CRA Outline of CS Overview

Prepared by Dave Penniman, Dean
School of Informatics
University at Buffalo

July 13, 2003

Introduction

Unlike LS where the problem has generally been size (too small to be a free-standing unit), CS has, until recently, been dealing with the problem of growth. As such, these academic units, often departments, are the size of many LS schools. Recently, however, the growth trend has slowed. The attrition rate of as much as 60% in some units is now attracting attention. And the lack or decline of women and minorities in CS has caused concern. Finally, the rapidly changing technology, at the core of CS (and CE) has challenged these academic units to update an already crammed curriculum. How, then, can we liken the world of CS to the world of LS – or even IS where the challenges are also significant? I suggest the common factors are many, and revolve around a user-centered orientation – for design of systems, design of services, and design of educational programs to meet the needs of converging professions.

Growth and Attrition

While the early 1990's provided challenges of growth in CS, the current environment offers new challenges including halting growth and increasing attrition – a double whammy. "There is now evidence that even before the number of students choosing a computing major started to fall, undergraduates who were already in computing majors began migrating out at higher rates than in the past."[1]

While CS departments varied considerably in attrition rates (ranging from 1 to 66 percent) the factors leading to this attrition and its variability are not clear cut. Public versus private institutional setting or student/faculty ratios were not significant predictors of attrition.

Factors identified (though analysis is still underway) include poor teaching and advising, harsh grading, heavy demands and little reward for great effort.

Where faculty were rated high in scholarly quality (highly correlated with retention), the lecture was not seen as the most important element of instruction and presenting information to the students was not seen as sufficient to satisfying their role as instructors.

CS units must now focus on attracting and retaining students – a new game for those used to dealing with unfettered growth.

[1] Computing Research News, March 2003, pp. 2-3.

Technology

Technology trends identified in the ACM study of CS curriculum presented in 2001[2] included the following list:

- WWW and its applications
- NW technologies – especially those based on TCP/IP
- Graphics and multimedia
- Embedded systems
- Relational databases
- Interoperability
- Object-oriented programming
- Sophisticated application programming interfaces (API's)
- HCI
- Software safety
- Security and cryptology
- Application domains

The report points out that unfortunately curriculum design is a zero-sum game and to add new courses in any of these areas means that something else must be taken away. The solution seems to be to "thin down" what was already there to make room for the new.

And what about all the issues arising after 9-11 focusing on security as well as new demand for more intelligent data analysis and mining? And what about new technology on the horizon such as quantum computing?

Culture

In addition to technology challenges, CS faces cultural challenges as well. Again turning to the ACM study of CS curriculum[3] the following cultural forces were identified:

- Changes in pedagogy enabled by new technologies – DE via networks for example, as well as sharing of curriculum resources across institutions (does reward system support this, however?) Also, demo sw, computer projection, and individual lab workstations
- Growth of computing throughout world – increasing access to internet from the home, increase in familiarity of students with computing and applications, widening skill gap between those with access and those without
- Growing economic influence of computing technology – vision of "easy wealth" draws more students to the field, including those who lack the deep interest. Also industry draws away faculty with needed skills. Together these make it difficult to meet growth demand (see later comments on slowing growth)

[2] See http://www.acm.org/sigcse/cc2001/cs-changes.html Chapter 3, Changes in the Computer Science Curriculum
[3] ibid

- Greater acceptance of CS as an academic discipline – the battle of legitimacy has "largely been won". Problem now is how to meet demand
- Broadening of the discipline – focus on CS has now expanded to more encompassing discipline of "computing" (see for example the distinction now being drawn between computer science and "computational science")[4]

<u>Converging Professions</u>

There are areas of overlap between LS, CS and IS and these should encourage us to discuss in realistic terms the turf we have previously so carefully guarded. LS and IS have confronted this via mergers and acquisitions as well as new program development, while CS has stayed on the sidelines for the most part. Confronted until recently with significant growth and a "hard science" positioning, CS has not been interested in the challenges faced by the other two related disciplines. The times, however, are changing. One has only to look at the technological and cultural forces identified earlier to see that the future rests with "re-visioning" the domains and approaches within CS. Jim Foley gave us the "rabbit-eared" VENN diagram[5] (which we saw at our last meeting). This diagram helped to position the field of CS within a larger domain as well as showing the relationship to IT. We need a similar conceptualization regarding its positioning with IS and LS.

Oxford University Computing Laboratory has on its website[6] the following: information about computer scientists (paraphrased);

Computer Scientists study mechanized problem solving, develop the principles that guide future technological advances, provide tools for mastering the complexity of information systems (ranging from word processors to washing machines) and define which problems can and cannot be solved by mechanized means.

Computer Scientists have passion (for solving problems), ability (for using mathematics to find solutions), imagination (for modeling and solving problems not seen before), commonsense (to know if you are solving the right problem) and teamwork (when working on problems with others)

I'd like to work with these types of people and so would my colleagues in Informatics[7] in order to design better systems, better services, and better educational programs. And I'd bet we would all come out better off.

---

[4] Computational Science Education Project http://csep1.phy.ornl.gov/CSEP/PREFACE/PREFACE.html which uses a VENN diagram to depict the intersection of computer science, mathematics and applied disciplines to define computational science

[5] Computing >Computer Science http://www.cra.org/reports/computing/index.html

[6] http://web.comlab.ox.ac.uk/oucl

[7] As you already know from my previous presentations, "Informatics" is concerned with the intersection of PEOPLE, INFORMATION, and TECHNOLOGY – another VENN diagram is used to show this.