Peter J. Denning

The IT Schools Movement

Building an education for an IT professional without the idiosyncratic constraints of any particular specialty … from the ground up.

An important movement is taking shape on college campuses. This is the movement to organize IT schools. It is a welcome development in the growth of an IT profession. The movement is gaining a momentum that overcomes the traditional territorialism of academic departments. We are at last beginning to address the challenging problem of designing an education for an IT professional that is not constrained by idiosyncrasies of any particular specialty, most notably computer science.

The movement is propelled by three new realities: (1) IT is a profession of many specialties. No one traditional degree program has the required breadth. (2) IT curricula must include a professional body of knowledge complementing the intellectual body of knowledge. This is within the scope of engineering accreditation guidelines (the Computer Science Accrediting Board is now part of ABET), and is recognized by some international professional groups such as the British Computing Society. (3) Many universities have declared they will be leaders in educating IT workers. A school gives them the means to bring their IT programs into prominence.

Until 2000, colleges of computing and IT were few in number. The pioneers include the School of Information Technology and Engineering at George Mason University (1986), the School of Computer Science at Carnegie Mellon University (1988), the College of Computing at Georgia Institute of Technology (1991), the College of Information Science and Technology at the University of Nebraska, Omaha (1996), and the College of IT at the United Arab Emirates University (2000).

In 2000, the Computing Research Association (CRA) and ACM sponsored the formation of a community of IT deans, which numbered about three dozen when the new programs were included. The group, currently hosted by the CRA, plans one or two meetings a year. Also in 2000, the first model curriculum for an IT college appeared [4].

Foundations of IT Education

Ours is a field of buzzwords where meanings have blurred under a barrage of flashy vendor advertisements. Nowhere is the blur more obvious than with four words at the very foundation of our profession. The distinction between "data" and "information," once carefully observed by computing professionals, has all but disappeared [5]. "Knowledge" has been trivialized to the content of databases. "Practices" are seldom seen as legitimate forms of knowledge. Our sloppiness with these

PETER HOEY

terms undermines our ability to design education programs for IT professionals. It also undermines our credibility with others, who wonder whether to believe our claims for a solid scientific basis for IT, for effective professional education, for productivity-enhancing business systems, and for safe and dependable software. I use these definitions of our four foundation terms:

- "Data" are symbols inscribed in formalized patterns by human hands or by instruments.
- "Information" is the judgment, by an individual or group, that given data resolve questions. In other words, information is the meaning someone assigns to data. Information thus exists in the eyes of the beholder; the same data can be nonsense to one person and gold to another.
- "Knowledge" is the capacity for effective action in a domain of human practice.
- "Practices" are recurrent patterns of action effectively accomplishing certain objectives with little or no thought. Practices are embodied knowledge.

Lewis Perelman [9] likens these distinctions to eating in a restaurant. The data are the symbols on the menu; information is the understanding of what the menu offers; knowledge is the dinner; practice is the digestion that turns the dinner into useful nutrients.

Although these distinctions are not practiced rigorously in colleges and universities, their widening adoption in the industry and profession will almost certainly engender significant shifts in education [2].

Because these distinctions call attention to knowledge as performance and to practices, they directly support the criteria for a profession:

- A durable domain of human concerns;
- A codified body of principles (conceptual knowledge);
- A codified body of practices (embodied knowledge);
- Standards for performance; and
- Standards for ethics and responsibility.

Ours is a field of buzzwords where meanings have blurred under a barrage of flashy vendor advertisements.

The curriculum of an IT school can be designed to support these defining aspects of the profession. We can draw directly on the well-developed bodies of principles already built into the curricula of our degree programs. We cannot, however, draw on any bodies of practices, as none have been defined for IT. Software engineering is ahead of the other specialties in defining its body of practice, but the guidelines have not yet found their way into curriculum specifications [8]. The ACM and IEEE Computer Society have adopted ethical standards for members and also for software engineers; neither enforces these standards, however, and most curricula only offer a single course in ethical responsibility. (Paradoxically, faculty everywhere complain that cheating is on the rise—even in ethics courses.)

What might a framework for a body of practices look like? I sketch one that I have found useful. Established professors recognize levels of professional competence. Most people in the profession can state the criteria for each level and can give examples of persons in each level. Seven levels are mentioned most often:

- Beginner;
- Advanced beginner;
- Entry-level professional;
- Proficient professional;
- Expert;
- Virtuoso; and
- Master.

Each level has its own standards of performance derived from the standards of the community. These are levels of embodied competence—performance in action. It takes time, practice, and experience for a person to attain each level. One's professional career can

IT Core Curriculum in the UAEU Model.

Area	Freshman	Sophomore			Junior and Senior
Math		Calculus concepts, (derivative, integral, simple diff eq, linear algebra)	Probability and statistics	Discrete math	
Systems		Digital hardware and communication	Computing systems (OS, networks, architecture)	Information systems (Web, database, security)	
Programming	Prog I	Prog II			Prog III
Business			Business basics	Enterprise basics	
Science			Biology concepts	Physics concepts	
Communication					Speaking and writing
Professional responsibility		Sophomore professional responsibility workshop			Junior and senior professional responsibility workshops
Exhibition				Core exhibition	

be interpreted as a journey on a path to mastery.

One way to map this into an IT curriculum is as follows: Set the overall objective of a curriculum to deliver entry-level professional competence. (That objective is met by few undergraduate programs.) Map the first three levels to an IT college curriculum as follows:

Beginner (sophomore year). This person is aware of the structure of the IT field and the nature of the work in the various specialties; able to develop algorithms, data structures, and simple circuits to solve well-defined problems; and able to program and test those algorithms and circuits. A beginner is not expected to see the connections and interrelations among all the components that make up typical computing systems, but is expected to understand the purpose of each component.

Advanced beginner (junior year). The advanced beginner knows the terminology and concepts of the specialty, sees many of the connections among components of computing systems; is able to design algorithms (and possibly circuits) of moderate complexity (several dozen modules), program them, and test them; is able to carry out tasks for a customer but needs supervision to avoid common pitfalls and breakdowns. This person is able to communicate effectively in speaking and in writing.

Entry-level professional (senior year). This person is thoroughly familiar with the specialty; can understand systems and diagnose system problems; can design systems of moderately large complexity (hundreds of modules), program them, test them, document them, and present them; and can carry out standard professional tasks for customers in application domains without supervision. This person understands professional ethics and acts ethically.

Models

I led a team that produced a model curriculum for a new IT College at the United Arab Emirates University (UAEU) [4]. The model incorporates these principles as follows:

- Specify a common program of general education for freshman and a new IT core curriculum for sophomores (see the table). The core program is consistent with the new ACM/IEEE-CS Curriculum 2001 guidelines, which are designed to enable IT specialties to draw on the basic computing concepts relevant to their areas [1].
- It specifies concentrations in seven IT specialties, leading to accreditable degrees by the end of senior year. The concentrations are computer science, computer engineering, software engineering, network engineering, information security, information systems, e-commerce, and educational technology.

These concentrations follow curriculum guidelines issued by the professional societies whenever possible.

- It defines competence levels for students to attain by the end of their sophomore, junior, and senior years; and correspond respectively to the definitions for beginner, advanced beginner, and entry-level professional. Each spring semester includes a team project shown publicly at the end of the year during Exhibition Week. Faculty assess student competence in professional practice during these exhibitions [3].
- It specifies that every major spend one semester with industry through an internship program.
- It specifies a professional responsibility workshop, which meets one hour a week during the fall semester of sophomore, junior, and senior years. The workshop is an opportunity for students and faculty to engage in ongoing conversations about professional responsibility and ethics at increasing levels of maturity.

This curriculum envisions that the student-teacher relation of "apprentice-master" will be the most appropriate in the practices segments of the curriculum. Teachers will need to be competent both as presenters and as coaches.

The UAEU model is best suited for a school that wants to offer degrees in a set of IT core specialties. But it is not the only model.

Another model is exemplified by the new School of Informatics at Indiana University, Bloomington. Its novel academic structure is likely to appeal to a great many universities. Rather than operate as a completely self-contained unit, it has a small core faculty and partners with the faculty of participating departments from other schools. Its common core program in IT includes segments on computer science, information science, public policy, business, and applications. Each participating department offers a specialization track for students who have completed the core. With this structure, students can achieve solid grounding in IT, which they can then combine with business, law, health, or humanities.

A smaller-scale example of the same model can be used in a single interdisciplinary degree program. About half the credits can be in a core program administered by the degree; the other half are in electives administered by participating departments. A capstone project featuring interdisciplinary teams can complete the curriculum.

Conclusion

In another decade, we can expect to see many universities offering degrees with models such as these. It is only a matter of time until the professional societies offer guidelines on curricula for IT schools. These schools will become the educational backbone for new entrants into the IT profession.

These schools may offer a way to address the perplexing problem of poor cooperation between university and industry education. There is a vast network of over 2,000 corporate universities whose annual spending equals that in the public universities—and yet there is almost no interaction between them and the public universities in regard to curriculum, professional degrees, and continuing education [7]. Universities and businesses have difficulties agreeing on joint research projects because of intellectual property issues. Many business leaders are loath to support public academic programs, believing the state is already paying for that from their taxes.

Ten years go, it would be anathema to consider such an educational program. Now it's about to become mainstream. **C**

References

1. CC2001. Computing Curriculum 2001 Web site; acm.org/sigcse/cc2001 or computer.org/education/cc2001.
2. Denning, P. How we will learn. In *Beyond Calculation: The Next 50 Years of Computing*, P. Denning and R. Metcalfe, Eds., Copernicus Books, NY, 1997.
3. Denning, P. Educating a new engineer. *Commun. ACM 35*, 12 (Dec. 1992), 83–97.
4. Denning, P., et al. A model curriculum for an IT college. (2000); cne.gmu.edu/pjd/UAE.
5. Holmes, N. The great term robbery. *IEEE Computer* (May 2001), 94–96.
6. Indiana University. School of Informatics, 2001; informatics.indiana.edu.
7. Meister, J. *Corporate Universities*. McGraw-Hill, NY, 1998.
8. Meyer, B. Software engineering in the academy. *IEEE Computer* (May 2001), 28–35.
9. Perelman, L. *School's Out*. Avon, 1992.

Peter Denning (pjd@cs.gmu.edu) is the past president of ACM and the director of the ACM IT Profession Initiative.