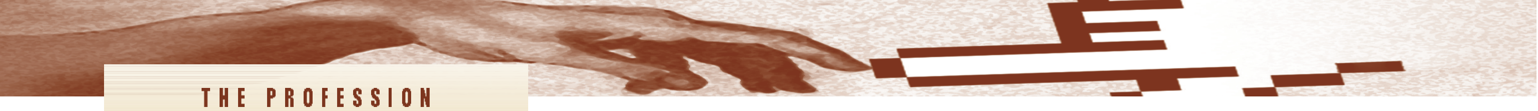


Computing as an Evolving Discipline: 10 Observations

Jiming Liu, University of Windsor

The profession must strive to make computing schools and departments the pillars of technological advancement.

Computing as a discipline can be traced to the 1960s, when universities such as Stanford and CMU established computer science as a stand-alone department. Since its inception, computing has been constantly evolving as a vibrant discipline.

Here, I make 10 observations on its recent development, which can serve as references for computing stakeholders—including academic leaders, faculty members, administrators, staff, students and their parents, alumni, and potential employers—to rationally embrace the changes and opportunities ahead.

A VIBRANT PROFESSION

With my first observation I stress the reality of a *strong job market*. Many people might have the misconception that a significant number of computing jobs have disappeared because of post-dot-com consolidation. Yet today's demand for computing professionals worldwide is higher than ever.

The high-tech hub of Silicon Valley clearly reflects this comparison between the current situation and the dot-com era. According to a recent US Department of Labor projection, computing graduates will remain highly sought after in the years to come. At the same time, the industry struggles to fill the available positions because we lack enough qualified graduates. This stems in part from declining enrollments in computing and in part from the mismatch between industry requirements and recent graduates' skill sets.

Acquiring new skills

With my second observation I acknowledge the need for acquiring *new skills*. Educators based many early computing curricula on the assumption that understanding general concepts, techniques, and programming languages would be sufficient for building a computing career. Today, industry increasingly adopts sophisticated tools to enhance productivity. Developers thus require specialized yet versatile skills in working with off-the-shelf software or hardware solutions. They also need expertise in using development environments for real-life on-demand tasks.

The job market is undergoing a shift in requirements from general computing knowledge and programming skills to those of interdisciplinary domain knowledge and integrated application development and problem-solving skills. Examples of core computing skills include but are not limited to Web application development and support, data warehouse development, QA and testing, Linux and embedded systems programming, and working with existing legacy systems. These skills also include Java, .NET, C++, and SQL programming and Oracle database management.

Emerging roles

My third observation recognizes the importance of promoting the *emerging roles* of computing professionals. Many people tend to see computing career prospects as relating to previously popular jobs, such as software engineers, systems analysts, networks administrators, and programmers. This perception will soon change, as new roles are currently being defined and redefined.

Computing graduates now fill positions outside software houses and computer services departments. For example, they work in the financial industry as data modelers or online business specialists and in the bioinformatics industry as application engineers or data administrators and analysts.

Today, the world operates on computing technologies that go much beyond daily gadgets, desktop applications, or online services. We must effectively communicate to parents and future students the emerging roles and impacts of computing professionals in society.

Students majoring in computing should have lasting enthusiasm and high aspirations for building the next generation of computing technologies—for example, setting up new industry standards and infrastructures for intelligent personalized media technologies or developing Web x.0 applications. As the agents of innovation, computing students must achieve the

Continued on page 110

Continued from page 116

core competence necessary for undertaking future sociotechnological development. Universities must continue to instill intellectual excitement, offer innovation opportunities, and reward risk-taking experience in the discipline.

AN ADAPTIVE DISCIPLINE

Neither mass computing education nor specialized training will adequately address the issues examined thus far. With my fourth observation I recommend *dual-emphasis, flexible* computing programs. The DEF programs' key lies in their comprehensive yet flexible computing curricula, which demonstrate some distinct features.

Dual emphasis

DEF programs should aim at achieving a delicate balance between fundamental knowledge preparation for students pursuing advanced studies or development and essential hands-on experience for those who will enter the job market immediately. Different regional socioeconomic environments might dominate how computing opportunities arise, thus determining the scope and scale of the actual program implementations. In dual-emphasis computing programs and curricula, certain foundations—such as math, logic, discrete structures, algorithms, and programming—will remain core for all students.

With respect to both emphases, we must integrate and reinforce the learning of key computing concepts in application-driven, problem-solving experience. While injecting new courses and training opportunities to meet future requirements, we should avoid simply turning them into vocational training on an array of specific tools or gadgets.

Flexibility

The DEF programs should accommodate students from varying backgrounds—whether high school graduates, returning students, or students with a first academic degree in other disciplines—to reach the level of computing foundation and competence needed to meet industry requirements and the demands of advanced studies. Bridging or introductory courses will help, but the ultimate solution would be to provide better training in the high school curriculum for math, logic, data management, and even programming.

The evolution of computing offers an expanding array of opportunities

The DEF programs should also offer multiple exit options for

- degrees in general computing, applied IT or computing applications, information systems, comprehensive computer science honors, software engineering, or interdisciplinary computing, with some practicum, internship, co-op placement opportunities;
- second or joint computing majors or degrees in addition to business, communication studies, education, engineering, management, medical studies, physical sciences, social sciences, and visual arts among others; and
- top-up or integrated postgraduate diplomas and degrees in computing.

In providing these options, less early-year specialization labeling, fewer course-sequencing constraints, and more elective courses or modules would be most desirable.

LEARNING INDICATORS

My fifth observation addresses misperceptions involving *learning*. Many might perceive computing as difficult for beginners because technologies in daily life seem increasingly sophisticated. The way we introduce new technologies should eliminate this misperception. Computing is essentially a discipline much richer than its attendant jargons. This discipline is by no means designed exclusively for a small group of math- or engineering-oriented learners.

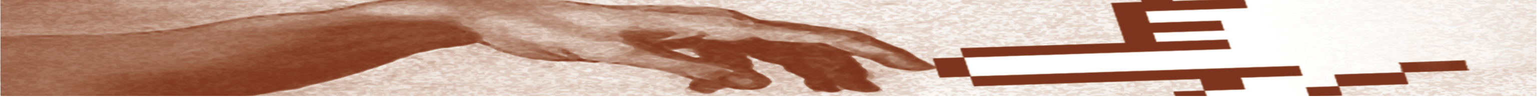
The drill-and-practice of computing skills in today's programs has, to a certain extent, succeeded in imparting computing training's fundamentals. We must go further and educate our students to become more creative and adaptive in systematically dealing with real-world problems. The students must understand and practice software development principles and integrated systems thinking to become application engineers and system specialists. As an analogy, an artist who knows only the different types of paint and brushes could never create a great work of art.

With my sixth observation I recommend applying *success indicators*. Because computing schools and departments traditionally have been managed like their sibling departments in engineering or science, they might sometimes lose their unique identity. To establish the DEF programs or curricula, we need faculty with varying expertise in theory and practice: Some members might focus on advanced fundamental research, while others might play an active role in applied research and technology transfer. Each should have a corresponding reward system.

Given the discipline's distinct nature, we face a general need to envision a new set of success indicators or performance benchmarks specifically for computing schools and departments. These benchmarks can concern both faculty—how to define and measure the impact of their contributions in teaching, research, and services, for example—and students—such as how to define and evaluate their knowledge and core competence in technical, communication, management, design, and problem-solving skills while working in a collaborative environment.

COMPUTING'S FUTURE

The evolution of computing offers an expanding array of opportunities. My seventh observation is that computing will continue to *branch out*. As a result, new areas will become increasingly specialized, diversified, and transdisciplinary, with the seam-



less connection between university education and forefront innovative applications. In this respect, some schools and departments have already successfully developed courses or programs in some of the rapidly growing areas, such as bioinformatics, digital arts, and health informatics.

A trend has been developed for reorganizing and establishing new streams, divisions, departments, or schools that can better reflect the areas of present computing education and future needs.

Merging disciplines

My eighth observation acknowledges computing's *interdisciplinary* nature. Clearly a science discipline, computing subscribes to many fundamental theories and principles. Computing is an engineering discipline because it involves putting theories and principles into practical use. It is also a psychological and social discipline because it directly involves man-machine interaction and human knowledge processing, while enriching human experience. Computing also has a professional dimension because it endorses and follows certain professional practices and social norms.

From the perspective of the specialized and diversified nature of real-world computing jobs, students should be exposed to the most relevant problems and challenges that address scientific, engineering, social, and business issues.

Advancing research

My ninth observation affirms that *advanced research* will remain the driving force for breakthroughs in computing technologies and the emergence of new computing paradigms. Equipping students with strong computing, math, and engineering foundations—and with the analytical skills for pursuing further studies—will be an integral part of the DEF programs and curricula. Similar requirements will also be observed as cutting-edge computing companies continue relying on graduates capable of venturing into uncharted technological territories, trying out new ideas, and developing new gadgets, tools, infrastructures, and services based on their knowledge of the applicable foundational sciences.

In this respect, we must update our programs and curricula to reflect the depth in basic training as well as to capture the breadth for promising interdisciplinary studies. Recent examples of such studies are the ones on understanding the structure and behavior of real-world networks including biological regulatory networks, WWW networks, and social networks and communities.

Shifting paradigms

With my tenth observation I call for astute management of continual *paradigm shifts*. As computing evolves, we will experience waves of boom and bust. Whenever a paradigm shift in computing technologies occurs, especially those that fundamentally change the nature of software, hardware, infrastructure, and applications, it will most likely create new development and application opportunities. Previously, shifts such as the move from analog to digital, from mainframe to personal computing, from stand-alone to networked computers, and from desktop to ubiquitous computing have all redrawn the computing landscape.

In this regard, one promising development lies in the emergence of new computing paradigms, such as nature-inspired or autonomy-oriented computing for large-scale, distributed applications in Web-based services, mobile sensor networks, and scientific modeling and computing. As an emerging paradigm, autonomy-oriented computing draws on the metaphors and models of autonomy that nature offers, as well as their roles in addressing practical computing needs, such as for distributed problem solving and complex systems modeling. Besides new computing paradigms, other opportunities could derive from the promises and impacts of some fast-developing interdisciplinary computing fields, such as bioinformatics and computational biology.

As we assess the recent changes in computing and identify evolving directions, investing in our computing discipline might seem an obvious step toward making our programs more attractive and our graduates more marketable. However, our ultimate vision remains: to make computing schools and departments the pillars of future technological advancement and to nurture new generations of computing professionals and leaders who will pursue cutting-edge innovation, tackle socioeconomic development and moral issues, and strive to make the future computing-centric world a better place to live, work, and play. ■

Jiming Liu *is a professor and the director of the School of Computer Science, University of Windsor. Contact him at jiming@uwindsor.ca.*

Editor: Neville Holmes, School of Computing, University of Tasmania; neville.holmes@utas.edu.au. Links to further material are at www.comp.utas.edu.au/users/nholmes/prfsn.

IEEE Computer Society members

save 25%

on all conferences sponsored by the IEEE Computer Society

www.computer.org/join