to denote a broader milieu than either "computer" or "science"; some United States universities have recently instituted programs in "information systems"; I use the term "computing" to indicate a broader scope than "computer," although many disagree with my rationale.)

Don't get me wrong. I firmly believe that a primary purpose of education is to educate rather than to train, to teach the "why" rather than the "how," to prepare the student voyager for the unknown rather than the known. However, with all due respect, I doubt that the typical university program in computing achieves this. Instead, it produces a rather shallow product with little depth: in values, in human interaction, or even in the many disciplines to which computing technology is applied; albeit one who is overly specialized in computing science: in mechanical languages, machine organization, compiler design, and the theory of computing. In short, the university computing program rarely educates the whole man, which is what we really need in these troubled times, but gives us, if you will, the "technician scientist"; one with little ability to synthesize, to relate facts and values, or with little understanding of the scientist's role in society, or even with little background in the individual's role in a democracy—all of these rarely possessed, but precious attributes. (I note in passing that most scientific programs share this shortcoming, as expounded most clearly in a recent study at the Massachusetts Institute of Technology [6], yet that fact makes it no more excusable for the computing practitioner, who must interface with and have some understanding of a multitude of other disciplines.)

Further, if a secondary function of university education is to prepare the graduate for gainful employment, as indeed it must be, the typical educational program rates little better. The graduate is useful in the industrial core of computing—languages, hardware, compilers, and the like—but he is much less useful as an interface with the user community or as a programmer of complex application systems, areas in which we in the computing field have a poor record indeed.

Glaser [7] comments: "What we desperately need are professional problem solvers, trained to practice computing—the kind of problem solvers who fully understand the burdens of the innovator. . . . Unless, and until, we develop a significant capacity to produce pragmatic problem solvers, we shall continue to turn out individuals for whom the 'agent of change' role is extremely painful and frustrating." He might have added that we equally desperately need people with common sense. Unfortunately, our higher education programs do not impart the common sense approach. If the graduate comes out of the program with common sense, it's probably because he went in with common sense, not because of anything he was taught at the university.

Hamming (as quoted in [8]) has referred to these graduates as "idiot savants" or "computerniks [a term I have usually regarded as one of praise] with little idea of what the theory they have studied is good for." Oettinger (as quoted in [9]) notes that most computing science departments "are just getting out from under the influence of competing engineering and mathematics departments, and they are too busy teaching Simon-pure courses in their struggle for academic recognition to pay serious time and attention to the applied work necessary to educate programmers and analysts for the real world."

Perhaps these descriptions are overly pernicious and pejorative, and perhaps I unfairly overstate the situation to emphasize the point. Actually, I have little desire to downgrade United States universities that have developed a unique, often imitated educational institution, or to engage in polemics with their computing science departments that have developed remarkable academic credentials in so short a time period. The point is, I urgently hope our institutions of higher education would develop a more professional product—by my definition, a broadly educated person with breadth and perception, and common sense and useful skills—rather than a narrowly educated person who is perceived as a copy of the educator. Obviously, if our field is to advance, we must have the underlying theory, developed by the academic and the researcher; but is is just that we do not need so many "academic copies" at the undergraduate level. (To be fair, some four-year colleges, with a liberal arts tradition, do try to produce a less specialized undergraduate; and some universities (see [6]) are reviewing their curricula with the aim of reversing the undergraduate trend to over-specialization.)

Industry

Industry also contributes little glory to the professional image of the computing practitioner, who is typically regarded as an intense mystic producing those dreadful operation systems, or as a tunnel-visioned technician who must be furnished with detailed program layouts by the user. Industry justifies the claim that the user can do just as well (or rather, as poorly) as the professional programmer by pointing to the catastrophic failures in integrated management systems, airline reservation systems, or other large application systems that by their failure have forced some companies out of business and created chaos in others (see [9] for example). It has become fashionable to claim that computing has become too important to be left in the hands of the computing people. In *Up the Organization*, Townsend (as quoted in [1]) observes: "Most of the computer technicians are complicators, not simplifiers. They're trying to make it look tough. Not easy. They're building a mystique, a priesthood, their own mumbo-jumbo to keep you from knowing what they—and you—are doing."

I note that at times these industry images of the computing practitioner intentionally set up a convenient strawman in order to disguise the poor judgment and the incompetence often exhibited by the user and by management. Seegmuller [10] notes that many organizations want a computing superman "so that he may solve their organizational mess (which was brought into the open when they set out for automation)." Whether misguided or justified, however, these attitudes are prevalent. They exist, and thus call into question the value of the services performed by the computing practitioner—by any definition a fundamental attribute of the professional. The end result is that the computing practitioner is not regarded as a true professional—at least not in the same sense as the engineer or the business administrator—and has little opportunity to advance up the corporate ladder to positions requiring a broad management perspective.

Perhaps some of this is to be expected. Computing, still an infant field, has suddenly and explosively become one of the primary technological constituents of our society. It was not too long ago that little difference existed, at least in the public eye, between draftsman and engineer, or even between barber and doctor. We can expect that, as our profession matures, the professional will become not only vital for the functioning of industry but sufficiently adept to be recognized for a valuable service performed.

However, industry does little to improve the situation and, indeed, often exacerbates it. For too long, industry has tried to cut corners (for the sake of expediency and money) by bringing in people who do not possess a broad level of education or experience. Almost any warm body from other walks of life has been considered suitable material for a career in computing. This has been referred to as the migration phenomenon: "bakers, truck drivers, butchers, secretaries, salesmen" [10], and the like, diffusing into the computing field. In fact, more often than not, computing practitioners have played the role of proselytizers with evangelical enthusiasm; rejecting any attempts to limit entry into the field, they have invited in all candidates, no matter how unqualified, with gusto. And even today, the largest percentage of computing practitioners are products of manufacturer courses or vocational schools (see [11]) or in some other way stuffed with hasty and far from professional computing skills. They possess neither breadth nor depth. They are our "instant experts," all too often working for a computing manager "promoted" from a managerial position in some other field, and himself lacking prerequisite experience, technical background, or even intellectual interest in computing technology. (This "electric bulb management theory" assumes that a manager is a manager, whether of an electric bulb or of a computing enterprise.)

Obviously, this industry attitude is not helped by the university attitude toward the undergraduate computing program already discussed. Often, the type of computing practitioner previously described by Townsend is, indeed, the result. Often, the practitioner exhibits "a simple lack of feeling for people, and an inability to go beyond a disastrous machine concept of humanity," as discussed by Balk [12]. Glaser [7] further notes, somewhat in the same vein: "Professionals help people. They do not worship things. Yet it seems to me that far too many computer people are unduly impressed with their machines . . . And the more zealous of them act as though the use of the computer is inevitably good; whereas its use often can be questionable, or even precarious."

Again let me note that I do not wish to overstate the situation, or to malign our industrial institutions, or to risk the friendship of many dedicated computing colleagues. However, I do believe that improvements to change attitudes which breed a machine concept of humanity are in order. The university undergraduate program must take on greater breadth in the humanities, the social sciences, the application areas of computing, and the "dirty" engineering and business data processing aspects as well as the "clean" scientific aspects. This would all help to impart a broader understanding of the human and humane approach to offset the machine concept, of the interfacing disciplines as well as the computing discipline, and of the pragmatic aspects of computing to complement the theoretical. Industry must insist on the same high standards of education, experience, and "professionalism" in computing as in other professions. To do less will be to perpetuate the current less-than-professional image of the computing field and of its practitioners.

Government

Government inevitably enters this potpourri since the state is responsible for defining the legal aspects of a profession, most visibly by licensing. Most fields normally considered professional—teaching, engineering, law, medicine—already have licensing requirements. The justification, of course, is that the license indicates that certain standards of competence and behavior have been satisfied by the possessor; thereby, the public is protected from the quack, the dishonest, the thief. Presumably, in computing, licensing would help protect against practitioners who might falsify financial records, obtain unauthorized access to or actually destroy information, appropriate software, or make wildly exaggerated claims of ability (see [2]). The last, you will recognize, is similar to a disease already rampant among "honest" computing practitioners: optimistic enthusiasm or enthusiastic optimism, the symptoms being, "can do; the technology is there" (only to discover later that there is no "there" there). Note, I make no claim that licensing is an absolute deterrent to any of these things—recent events in other professions demonstrate clearly that professional license notwithstanding, incompetence and unethical behavior are facts of life; indeed, at times, the licensing process itself imparts a false sense of security and makes the unwary even more so. Licensing, however, does indicate some a priori attempt to weed out the incompetent and the dishonest, and contains the inherent ex post facto threat to "delicense."

Licensing in computing is not yet a reality, and in a legal sense computing is not yet accepted as a profession. Most arguments against licensing have been based on the newness of the field. For example, it is noted that certain prerequisites of licensing—standards of competence, a code of ethics, job classification—have not yet been sufficiently identified and defined. Another argument is that it is still too early to limit entry into the field. To me, the validity of the last argument is most doubtful at best. A Groschism for the computing practitioners who could not find jobs during the 1969 to 1971 recession was, "Let them eat the cake they saved up" (presumably accomplished during the lush days when a people-shortage in computing was the conventional wisdom).

I believe that licensing in computing is simply a matter of time. When it does come, it will have some effect on the other institutions. For example, some computing science departments today do not teach Fortran since "Fortran is the language of the streets, graffiti upon the bathroom walls." (This philosophy was eloquently articulated by Alan Perlis in a talk at the ACM Special Interest Group on Computer Science Education 1973 Annual Symposium.) If anything, Cobol is viewed with even greater disdain, and data processing, in general, is regarded as a technician's trade by many computing science departments. Licensing requirements will create some pressures to teach for the license, to offer these "dirty" languages and applications. I think these pressures will be to the good. Obviously, if overdone, they may subvert the intent of higher education; however, looking at more established professions, I don't take this possibility too seriously. Certainly, licensing requirements will lead to acceptance of the computing professional by industry.

All in all, I believe that licensing will serve a useful purpose, although it would be unfortunate if licensing succeeded simply in transforming the computing practitioner into a professional only in the legal sense.

The Professional Society

Finally, the professional society has its role in shaping, nurturing, and protecting the professional. The most apparent example of a "professional" society that comes to mind in this country is the Association for Computing Machinery, yet actually, ACM is more an educational-scientific-technical society. Mainly, it was established by and for computing scientists to help disseminate knowledge among its members, primarily achieved through publications, but also by visiting lecturers, visiting scientists, conferences, and committees for curricula development (for example, the Curriculum Committee on Computer Science and the Curriculum Committee on Computer Education for Management).

More recently, ACM has been slowly moving in the direction of the professional society. Many newer members are not scientists; by inclination and by experience they are practitioners. This has created conflicts at times, and given ACM a mixed identity. Most conflicts have been handled reasonably well, and recent ACM activities have highlighted the mixed identity: the professional aspects to complement the scientific. Thus, ACM (principally together with the Data Processing Management Association) was a prime mover in the creation of and a charter member of the Institute for Certification of Computer Professionals (see [13]); certainly, certification is a necessary step in the direction of licensing. Further, ACM and the National Bureau of Standards have embarked on a joint action program (see [5]), including a project to define job classifications in computing, another forerunner of licensing. Recently ACM passed a Code of Professional Conduct, still another step in the licensing process.

Many of these activities have been undertaken with some misgivings, especially by members from the science community. With some justification, Miller (as quoted in [2]) has said of the ACM Code of Professional Conduct: "I must say that provisions of this type are platitudinous; they are somewhat like asking for a pledge of allegiance to apple pie, motherhood and the American flag." Moreover, some regard the professional society itself more as a device to protect the professional against the public than the public against the professional. Some of these and other similar attitudes make it uncertain that ACM will adopt a more complete professional society posture. For example, at present there are no rigorous entrance or graded membership requirements, nor are these likely to be adopted in the near future. Although there is a code of conduct, it is unlikely that formal enforcement procedures will be enacted soon because of legal, financial, and technical implications. ACM has proposed that the American Federation of Information Processing Societies establish a Washington office to furnish technical guidance for proposed legislation involving computing technology and to help make computing a more accepted part of the scientific and engineering "establishment"; however, the wheels of progress will move slowly in this area also.

Be that as it may, many of the recent activities of the ACM (and the DPMA and the British Computer Society) can't help but accelerate the movement toward professional status in the computing field. I believe that this is as inevitable as it is worthwhile, if we recognize and are prepared to accept the more formal ethical, legal, and technical implications. Paraphrasing Ralston [14]: Ultimately, professional status is not bestowed; it is earned —earned by technical performance, but also by attitude toward oneself and toward society at large; earned by education and experience, but also by a sense of mind. Working for the public good sounds somewhat square in these sophisticated times, but in the last analysis, it is the true hallmark of the true professional.

Acknowledgment. Let me express my appreciation to a number of people who helped shape my thoughts, and who critiqued this paper for content and for style: Joseph Cunningham, Carol Finerman, Bernard Galler, Calvin Gotlieb, Herbert Grosch, Richard Kieburtz, Glenn Lairmore, David Levine, Haskell O'Brien, Anthony Ralston, Lee Revens, Jean Sammet, and Robertson Stevens.

References

1. Stone, M. In search of an identity. *Datamation* (Mar. 1972), 52–59.
2. Gotlieb, C.C., and Borodin, A. *Social Implications in Computing*. Academic Press, New York, 1973.
3. Finerman, A., and Ralston, A. Undergraduate programs in computing science education in the tradition of liberal education. 1970 IFIP World Conf. on Computer Education, Sci. Assoc. Internat.
4. Gotlieb, C.C. *Informatiks as a Science*. Springer-Verlag, New York, 1971.
5. Carlson, W. Summary report to the ACM Council—ACM/NBS Action Programs. Memo., Apr. 23, 1974.
6. Hoffman, K., et al. Creative renewal in a time of crisis. Rep. of the Commission on MIT Education, 1970.
7. Glaser, G. Keynote Address. 1974 NCC, May 6, 1974.
8. Archibald, J.A., Jr., and Katzper, M. On the preparation of computer science professionals in academic institutions. Proc. 1974 NCC, AFIPS Press, Montvale, N.J.
9. Kapur, B.K. EDP education—An acute crisis. Proc. 1974 NCC, AFIPS Press, Montvale, N.J.
10. Seegmuller, G. Computer science and the quality of future computer specialists education. Organ. for Economic Coop. and Develop., 1972.
11. Gilchrist, B., and Weber, R. Sources of trained computer personnel, and Employment of trained computer personnel. Proc. AFIPS 1972 SJCC, Vol. 40, AFIPS Press, Montvale, N.J., pp. 733–748.
12. Balk, W. The human dilemmas of MIS. In R. Teague and C. Erikson, *Computers and Society*, West Publishing, 1974, pp. 3–8.
13. Harris, F.H. The role of computer science education in certification of computer professionals. ACM Special Interest Group on Computer Science Education, SIGCSE Bul., Vol. 6, 1, pp. 1–5.
14. Ralston, A. The roles of ACM and its members: Relevancy and responsibility. ACM Pres. Letter, *Comm. ACM 16*, 7 (July 1973), 397–398.

Positivity and Norms

F.L. Bauer
Mathematics Institute
Technical University of Munich

Following some lines of joint work with A.S. Householder, the character and use of algebraic methods in the theory of norms is demonstrated. New results concerning norms with values in an Archimedian vector lattice (not necessarily being totally ordered) are given, in particular for the generalization of order unit norms, L-norms and M-norms. An example of application to operator norms is given concerning contraction properties of positive operators.

Key Words and Phrases: matrices, norms, positivity, numerical range, positivity cones, vector lattice, absolute, monotonic

CR Categories: 5.11, 5.14

1. In the fifties, my interest in norms, coming from the usual motivation of the numerical analyst to "estimate errors" (V.N. Faddeeva, 1950 [1]), was strongly stimulated by Alston Householder, after first having met him at the Darmstadt meeting in 1955. In those days it was fascinating to see that norms in vector spaces could also be used to obtain, in some generality, results that had been known only for "special cases"; for example, the eigenvalue exclusion theorem for the mapping, A, of a vector space into itself
$|\lambda| \leq \text{lub}(A) \leq \|A\|$ was the generalization of the results of Hirsch, 1902 [2], and Frobenius, 1908 [3],
$|\lambda| \leq \max_i \sum_j |a_{ij}| \leq n \max_{ij} |a_{ij}|$ for the operator norm, lub, belonging to the maximum norm and for the submultiplicative norm
$\|A\| = n \max_{ij} |a_{ij}|$. The definition of the operator

Invited Householder lecture, given at the ACM Southeast Regional Meeting, Nashville, Tennessee, April 18, 1974. Author's address: Technical University of Munich, Mathematics Institute, Arcisstrasse 21, 8 Munich 2, Federal Republic of Germany.