Humanization of the Computer Science Curriculum

J.A.N. Lee
Virginia Tech, Blacksburg VA, USA

janlee@cs.vt.edu

Abstract

Computer science programs in many universities are reacting to pedagogical initiatives to improve and broaden the education, particularly in the domain of adding less quantitative and more humanistic studies to a curriculum that is predominately technical. Integrating these concepts into the syllabus of computer science courses can be extremely difficult especially in the light of the lack of preparedness of many faculty. This paper examines some of these initiatives and suggests solutions to take the greatest advantage of these opportunities while at the same time improving the caliber of computer science graduates.

Keywords: Curriculum, Education, Ethics, Social Impact, Internationalism, Diversity.

Introduction

The curriculum of Computer Science programs are strongly influenced by the recommendations that have been published over the years by the Association of Computing Machinery (ACM) with later assistance from the IEEE Computer Society, and by the guidelines of the Computer Science Accreditation Board. (CSAB 1986) The most recent curriculum proposal (Curriculum 2001 [ACM/IEEE-CS 2001]) not only updates the curriculum to include more modern subjects, but also includes for the first time concerns for less technical topics in the form of computer history and professional concerns. Even so the coverage of non-technical topics lags behind the expectations of many academic institutions, particularly in those cases where the computer science resides in a college of Arts and Sciences or similar academic unit.

Typical of many other institutions since the mid-1980s, Virginia Tech has gradually imposed a series of university-wide initiatives designed to provide the graduating student with a broader view of the world than they were likely to achieve in a purely technical program. Each of these may be classified under the general heading of “… Across the Curriculum” initiatives, that were to be incorporated into each program of study in a way that was best suited to that field.

… Across the Curriculum

In some cases the stipulations for modifying the curriculum to meet some minimum pedagogical requirements were already satisfied in the existing curriculum. For example an initiative to have “Computing Across the Curriculum” had no impact on most of the engineering disciplines, while “Writing Across the Curriculum” was already fully fulfilled in the English major’s curriculum. In the majority of cases the first stage of the curriculum modification simply required the introduction of a single mandatory course in the program of study, while in other cases it was as simple as an “exposure”, perhaps one or two lectures or activities. Attempts to incorporate “across the curriculum” concepts into recommended programs of study by the professional societies were not as successful until the recent Curriculum 2001. However this new proposal is incomplete in distributing either history or ethics across the topics listed.

This paper concentrates on the inclusion of the more humanistic elements into the computer science curriculum, and the experiences at Virginia Tech of undertaking these modifications over the past decade. From our experiences others may not be led into the errors that we encountered or through the machinations of introduction that were later reversed.

A major shortcoming of the programs that we encountered was the false assumption that the introduction of these special topics in the programs of study of computer science students would be achieved by computer science faculty. Two competing points of view immediately emerged. On the one hand, perhaps the best prepared faculty to teach specialized topics were those whose original training was in those areas. Historians to teach computer history; philosophers and ethicists to teach computer ethics; English instructors to teach writing; and so on. On the other hand, who best to know how these topics would be affixed to the computer science curriculum than those who were experts in the field? After all, computer science faculty are, in the main, trained in computer science research not in educational methods, and yet they seem to do a reasonable job of educating computer science undergraduate students. Of course, they have had many years of observing their own teachers and they can mimic what they saw. This does not suggest then that they are well prepared in the development of teaching and lesson plans, or understand the "behind the scenes" activities that supported a teaching/learning experience. Moreover, in the past two decades, the manner of managing a teaching/learning experience has changed radically, some of which are the result of the introduction of computer technology that a computer scientist ought to understand, and some to better accommodate a changing student psyche that is more used to learning from television than from a preacher.

Not until the current generation of students have completed their own preparation will they join the ranks of faculty with some exposure to the topics in the "Across the Curriculum" collection. Like computer science in the early 1960s, the teaching of these specialized topics suffers from the lack of easily accessible resources that a teacher can use to prepare their classes. Bearing in mind the Richard Hamming observation that "anything a faculty member can learn a student can easily", and realizing that we survived the 1960s to create a viable academic entity, we perhaps should rely on time to solve the resource problems.

Two other unresolved questions were (and are) the manner of introduction of the topics into the curriculum. The simple solution is the insertion of an additional computer science course, or the requirement to take an already existing course as part of the electives. An alternative is the inclusion of the "Across the Curriculum" topics within existing courses where they will find immediate application. The former methodology requires the training (disruption) of only one faculty member but then seems to set the subject apart from the rest of the field and to have no application outside the course. Depending on a distributed approach requires the training of a large cadre of faculty and takes away from the available time to teach more technical and more satisfying element s to the embedding course. Further in a field that is still expanding requires that faculty pack more and more into existing courses and thus there is no room to add non-technical topics, or where "Across the Curriculum" topics have been added already, they are discarded in favor of new technology considerations.

Ethics

"Computer Ethics" became one of the first specialized topics courses to be added to the computer science curriculum when the guidelines for the computer science programs were originally published in 1986 (see CSAB, 1986). This was the time that the early computer hacking episodes were being reported by the press and there was a general feeling that computer science students were lacking in ethics. While there had been some attention paid to "computers and society" earlier, the integration of these topics into the curriculum was slow until accreditation was important. More recently the ABET (Accreditation Board for

Engineering and Technology) requirements for engineering programs has required similar incorporation of engineering ethics into their programs.

Ethics courses are generally found in most computer science programs today, though their domain has expanded to encompass the social impact of the computer and other career-related topics. At Virginia Tech the original proposal for such a course was entitled, and restricted to, “Computer Ethics”, but this was rejected by the college curriculum committee as not being in the province of a computer science department. A second submission entitled “The Social Impact of the Computer” suffered a similar fate, but the third attempt entitled “Professionalism in Computing” succeeded. This was a fate that in the end was for the best, since it has allowed this course to extend to topics beyond just ethics and social impact. This course has remained a separate course offered to junior students. A major element is now concerns for careers in computing, and the preparation for employment. Strongly supported by an industrial advisory consortium, this course deals with ethical and social concerns in both school and industry.

As the only non-technical course in the computer science repertoire, and bearing the title forced upon us, this course has become the host to the majority of other “Across the Curriculum” initiatives. These decisions have always been based on the impossibility of including the initiatives in other courses required in the accredited program of study.

Starting with a National Science Foundation Educational Infrastructure grant in 1993, this course has become a web based course that has gradually been converted to a digital library available to other institutions and now known to serve fourteen other programs throughout the US.

Writing

“Writing Across the Curriculum” was a 1990 initiative designed to improve the writing capabilities of students, but with the emphasis of giving experience in writing within their major. Following up on freshman English courses, but separate from a course in Technical Writing, the Professionalism course at Virginia Tech is also the writing intensive course. Requirements include having students prepare drafts of papers that can then be reviewed, edited, and revised before final submission, and to satisfy administrative accountability provisions, must be compared to previous submissions to demonstrate a favorable progression of writing development. With several hundred undergraduate students in the course per year these requirements are difficult to fulfill if the instructor is the only evaluator. With student cooperation while attempting to preserve privacy and anonymity, it is possible to partially complete this mission. Using the same groups of students as used within the course for other projects, a round-robin exchange of draft reports allows them to review other student’s compositions while learning to review. This has the advantage of allowing students to compare themselves with others both in their writing and their coverage of the subject. Since they will have the opportunity to revise their essays, it is possible that they will receive assistance from the person for whom they completed a review to enhance their own work. This is an excellent opportunity to discuss plagiarism in the course and the need for proper attribution. Students are required to do a self-evaluation of their work, and to compare their capabilities with their own prior submissions. Spot checks of reviews and editorial mark-ups helps to maintain standards of both editing and writing.

To encourage students to write while at the same time solving the problems of attempting to have meaningful discussions or case study analyses in a class of sixty plus students, an on-line asynchronous debate system was developed that has similarities to chat rooms and MOOs. This system is highly structured so to enforce an “Oxford-Union” style debate starting with pro and con positions and followed by arguments and contributions from all class participants. By requiring each contribution to either agree or disagree with the prior contribution together with a justification of the new position, students unashamedly write without realizing that they are writing!

Internationalism

The concern for students to realize that they live in a world community and not just a parochial environment, and that there are attitudes and beliefs that deserve as much respect as those of their home country, is an important aspect of any undergraduate program of study. Computer science graduates can expect to travel to other countries during their years of employment to visit their customers, or to attend technical conferences; they can expect to work with professionals from other countries in their development teams; they can expect to be required to conform to the laws and customs of other countries as part of their development activities; they can expect to need to be tolerant of other points of view in their daily lives.

This is a topic which is difficult to build into a technical course such as "operating systems" or "computer architecture", and thus it is appropriate that once again this is an element of the professionalism course.

Through cooperative arrangements with national representatives to IFIP (International Federation for Information Processing) Technical Committee 9 (Social Impact of the Computer) and the use of the Internet, US students have been able "compare notes" with students in other countries in responding to case studies and scenarios. Studying the constitution, laws and customs of another country in cooperation with a student from that country, they have been able to realize that their US centered response is not the one and only response. While they may admit that they still favor their US-centric point of view they learn that to respect the views of others. The results of this interaction compose new contributions to the resources of the course resources and can readily be used in an isolated environment since the collected assets are preserved on-line (Lee, 2001) To this time this comparison has focussed on three major concerns – privacy, censorship, and freedom of speech. In the development of resources in support of these aspects of teaching computer science is the need to provide differing views on these and further concerns.

Diversity

Most of our educational institutions are 'struggling" to provide opportunities to recruit a more ethically representative student body, and to develop campus environments that are nurturing to all communities. Computer science is particularly weak in its recruitment and retention of women, the proportion of women in university and college programs, as well as in industry, being much lower than the percentages in other scientific fields. (Cuny, 2000; Lazowska, 1999). While there are opportunities in the Professionalism course to include discussions of the need for diversity in the field, and to make students aware of their need to be tolerant of others by not discriminating, there is a potential to take positive action through the Service Learning program.

The Service-Learning program at Virginia Tech is part of the "Learn and Serve" nationwide network (http://www.learnandserve.org) which describes Service Learning as:

- … a method whereby students learn and develop through active participation in thoughtfully organized service that is conducted in and meets the needs of communities;
- … coordinated with an elementary school, secondary school, institution of higher education, or community service program and the community;
- [helping] foster civic responsibility;
- … integrated into and enhances the academic curriculum of the students, or the education components of the community service program in which the participants are enrolled;
- and provides structured time for students or participants to reflect on the service experience.

Working in school systems that are ethically representative of the local community, students from computer science programs can demonstrate that computing is not a science that is restricted to certain groups, and that all students have the opportunity to prepare for entry to a computer science program. For several years, the Virginia Tech computer science program has also organized “women in computing” days as part of the leadership development activities of the Professionalism course to provide rising seniors in high schools with an introduction to the field and the undergraduate program.

History

While the Commonwealth of Virginia did institute a requirement for all students to include a study of American History in their program of study, there has so far not been an obligation to include the study of the history of the student’s major. In many programs such a course is an elective but frequently selected course. In 1998 a Joint Task group of IFIP published a proposed syllabus for a computer history course suitable for inclusion in a computer science curriculum. (Impagliazzo et al, 1998) Like computer ethics, this topic can be offered as a separate course or incorporated into other courses. Like computer ethics, there are few faculty who have the background to enable them to complete this assignment, even though one would expect that serious students of any specialty would be familiar with the essential elements of its history. The saying “those who ignore history are doomed to repeat it”, should be sufficient motivation for any student to learn sufficient of the history of their field (or specialty). Regrettably, the archival journals of computing primarily stress the publication of reports of successful projects, and while these are interesting, to prevent the repetition of history it is often appropriate to know of the undertakings that were not triumphant and the reasons for their inadequacy. Even a study of those objects that have been made obsolete by recent developments can be important.

History can also be used as a vehicle to enhance undergraduate education. Not only can history give a human face to the technology of computing, but it can also provide the kind of anecdotal stories that can provide memory jogging artifacts that can assist in the recollection of important material. Looking at the history of the development of a computing object, it is possible to determine its time dependent perceived complexity, and therefrom to identify the point of greatest acceptance and understanding. This is the stage of development from which to commence a directed study of the object, it being the stage of greatest simplicity on which can be built the concepts of enhancement and improvement. For example, though not truly chronological, a study of the history of operating systems will reveal that the point of greatest understanding (and simplicity) was at the introduction of DOS with the IBM PC in 1981, and that from that point, studies of enhanced systems such as Windows and UNIX can be readily contrived. This ignores the early (1950s) work on monitors and supervisors such as those that led to IBSYS and OS/360.

Resource Development

The primary road-block to including humanistic elements within the computer science curriculum is faculty who have a myopic view of the field; the secondary obstacle is the lack of resources to enable the well intentioned teacher to extend their capabilities. This latter obstacle is being attacked through the recently approved National Science Foundation CITIDEL project. CITIDEL is a consortium of five universities that includes Virginia Tech, Hofstra University, Penn State University, The College of New Jersey, and Villanova University. CITIDEL will establish, operate, and maintain part of the digital library that includes computer science, information systems, information technology, software engineering, computer engineering, and other related fields. The principal investigators of the project draw upon an extensive set of experiences related to the goals of the project. These include involvement in the computing education community, expertise for gathering a broad range of resources, and technical expertise in the development and support of digital libraries. CITIDEL will manage a front-end portal to all educational activities related to computing and information technology. It will encourage and actively support content developers to make new resources available to others. It will assist users in learning not only about the resources accessible through CITIDEL, but also guide them to incorporate these resources in their own instructional environments through interactive and engaging educational activities. (Impagliazzo, 2002)

Important aspects of the CITIDEL project, and being followed extensively in the first stages of development, are the collection of resources in history and ethics. This has two purposes – to create collections in these areas of interest, and to use those collections as the vehicles for testing the viability of the services and library systems. As the work continues other specialty subjects will be added including programming languages, operating systems, and human computer interaction.

Summary

The graduates of computer science programs have a reputation of being less than worldly and somewhat disjoint from society. Their major concern is for meeting the challenge of directing a machine to do their bidding, with little regard for how the product might be used or misused. The time has come for us to graduate computer scientists who have a breadth of understanding that reaches beyond thoughtless objects and that better prepares them for involvement in the whole world of commerce and life. A holistic education can be created by designing a curriculum that involves more than computer science subjects and a variety of disciplines provided that there is a strong interrelationship between the topics. It is pointless, for example, to have students enrolled in a philosophical logic unit if that information is not put to use in their own program. However a curriculum that includes humanistic elements within the computer science program provides immediate application and relevance that cannot be achieved by a non-computer science faculty member.

It will take time, but if we can start now developing future faculty who have received a broad education, then there is the hope that a holistic computer science curriculum will be available to the next generation of students.

References

ACM/IEEE-CS. (2001) “Computing Curricula 2001 – Computer Science Volume”, Association for Computing Machinery and IEEE Computer Society, 15 December 2001, http://www.acm.org/sigcse/cc2001/ (last accessed 2002/02/05)

Aspray, William, and Andrew Bernat. (2000) “Recruitment and Retention of Underrepresented Minority Graduate Students in Computer Science”, Computer Research Association, Report of a workshop, 4-5 March 2000, 51 pp., http://www.cra.org/reports/r&rminorities.pdf (last accessed 2001/10/16)

Bowyer, Kevin. (1998) “Ethics and Computing”, Resources from NSF Workshops, University of South Florida, 1998-1999, http://www.cse.nd.edu/~kwb/nsf-ufe/index.html (last accessed 2001/10/11)

CSAB. (1986) “Computer Science Accreditation”, Accreditation Board for Engineering and Technology (ABET), http://www.abet.org/cac1.html (last accessed 2001/10/11)

Cuny, Janice, and William Aspray. (2000) “Recruitment and Retention of Women Graduate Students in Computer Science and Engineering“, Report of a workshop, June, 2000, http://www.cra.org/reports/r&rwomen.pdf (last accessed 2001/10/16)

Impagliazzo, John, Martin Campbell-Kelly, Gordon Davies, John A. N. Lee, and Michael R. Williams. (1998) “History in the Computing Curriculum”, IFIP TC3/WG 9.7 Joint Task Group, 1998 October, 16 pp., http://www.hofstra.edu/pdf/CompHist_9810rprt.PDF (last accessed 2001/10/16)

Impagliazzo, John, and J.A.N. Lee. (2002) “CITIDEL: A Gateway for Information Technology”, submitted to IS2002 Conference, October 2001.

Lazowska, Edward D. (1999) “CRA Testimony on the under-representation of women and minorities in computing”, Testimony before the Commission on the Advancement of Women and Minorities in Science, Engineering, and Technology Development (CAWMSET), October 6, 1999, http://www.cra.org/Policy/testimony/lazowska-5.html (last accessed 2001/10/16)

Lee, J.A.N. (2002) “Professionalism in Computing”, A Web-based course, http://courses.cs.vt.edu/professionalism/ (last accessed 2002/05/013)

Biography

Dr. John A. N. (JAN) Lee is a Professor of Computer Science at Virginia Tech, and is a member of the Center for the Study of Science in Society. Recently designated as the “dean of computer historians” by the *New York Times,* he served as the Editor-in-Chief of the *IEEE Annals of the History of Computing* for nine years and is the author of Computer Pioneers, an encyclopedia of biographies of computer pioneers throughout the world. He serves as the representative of the IEEE Computer Society to the International Federation for Information Processing; he is also a Trustee of IFIP. His current work centers on the development of a digital library for Computer Science education through the CITIDEL project and preparation for his forthcoming retirement.