Expanding Computing as an Academic Service Discipline

Atanas Radenski, *Chapman University*

In addition to its traditional role as a professionally oriented major, computing must expand as a mainstream service discipline, with significance for all schools in academia.

On a pleasant June evening while at the Los Angeles Opera, I found myself thinking of computing while watching the world premiere of Elliot Goldenthal and Julie Taymor's production of *Grendel*. This opera presents a musical version of the old Beowulf legend from the monster's viewpoint, in a way that makes the audience feel sympathy for the monster.

The media reported that the opera's budget had become monstrous as well, at least partially because of the sophisticated high-tech equipment employed, which cost nearly $2.8 million. A special computer program controlled the production's 20-ton set, which represented an icy winter scene on one side and a warm spring vista on the other. The computer program controlled the set's rotation and motion for smooth transitions from one season to another, while singers performed confidently atop the precisely programmed moving set.

COMPUTING FOR EVERYONE

In times such as these, when computing belongs to opera, it belongs to every discipline. Academic computer science programs must expand their reach and promote computing as an academic service discipline to ensure that all students graduate prepared for a competitive and dynamic job market.

Thus, computing faculty must help other departments recognize and accept the value of computing to their curricula. Computing faculty should energetically popularize successful computing techniques in traditionally noncomputing majors. Telling compelling stories offers one way to achieve this.

Twining disciplines

In filmmaking, for example, the performance-capture technique digitally records movements and digital character animation. This permitted Tom Hanks to perform six different parts in *The Polar Express*, the first feature film completely implemented with the performance-capture technique. The film's director, Robert Zemeckis, has also completed *Beowulf*, another feature movie fully based on motion capture.

I met with Robert in his *Beowulf* studio, together with several film and computer science students as well as faculty from Chapman University. During our conversation, Zemeckis expressed his concern that motion-capture personnel are either proficient in film or in computing, while the real need is for professionals competent in both areas.

I could easily agree with the renowned film director, at least because modern motion-capture software, such as AutoDesk MotionBuilder, is open to beneficial customization and enhancements by end users who write small programs in programming languages such as Python. The *Grendel* and *Beowulf* stories in general, and Zemeckis' opinion in particular, have helped me convince my liberal arts colleagues of computing education's significance to noncomputing majors.

A discipline for all fields

While computing is still news in the opera world, it is well-recognized in science and engineering. The broad field of scientific computing combines techniques to analyze

Continued on page 101

Published by the IEEE Computer Society

Continued from page 104

and solve scientific and engineering problems. In biology, for example, high-performance computers analyze the genome sequences of humans and other organisms and assist in a variety of other tasks. In social sciences, humanities, and linguistics, computers mine large textual datasets, conduct morphological analysis, execute computationally demanding visualizations, and perform pattern recognition and analysis.

Outside the sciences, researchers apply computational power to music for composition, law for legal information storage and retrieval, business and economics for advanced computer modeling, and dance to generate images and music to be mixed with live performances. Computing faculty must publicize such stories to help nonscience departments accept the value of computing education to their disciplines.

GOVERNMENT SUPPORT

Government funding agencies have increasingly recognized and promoted computing's importance as a service discipline. The NSF's Cyber-Enabled Discovery and Innovation program, for example, promotes computational concepts, methods, models, algorithms, and tools in a wide variety of science and engineering research and education contexts. Further, the National Endowment for the Humanities has established a special Office of Digital Humanities and is even sponsoring a novel Humanities High-Performance Computing program in cooperation with the Department of Energy. Computing departments must promote such programs in other departments and also advocate computing as an academic service discipline.

In the age of omnipresent computing, qualified computing professionals must support noncomputing professionals, such as set designers, filmmakers, scientists, engineers, lawyers, accountants, and doctors. In this context, Neville Holmes calls the computing profession secondary, thus underlying its important service role to the serviced—or primary—professions (http://doi.ieeecomputersociety.org/10.1109/MC.2007.35). Computing departments can benefit from promoting the idea that the computing profession is secondary to other professions—specifically that it fulfills other professions' data processing needs. Indeed, all students who major in primary professions should obtain general education in the computing field so that they can be well-prepared to work effectively with computing professionals.

Computing is steadily gaining recognition as an indispensable component of virtually any profession and of everyone's personal life.

Further, this view can convince students from noncomputing departments to pursue computer science as a beneficial second major; pursuing double degrees is supported in the US, Australia, Canada, and elsewhere.

THE GRAND CHALLENGE

Too often, potential students perceive traditional computer science as a discipline that prepares professionals to deal with commercial software development and little else. According to Jill Ross, improving this image is "perhaps the greatest Grand Challenge" in computing today (www.cra.org/CRN/articles/nov07/ggc.html), while Neville Holmes proposes reimaging computing as a profession that facilitates other professions' use of data; program coding as a craft performed by technicians, not by computing professionals; and constructing software as a branch of digital engineering rather than computer science (http://doi.ieeecomputersociety.org/10.1109/MC.2007.35).

Overcoming stagnation

The inadequate program-coding image provides one reason for the prolonged enrollment stagnation found in traditional computer science programs, despite the remarkable rebound of the job market since the 2000 dot-com bubble burst.

On the positive side, computing is steadily gaining recognition as an indispensable component of virtually any profession and of everyone's personal life. Under these circumstances—decline of enrollment but growth in recognition across the disciplines—computer science can and should be promoted and expanded as an academic service discipline with growing significance for all other disciplines, in addition to its traditional role as a professionally oriented major. Mark Guzdial and Elliot Soloway assert that academic "computer science is more important than calculus" (http://portal.acm.org/citation.cfm?id=782943) and believe it can be stimulating and beneficial for any student to learn how to communicate with a complex machine, and do so elegantly.

Computing education: A two-tier view

Figure 1 shows a two-tier view of computer science educational activities, including core- and discipline-oriented computing. Accordingly, computer science departments already tend to hire two types of faculty: core computer science specialists and computing experts in various disciplines, such as computational finance, computational linguistics, computing in the humanities, and so on.

Such two-tier departments can maintain and enhance traditional core computing programs and, in addition, offer custom computing courses across the disciplines, such

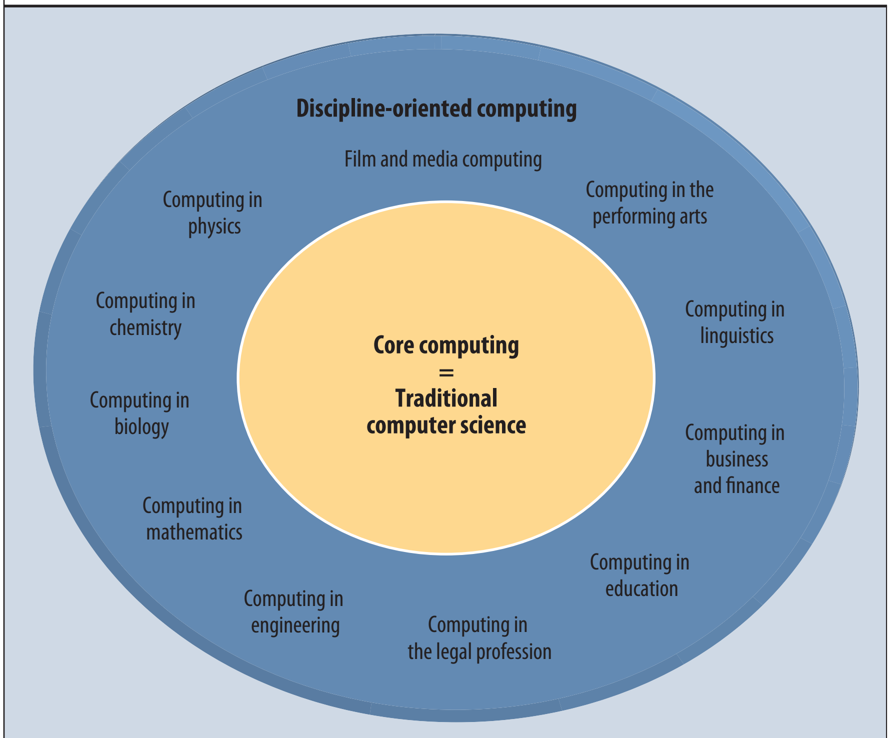


Figure 1. Two-tier view of computer science educational activities, including core- and discipline-oriented computing.

as computing for performing arts, scientific computing, computing for the legal profession, and so on. Such courses can prepare computing and noncomputing majors for cross-discipline professional cooperation.

There is evidence that academia might be moving toward two-tier departments. A recent CRA Best Practices Memo, "Promotion and Tenure of Interdisciplinary Faculty," states that "academic departments of computer and information science are increasingly recruiting and hiring people with interdisciplinary skills." The same report clarifies that " a number of universities avoid giving appointments in two or more departments to junior faculty, but instead hire them into one department" to avoid challenges in tenure and promotion with joint appointments. A single two-tier computing department can straightforwardly manage computing-related interdisciplinary programs, instead of being jointly hosted by a heterogeneous group of academic units.

CURRICULUM EXPANSION

The need to stimulate and increase interest in computing after the dot-com bubble burst spawned innovative undergraduate course development. Examples include gentle programming introductory courses; theoretical introductory courses; contextualized introductory courses; and courses centered on game development, virtual reality, multimedia, and robotics. New courses involve innovative pedagogy such as active, team-based, and collaborative learning, and utilize emerging information technologies, such as learning management systems, mobile devices, tablets, and social networking. Technology-oriented schools, such as Georgia Tech and Harvey Mudd, developed new courses to promote CS as a core subject taken by all students.

CS1: A SUCCESS STORY

To fuel student interest and satisfaction, we have employed active learning techniques in a newly designed CS1 course at California's Chapman University, a private liberal arts school that hosts computer science courses alongside mathematics in a single mixed department. Our CS1 course was so welcome during its first offering in the fall of 2004 that the university promoted the course to a science elective in 2005.

CS1 grew to be a true service course in 2007 when it became a general education quantitative reasoning option, together with calculus and statistics. The expansion of our CS1 course continued in 2008 when the Biology Department decided to require it for all biology majors.

CS1's successful expansion can be attributed to several factors. First, we made the course more accessible to students by using a Python subset, thus freeing freshmen from the complexity of our previous CS1 language, Java. Python is an interactive language that permits students to gain direct control of what they do on the computer. It provides an excellent vehicle for computing beginners to learn and enjoy programming.

Second, we based our CS1 course on active learning methodology, focused on drill and practice through self-guided interactive and programming labs.

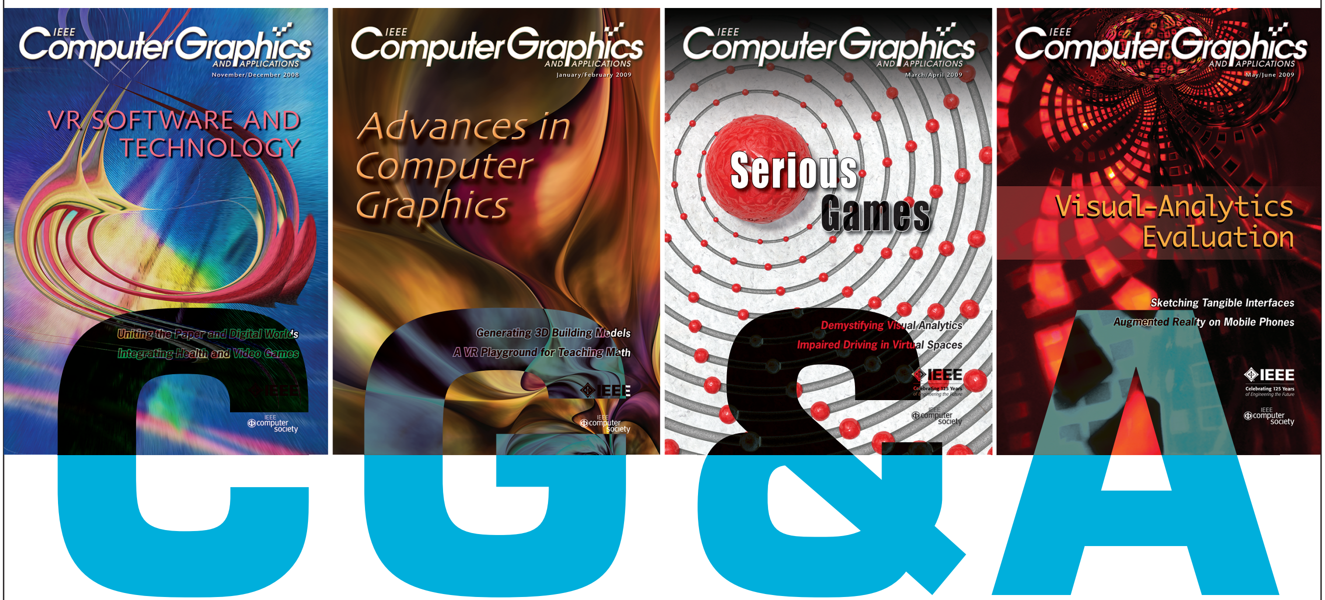
Third, we employed the Moodle learning management system (http://moodle.org) to make the entire course contents available online 24/7, in the form of a digital study pack (http://studypack.com) that supplements our regular lectures and lab sessions. The study pack integrates resources (complete textbook and lab manual) and activities (assignments, quizzes, forums) that free students from location and time constraints.

Fourth, at the university level we have systematically promoted computer science in general, and our CS1 course in particular, through faculty forums, student newspaper articles, and faculty-focused Python workshops.

Liberal arts institutions offer particularly good potential to promote computer science as an academic service discipline, not only because computing research and practice span nearly all disciplines taught in a liberal arts school, but also because adhering to a multidisciplinary spirit can be easier than in traditional engineering-oriented environments with their specialized in-depth technical programs. Partnerships between technical and liberal arts schools can ensure that liberal arts students learn the needed basic skills, and that technical students are educated in the social and humanitarian aspects of computing. In mixed mathematics and computer science departments, computer science can be established as a quantitative reasoning service discipline by adopting curriculum practices already in place for mathematics. C

Atanas Radenski is a professor of computer science at Chapman University. Contact him at radenski@chapman.edu.

Editor: Neville Holmes, School of Computing and Information Systems, University of Tasmania; neville.holmes@utas.edu.au



IEEE Computer Graphics and Applications bridges the theory and practice of computer graphics. From specific algorithms to full system implementations, *CG&A* offers a unique combination of peer-reviewed feature articles and informal departments. *CG&A* is indispensable reading for people working at the leading edge of computer graphics technology and its applications in everything from business to the arts.

Visit us at www.computer.org/cga