

Enhancing the Social Issues Components in our Computing Curriculum: Computing for the Social Good

Mikey Goldweber
Xavier University
Cincinnati, Ohio USA
mikeyg@cs.xu.edu

Joyce Currie Little
Towson University
Baltimore, Maryland USA
jclittle@towson.edu

Gerry Cross
Mount Royal University
Calgary, Alberta Canada
gcross@mtroyal.ca

Renzo Davoli
University of Bologna
Bologna, Italy
renzo@cs.unibo.it

Charles Riedesel
University of Nebraska, Lincoln
Lincoln, Nebraska USA
riedesel@cse.unl.edu

Brian R. von Konsky
Curtin University
Perth Australia
B.vonKonsky@curtin.edu.au

Henry Walker
Grinnell College
Grinnell, Iowa USA
walker@cs.rinnell.edu

ABSTRACT

The acceptance and integration of social issues into computing curricula is still a work in progress twenty years after it was first incorporated into the ACM Computing Curricula. Through an international survey of computing instructors, this paper corroborates prior work showing that most institutions include the societal impact of ICT in their programs. However, topics often concentrate on computer history, codes of ethics and intellectual property, while neglecting broader issues of societal impact. This paper explores how these neglected topics can be better developed through a subtle change of focus to the significant role that ICT plays in addressing the needs of the community. Drawing on the survey and a set of implementation cases, the paper provides guidance by means of examples and resources to empower teaching teams to engage students in the application of ICT to bring about positive social outcomes – *computing for the social good*.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education – *curriculum, accreditation*.

K4 [Computers and Society]

K5 [Legal Aspects of Computing]

K.7.4 [The Computing Profession]: Professional Ethics

General Terms

Human Factors, Legal Aspects, Reliability, Security.

Keywords

Societal impact, ethics, curriculum, professional societies.

1. INTRODUCTION AND MOTIVATION

Most academic programs in Information and Communications Technology (ICT)¹ expect that graduates will be prepared for practice as emerging professionals in their discipline [1, 2]. In addition to possessing significant technical skills and knowledge, graduates should be able to assess the societal impact of their work, be prepared to commit to standards of professional ethics, and have the life skills necessary to undertake on-going professional development in their discipline. The development of these professional skills is not a new component in international computing curricula. The ACM/IEEE Joint Task Force Computing Curricula 91 articulated these goals nearly 20 years ago:

Undergraduates also need to understand the basic cultural, social, legal, and ethical issues inherent in the discipline of computing. They should understand where the discipline has been, where it is, and where it is heading. They should also understand their individual roles in this process, as well as appreciate the philosophical questions, technical problems, and aesthetic values that play an important part in the development of the discipline.

¹ We use ICT as a broad term that encompasses a range of computing disciplines including computer science, software engineering, information technology, information systems and computer engineering.

Students also need to develop the ability to ask serious questions about the social impact of computing and to evaluate proposed answers to those questions. Future practitioners must be able to anticipate the impact of introducing a given product into a given environment. Will that product enhance or degrade the quality of life? What will the impact be upon individuals, groups, and institutions?

Finally, students need to be aware of the basic legal rights of software and hardware vendors and users, and they also need to appreciate the ethical values that are the basis for those rights. Future practitioners must understand the responsibility that they will bear, and the possible consequences of failure. They must understand their own limitations as well as the limitations of their tools. All practitioners must make a long-term commitment to remaining current in their chosen specialties and in the discipline of computing as a whole.

Computing curricula 1991 [3]

Applications of computing and communications technologies have changed the world in profound ways and have generated complex social issues related to their use [4]. For example, computing technologies have facilitated:

- improved healthcare arising from medical imaging and health informatics [5],
- business continuity and access to news and information following emergencies caused by natural disasters and acts of terrorism [6],
- applications of Global Positioning System (GPS) such as surveying and navigation,
- genome research leading to medical advances and specific pharmacology, and
- many other scientific advances that would have been difficult or impossible to achieve without high performance computing and computer-based simulation and visualization.

Computing technology has also had an impact on fundamental rights in the modern world:

- informed democratic engagement as a result of public access to information, virtual town meetings and Internet voting [7],
- creation and maintenance of a neutral Internet that provides a democratic means of communication, even for people under totalitarian regimes [8-10], and
- creation of independent, international, cultural common grounds both through free software and collaborative applications such as Wikipedia [11-13].

While system developers may envision positive goals for their products, these new systems also introduce new societal and cultural challenges as system designers attempt to strike a balance between competing aspects of privacy, security, and usability. For example, storing the medical history of patients on-line can improve the quality of health care, particularly when multiple health care providers are involved. However, personal information on genetic predispositions for developing inherited diseases could have negative financial consequences for patients if this information were to become available to insurance companies. Ensuring the security and appropriate use of private data is inherently the responsibility of those designing and implementing

information systems, even though they may not be the policy makers.

Some aspects of societal impact are well established in many ICT academic programs [14]. These include the history of computing, professional ethics, computer crime, security, and intellectual property. Other aspects of societal impact are often not included or are given minimal treatment. These include cultural issues, accessibility issues, the impact of the free-open source software movement, computing and public policy, green computing, and computing for sustainability. Incorporating these other aspects into the curriculum would open up many new opportunities to inspire students with the social relevance of computing.

More specifically, academic programs have a unique opportunity to position ICT as a force for positive world change by developing graduates who produce products and provide services that are of value to the community. Examples might include the development of free-open source software, software for humanitarian purposes, software contributing to the solution of pressing social problems such as global warming, leveraging social networking tools to help overcome the digital divide, the creation of computer art, music, and poetry, and assisting indigenous peoples to maintain their cultural identity.

A curricular focus on the societal impact of ICT has the potential to attract a larger cohort of students and to improve retention rates, as students begin to appreciate the tangible and practical social impact of ICT. For example, Buckley, Nordlinger and Subramanian [15] argue that the “*4x rate of graduation in Social Sciences as compared to Computer Sciences is due at least in part because of students’ desire to have a societal impact.*”

This suggests that academic institutions should strive to demonstrate to prospective and current students that ICT professionals contribute to an international community by cooperatively working towards solving problems of global and international significance. To attract prospective students who want to make a contribution to society, academic institutions should promote and publicize the success of graduates who are employed on IT-related projects at the local hospital or for non-profit organizations like the International Red Cross, Greenpeace, or the World Wildlife Fund. When prospective students choose to study ICT, they should find opportunities to explore the societal impact of ICT emphasized at least to the extent recommended by the ACM Computing Curricula. Where possible, opportunities to work on capstone projects that serve the public good should be favored over projects with a strictly commercial focus, provided they develop equivalent technical and professional outcomes. In this spirit, this paper provides guidance on how institutions can introduce or enhance their treatment of societal impact issues.

2. BACKGROUND

After the ACM/IEEE Curricula 1991 report recommended that social issues be included in the curriculum, a National Science Foundation grant entitled Project ImpactCS supported the work of a large team to produce detailed instructional resources for use in computer science departments. Huff and Martin [16] articulated the importance of these topics in an article that described some of the dramatic impacts of computing. A later report documented the recommendations of the project ImpactCS group, which included a stronger emphasis on social issues in the curriculum [17].

The ITiCSE conferences of the ACM provided opportunities for collaborative endeavors in an international setting, and in 1997 an ITiCSE working group produced an array of exercises that could be used to incorporate social issues as modules within computing courses [18]. Other ITiCSE working groups followed: historical perspectives on the computing curricula [19]; a report offering ways to incorporate professional issues [1]; and, incorporating cultural issues [20].

The following subsections describe the state of the teaching of social issues today by considering the professionalism, societal responsibility and ethical conduct expectations placed on practicing ICT professionals by accrediting bodies, the ACM curricula recommendations for the inclusion of societal issues, and prior work surveying how these recommendations have been implemented.

2.1 Accreditation and Professional Obligation

The International Professional Practice Partnership (IP3) initiative of the International Federation for Information Processing (IFIP) is committed to fostering a global IT profession based on standards and local regulation. Program partners share the expectation that IT professionals have the skills and knowledge necessary to operate with a high level of autonomy and responsibility, remain current in their discipline through on-going professional development, and adhere to an established code of ethics. In part, the intention is that this will lead to enhanced levels of service, a high level of public confidence, and IT practitioners who are more socially responsible [21, 22].

Similarly, the Seoul Accord is a multi-lateral agreement among organizations responsible for the accreditation of ICT programs in various jurisdictions [23]. Signatories demonstrate the equivalency of the programs that they accredit, including in the development of graduate attributes related to professional responsibilities to society, ethics, and lifelong learning. In particular, graduates must be able to:

- *understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice,*
- *understand and commit to professional ethics, responsibilities, and norms of professional computing practice, and*
- *recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.*

Seoul Accord [24]

Seoul Accord signatories currently include the Accreditation Board for Engineering and Technology (ABET) in the United States, the Accreditation Board for Engineering Education of Korea (ABEEK) in the Republic of Korea, the British Computer Society (BCS) in the United Kingdom, the Hong Kong Institute of Engineering (HKIE) in Hong Kong, China, the Institute of Engineering Education Taiwan (IEET) in Taipei, the Australian Computer Society (ACS), the Canadian Information Processing Society (CIPS), and the Japan Accreditation Board for Engineering Education (JABEE) [25].

Accrediting bodies generally do not define curricula. Instead, their role may be better characterized as providing a quality assurance check to ensure coherence amongst recognized professional

attributes and capabilities, holistic course design and implementation, and stated program objectives. For example, the ACS recommends that Australian educational programs adopt recognized international curricula to develop the necessary skills and knowledge for a specific ICT discipline. This is on top of the ACS's Core Body of Knowledge that includes professionalism, ethics, and societal and legal issues that the ACS deems to be common to all ICT disciplines [26].

2.2 The ACM Curricula

The ACM defines five computing curricula for bachelor-level degree programs (CC-2005) [27]. These are curricula for computer science (CS-2008) [28, 29], information technology [30], information systems (IS-2010) [31], software engineering [32], and computer engineering [33].

The core ACM curriculum for computer science identifies a total of 16 hours related to social and professional issues as follows [29]: history of computing (1 hour), social context (3 hours), analytical tools (2 hours), professional ethics (3 hours), risks (2 hours), intellectual property (3 hours), and privacy and civil liberties (2 hours). The curriculum recommendations also define elective bodies of knowledge in security operations, computer crime, economics of computing, and philosophical frameworks. The core and elective topics are described briefly in Appendix A.

Similar recommendations are found in the ACM computing curricula for other ICT disciplines [30-33].

In an analysis of the ACM curricula, the Joint Task Force used a six-point scale to compare the emphasis placed on knowledge areas in the ICT disciplines. The scale ranges between 0 (lowest emphasis) and 5 (highest emphasis) [27]. The emphasis on professional, legal, ethical, and societal issues in computer engineering, information systems, and software engineering was deemed to be between a minimum of 2 and a maximum of 5. In computer science and information technology, the minimum was 2 and the maximum 4. These values represent the minimum specified in the curriculum recommendations and the maximum that could be expected in a program that follows the recommendations. Thus, the ACM computing curricula places strong emphasis on professional, legal, ethical, and societal issues as a core knowledge area in each of the five ICT disciplines [26, 27].

2.3 Implementing the ACM Curricula

There is no clear consensus on how professional and societal issues should be implemented in the curriculum.

For information systems, IS-2010 includes professional issues in the Information Systems knowledge area. This knowledge area encompasses the societal context of computing, legal issues, intellectual property, and privacy. These topics are covered in a range of courses including Foundations of Information Systems, IS Project Management, Systems Analysis and Design, Innovation and New Technologies, and IT Security and Risk Management.

The CS-2008 computer science curriculum recommends a single course on professional, ethical, and societal issues, plus complementary modules that integrate these topics into technical courses as appropriate. The report describes a trade-off regarding where the standalone course should be positioned within the program structure:

Having the course at the lower level

1. *Allows for coverage of methods and tools of analysis... prior to analyzing ethical issues in the context of different technical areas.*
2. *Assures that students who drop out early to enter the workforce will still be introduced to some professional and ethical issues.*

On the other hand, placing the course too early may lead to the following problems:

1. *Lower-level students may not have the technical knowledge and intellectual maturity to support in-depth ethical analysis. Without basic understanding of technical alternatives, it is difficult to consider their ethical implications.*
2. *Students need a certain level of maturity and sophistication to appreciate the background and issues involved...*

CS-2008 [29]

2.4 Prior Surveys on Implementation

In an attempt to characterize how departments implement curricula recommendations and address accreditation requirements for professional, ethical, societal, privacy, and discipline impact issues, Homkes and Strikwerda [34] conducted an analysis of ABET accredited undergraduate programs in information technology, computer science, and information systems. The analysis utilized a keyword search using a set of descriptors on the web pages available for each university in the sample. Findings showed that all programs incorporated professional, legal, privacy, social, and discipline impact issues *in situ* in a number of technical courses.

All of the nine randomly sampled computer science programs included a standalone course, with seven of these taught within the department and two by an outside entity. One computer science program placed the course in the second year of study, six programs placed it in the third year, and two programs placed it in the fourth year.

Only two of the nine information systems programs in the sample included a standalone course. The survey's authors suggest that this was influenced by the IS curriculum recommendations positioning professional and societal issues in the context of other IS subjects. The two IS programs in the study with a standalone course placed it in the final year of study. The same course was also required of computer science majors at those institutions.

Of the nine information technology programs in the study, four included a required standalone course and one had two elective courses.

Barroso and Melara [35] conducted a similar study, but based their analysis on a survey of academic staff, predominantly from the California state universities. In a sample of 69 academics, over 40% reported being in computer science, almost 30% in another ICT field, and the remainder were engineers. Just over 82% reported that they were involved in teaching ethics to some extent. Of these, 54.4% taught computer ethics in a standalone course and just over 20% taught it in the context of another technical subject.

A more comprehensive survey conducted by Spradling, Soh, and Ansoerge in 2005 found, in a large sample stratified by region and

cohort size, that a majority of U.S. computer science programs included professional, societal, and ethical issues (87%, n=220) [14]. In institutions with larger student numbers, 28.5% (n=95) used only a standalone course and 36.8% used a standalone course in combination with modules integrating professional and societal content across the program. Of those institutions with a standalone course, 17.9% offered it during the first or second year of study, and 78.6% offered it in the third or fourth year.

Spradling, Soh, and Ansoerge also found that institutions with fewer contact hours devoted to professional and societal content or those with smaller enrollments were more likely to distribute professional and societal content across courses in their program [14]. To some extent, this finding reflects the view of Martin [36] who recommends adopting an integrated solution for teaching professional and societal issues in those cases where a program must be constrained to a single approach. That recommendation was based on the argument that doing so best emphasizes the interrelationship between the technical, ethical, and social aspects of computing.

Institutions that do not teach professional and societal issues tend to have smaller enrollments [37]. Their most cited reasons were no room in the curriculum or the lack of staff qualified to teach professional and societal topics.

3. SURVEY RESULTS

In June 2010, prior to the ITiCSE 2010 Conference, the authors published a link to a short survey on the SIGCSE members ListServe, with the intent of obtaining a rough idea of the state of the teaching of social issues in undergraduate ICT programs. 86 of the 1057 ListServe recipients responded.

While this is a better than expected response rate from this venue, the sampling is certainly not scientific and few, if any, conclusions can be drawn from the results. It is assumed that the respondents over-represent computer science programs compared to other ICT disciplines and that they have an interest in teaching social issues. Most respondents are from American institutions, with only about half a dozen from other countries. Nevertheless, the results corroborate commonly held assumptions. In particular, they are consistent with the survey of Spradling, Soh, and Ansoerge. In view of the limited and non-representative data, all percentages are rounded to the nearest 5%.

3.1 Summary Results

Nearly all respondents (95%) indicated their schools offered a computer science major. Around 25% of the respondents reported other majors such as computer engineering or information systems. The sizes of programs represented a good spread from very small to large. The survey included a question related to demographics involving women. Findings relating percentages of women and various other factors are generally consistent with more detailed studies done elsewhere.

Only a few schools (5%) do not cover social, ethical, and/or professional issues. Of those that do something, about 55% recognize some connections with ACM/IEEE guidelines, but 30% of the respondents reported they were not very familiar with these guidelines. The most common topics discussed include (in descending order of popularity) professional ethics, social context, history, intellectual property, privacy and civil liberties, risks, and computer crime. Of the respondents who reported being familiar

with the ACM/IEEE guidelines, 75% felt the guidelines specify an appropriate level of coverage of social issues and the remainder was split, either believing it is too little or too much.

For schools that reported including this material, the courses involved were more likely to be offered within the department itself, but connections with other departments and courses were not uncommon. Also, somewhat more schools reported using a mix of standalone courses and topics integrated with technical courses than schools that use either approach by itself. About 85% of the respondents believe that computing faculty should be involved in teaching social issues, but about 35% also believe that faculty outside computing should be involved as well.

Substantial variation was reported regarding the level at which students encounter this material, even within a single institution. Some reported that students study it several times at differing levels of their undergraduate experience.

Of those institutions that reported covering social issues, roughly 35% have no mechanism for assessment of this material, while about 45% have formal assessment procedures (e.g. with grades) on at least most of the work. The remaining 20% are assessed on only portions of the work.

In presenting social topics, a traditional format involving lectures was found to be more common than various active-learning formats.

While 65% of those responding to the survey believe students should have a good understanding of the ACM Code of Ethics, even more want students to actively engage with this material, including through activities such as writing impact statements or articulating potential consequences of proposed systems.

Respondents to this survey had considerable experience teaching about the social issues of computing. 70% had included these topics in their courses, 85% had high or moderate confidence that they could teach this material, and faculty currently teaching it were more confident than those who were not.

Many faculty indicated an interest in having additional case studies and discussion starters to enhance their teaching of this material. At a lower level of interest, faculty also mentioned videos and outside speakers as desired resources to further enhance their courses.

3.2 Cross-tabulation Results

The data were cross-tabulated by program size, percentage of women, knowledge of the respondent, and degree of assessment. Only results that showed a significant correlation are presented here. Because of the lack of confidence in the results, no percentages are given.

3.2.1 Cross-tabulation by Program Size

When cross-tabulated by program size, some patterns emerge:

- Institutions graduating fewer than 25 per year tend to have a higher percentage of women than institutions graduating more than 50.
- The smallest institutions with fewer than 10 graduates per year were impacted more severely by the recent downturn in enrollments and are least likely to be recovering.

- The very smallest and very largest institutions are more likely to expose students to social issues inside the department.
- There is a negative correlation between size and formally assessing students on social issues.
- The smallest institutions are most likely to provide a combination of standalone and integrated components.
- The respondents at the smallest institutions were more likely to rate their own knowledge of social issues as moderate or high.

3.2.2 Cross-tabulation by Percentage of Women

Another cross-tabulation was done for number of female students. In addition to again showing the tendency of small institutions to graduate a higher percentage of women, there are the following patterns. Respondents at institutions with a higher percentage of women tend to:

- cover social issues in a combination of standalone and integrated components,
- prefer less involvement by other departments,
- include social issues in their own teaching,
- have higher confidence in their ability to teach social issues, and
- rank their knowledge of social issues as moderate and high.

3.2.3 Cross-tabulation by Knowledge of Respondent

Cross-tabulating by the respondents' knowledge of social and ethical issues in computing yielded a few interesting and expected results. The most knowledgeable respondents:

- tend to come from the smallest institutions, and tend to cover social context, risks, intellectual property, privacy, civil liberties, computer crime, and philosophical frameworks more than less knowledgeable respondents,
- also are most likely to believe social issues should be taught inside the department, have done so themselves, have the highest confidence, and are the primary instructor in their departments, and
- tend to endorse having the most extensive objectives related to teaching social issues, and are least likely to advocate limited objectives or no coverage at all.

3.2.4 Cross-tabulation by Degree of Assessment

In addition to again showing more formal assessment in smaller programs, the cross-tabulation by degree of assessment yielded the following:

- Students taught inside the department tend to be assessed the most.
- There is little or no correlation between the degree of assessment and the percentage of women or effect of the enrollment downturn.
- Assessment is more likely for the topics of analytical tools, intellectual property, privacy, civil liberties, computer crime, and philosophical frameworks.
- The degree of student assessment is expectedly high in standalone courses compared to integrated and embedded approaches.
- Experiential and collaborative learning experience also tended to be assessed.

- The degree of student assessment is high when the respondent's confidence in teaching is high, and also when the respondent's knowledge of social issues is high.

3.3 Survey Conclusions

While many of these individual results may be more or less interesting, they may not be very relevant to this paper. However, taken as a whole, the survey shows that there is still much uncertainty in what social issues include, how they relate to computing, what pedagogies to use, and where and when they can best be presented. There is an impression that the infrastructure is not mature, lacking adequate texts, supporting materials, and interest organizations with publications. There appears to be general acknowledgement that social issues are an important component of the curriculum, but lack of confidence in how they can be incorporated and taught. One can demonstrate the relative state of maturity by selecting random technical topics from the ACM Computing Body of Knowledge, and compare what most can say about them to what they can say about any of the social issues topics.

4. GETTING STARTED AND GOING FURTHER: INITIATING AND ENHANCING THE TEACHING OF SOCIAL ISSUES

In this section, the report shifts to strategies and resources for incorporating social issues in computing into the curriculum. It contains suggestions that will be of use for instructors who are teaching social issues for the first time, as well as for those who are at any stage of implementation and would like to do more. Special attention is given towards the goal of fostering computing for the social good. Appendix B expands upon this discussion by describing how social issues have been incorporated into existing and new courses at several colleges and universities.

4.1 Three Starting Questions

When considering how to begin or expand the coverage of social issues, one needs to consider three key questions:

1. **Where can new material be inserted or existing material refined?** Computing courses typically cover large amounts of material, so an instructor or department will likely need to re-conceptualize the existing content in order to add components on social issues.
2. **Where can resources be found and support gathered?** If an instructor feels isolated in initiating or expanding coverage of social issues, finding support inside or outside the institution can be especially beneficial.
3. **How can students be actively engaged in the material?** Throughout their education students have heard that they should help others and act responsibly, but theory often is disconnected from action, and the objective in teaching social issues is to affect the behavior of students in their careers.

4.2 Where Might New Material Fit?

One approach is to assess the impact of proposed curriculum changes on stakeholders such as students and computing faculty, and its impact on prerequisites for other courses, major programs, accreditation, industry, and the community. If a topic is added or

if more time is spent on an existing topic, what other topic might be diminished in coverage? Or can all or part of a course be reformulated so that coverage of social issues can be integrated with the other topics?

When considering strategies for incorporating the coverage of social issues into an ICT curriculum, several approaches are possible:

Standalone course(s): In this approach the traditional ICT technical content courses remain unchanged and all coverage of social and professional issues is segregated into a separate course or courses. While potentially simple to implement, there is the risk that students will perceive these topics as being orthogonal to their primary discipline-based studies.

Integrated Modules: Essentially, what could have been a standalone course is distributed across many modules throughout the ICT curriculum. An advantage of this approach is that the relevance of social and professional issues is repeated throughout the students' academic careers. Furthermore, unlike with the standalone course, this approach requires a commitment to teaching social and professional issues by a wide cross-section of the ICT instructors.

Embedded: This approach, which is orthogonal to the above, attempts to weave the coverage of social and professional issues throughout the ICT curriculum by means of judicious use of examples and case studies. Three examples of this approach follow.

- In an operating systems course, the Morris Worm case at Cornell University [38, 39] can be used as an example of inter-process communication. This 1988 event illustrates many important technical matters, but also can be used to discuss social impact and professional responsibility.
- In a software engineering course, reverse engineering of proprietary software and the free-open source philosophy can be discussed.
- In a database course, instead of using a banking or airline reservation system, one might consider a disaster relief management application. As with either banking or airline reservation systems, this example application can be initially conveyed in a simple form and can grow in complexity as needed for topic coverage. Furthermore, this example allows one to cover important professional issues such as data security, privacy, and ethical behavior by those with data access. One can essentially "embed" a short conversation related to one or more social or professional concerns with almost any technical concept.

In these cases, the instructor has a choice regarding what examples to use and can introduce social issues into the course by choosing examples that illustrate the broader context. However, unlike integrated modules and standalone courses, in which societal impact is an explicit part of the assessment objectives, topics covered via an embedded strategy present challenges for assessment as goals for social issues are merged with other learning outcomes.

Experiential: Another popular strategy involves "learning by doing" and sometimes is called service learning.

- Instead of a summer cooperative internship experience, consider encouraging students to spend a summer working

on a Humanitarian-Free and Open Source Software (H-FOSS) project at one of the H-FOSS summer institutes [40].

- A project-based software engineering course could develop a project to aid a non-profit community-based partner instead of a fictitious project or one for a university office or local industry partner. Service learning in computing is another experiential option: e.g. having students teach graphical programming using Alice or Scratch to disadvantaged youth.
- All software projects, including those with a commercial focus, can utilize Software Development Impact Statements (SoDIS) to assess the societal impact of a software project on stakeholders and to mitigate potential negative consequences [41]. Similar to an environmental impact statement, SoDIS provides an ethical framework in which project teams work to identify stakeholders, the risks that the project presents to them, and potential mitigating strategies.

The above approaches are neither exhaustive nor independent. One may choose an embedded approach to cover the history of computing or security concerns, while selecting an integrated module to cover professional ethics or professional codes of conduct.

Finally, change need not be revolutionary. An evolutionary approach can yield meaningful benefits with a modest time commitment and marginal impact on other topics.

4.3 Implementation Cases

Pragmatically, many ICT departments have already incorporated social issues in computing into existing and new courses. Appendix B provides examples for a variety of institutional types. In particular, the appendix describes seven approaches and settings with the following characteristics:

1. A large department at a large public university seeking to fulfill requirements for accreditation. (Towson University)
2. A large department at a large public university wishing to improve on the professionalism of its graduates. (Curtin University)
3. A small department at a small private college wanting to connect with its liberal arts mission. (Grinnell College)
4. A small department at a medium sized private university that implements an experiential learning opportunity by connecting students at the introductory level with economically disadvantaged students in the community. (John Carroll)
5. A large department at a large public university utilizing the Bologna Agreement approach to create multiple curricular student paths. (The University of Melbourne)
6. A medium sized department at a medium sized private university with an experiential learning opportunity that has students acting as consultants with local non-profit community partners. (Carnegie Mellon)
7. A large department at a large public university with a specialized graduate program focusing on computing for the social good. (University of Bologna)

The Appendix B narrative is by no means exhaustive, but it does illustrate a variety of approaches that have been tried at a range of institutions and that can provide insights for new or revised implementations.

4.4 Resources and Support

The process of adding, expanding, or refining material on social issues can seem daunting without help. Assistance can come from at least two sources: people and materials.

Support from people can come from one's own department, from other departments, or from outside the institution. Within an institution, one strategy is to simply make known what you want to do. By broadcasting your interests, a colleague with similar interests might contact you. As a second strategy, you might be more proactive and contact promising departments such as philosophy, education, or sociology. Outside the institution, there are people in the community who would be pleased to give guest lectures. There is also a group of SIGCSE members with a strong interest in social issues.

Alternatively, one might seek out others who have relevant experience. These individuals may be ICT colleagues or local colleagues from other departments. Also many institutions have dedicated support units to assist faculty in using experiential learning or engaging with community partners.

Materials can be relatively informal or highly structured. Some places to look for resources on social issues related to computing are:

- the bibliography of this working group report;
- news articles [42-44];
- articles to provide some depth in an issue, such as privacy or security [45-47];
- case studies such as the Morris Worm case at Cornell University in an operating systems course [38, 39];
- structured readings (using a textbook is easier than integrating miscellaneous readings) [4].

Other valuable resources are:

- Special Interest Group on Computers and Society (SIGCAS) - an ACM-sponsored special interest group on Computers and Society. The professional and ethical aspects of computing are particular strengths of this group [48].
- Humanitarian Free and Open Source Software (H-FOSS) - an organization for involving undergraduate students in building free and open source software that benefits the community. In addition to running summer institutes for interested students and faculty mentors, the organization has an active member base to assist those who wish to utilize an H-FOSS project in their curriculum [40, 49, 50].
- Socially Relevant Computing (SRC) - a new initiative being introduced by Microsoft Research in conjunction with two academic partners (University at Buffalo and Rice University). While at the time of this report SRC has no resources or programming to offer, one hopes that this initiative will soon grow in value [51].
- Computer Professionals for Social Responsibility (CPSR) - a public interest alliance of computer scientists and others promoting the responsible use of computer technology [52, 53].
- The Risks Digest – a forum of ICT risks to the public. The Forum on Risks to the Public in Computers and Related Systems arose from the Association for Computing Machinery's Committee on Computers and Public Policy, under the direction and continuing guidance of Dr. Peter G. Neumann, Moderator. This resource provides the public with a place to submit information about problems that have

arisen in computing. The Risks Forum is a moderated digest, with a USENET equivalent as comp.risks. This resource can be used in educational settings to provide links to information about computer-related social and ethical problems and concerns [54].

- IEEE Technology and Society Magazine – a publication of the IEEE on topics related to technology and society [55].

4.5 Engaging Students

Students learn the expected responses regarding human behavior during the early years of their education. Since they know what they are supposed to say in many settings, an instructor is unlikely to have much impact by simply restating these principles for computing applications. Rather, students need to be challenged to see the impact of computing from different perspectives, confronted with complexities that they have not considered previously, and engaged in situations having unexpected consequences or undesired behavior. For example, in an introductory programming course, students may consider integer overflow to be inconsequential because programs that do not protect against it will work in most cases. Their thinking might change if they knew that integer overflow in a database caused the cancellation of flights for about 30,000 Comair passengers in December 2004 [56]. An (embedded) discussion of social issues, or better yet, an experiential learning component provides excellent opportunities for engaging and challenging students, and need not be more of the predictable moralizing they likely have heard regularly for the past decade or more.

The efficacy of strategies for introducing professional and social issues into the curriculum is not well understood. Students may be able to recite a professional society code of ethics and write an essay on how it would inform the decision-making process in the context of a hypothetical case study. However, do they actually employ this framework in practice as an emerging professional? Do they consider professional and societal aspects of the curriculum to be an integral part of their discipline, or merely an orthogonal component required either for accreditation purposes or to satisfy an individual professor's focus?

As suggested by Buckley (2008) student engagement, in addition to retention and participation by women and minorities, may be improved the more students perceive the professional and societal aspects as part of the fabric of being an ICT professional. Indeed, as computing for the social good permeates ICT curricula, students may elect to study ICT out of a desire to have a societal impact.

4.6 Professional Development

As instructors expand coverage of social issues in their courses, they may need to use unfamiliar pedagogy. Active learning techniques are often used to engage students in social issues. Lectures, group discussions, cases studies, and readings are other popular pedagogies for introducing social issues into the computing curricula [14, 35]. Specific learning experiences that develop professionalism, social issues and ethics are well documented in the literature [18, 57-59]. In particular, project work is often suggested as a means to highlight social issues [1, 15, 41, 60].

Academic staff in computing departments may require professional development assistance in applying some of these

pedagogies, in developing rubrics for assessing student papers and other in-class activities such as debates on social issues, and in developing experiential learning opportunities. Campus resources such as workshops offered by teaching development centers, education faculty or the library are often available to meet some of these needs. As well, opportunities for professional development can be found at regional meetings and conferences of organizations such as SIGCSE and SIGCAS.

4.7 Acceptance for Tenure, Promotion, and Merit Pay

At any institution, rethinking courses and adding content on social issues will be considered for tenure, promotion and merit pay, but the value will vary. Experiential learning opportunities off campus, in particular, require significant time investments to both initiate and maintain. If applicable, a faculty member should discuss these issues with the department chair, the dean and members of the tenure and promotion committee.

5. CONCLUSIONS

Nearly 20 years ago the ACM articulated the need to include coverage of both professional and social issues in ICT curricula. Through efforts such as ImpactCS, SIGCAS, and the reports of three previous ITiCSE-based working groups, many ICT programs now include some coverage of these issues. In fact, the results of the admittedly non-scientific survey conducted by this working group in 2010 corroborate previous findings that almost 95% of the surveyed ICT programs provide some coverage of social, professional and ethical issues.

What is also clear from these surveys, including the one we are reporting on, is that much more needs to be done. In particular, there is a need for greater coverage of cultural issues, accessibility issues, the digital divide, the impact of the FOSS movement, and computing and public policy, among others. To assist in this the authors provide not only a set of resources and strategies to consider, but a set of implementation showcases, each of which provides a detailed description of how a particular institution is trying to meet this challenge.

Our intention is for the implementation showcases to not only aid those wishing to enhance the coverage of social issues in their curriculum, but especially to foster the growing realization of the advantages of computing for the social good. There is evidence to suggest that students do not believe an ICT career path will allow one to have a positive societal impact and, therefore, do not pursue an ICT major. In particular, some believe that this is especially true for women. Hence, the implementation showcases provided highlight programs that give specific emphasis on computing for the social good.

There are a number of future directions suggested by this work. The first is a more exhaustive, scientific, international survey to gain a better understanding of which professional and social issues are being adequately covered in ICT curricula and which are not. This survey may potentially be able to highlight best practices in this area as well. The other direction suggested by this work involves strategies for repositioning people's views, including ICT departments themselves, of both ICT degrees and ICT career paths with respect to societal impact: computing for the social good. A goal of this direction may one day be a freshman cohort where a majority of the students indicate that

they selected an ICT major not to become a games programmer, to be rich, or because they like computers, but because they were mathematically/technically oriented problem solvers who wanted to have a positive impact on society.

6. ACKNOWLEDGEMENTS

We wish to acknowledge the cooperation of Raj Gopalan (Curtin University), Tele Tan (Curtin University), Janet Davis (Grinnell College), Alistair Moffat (University of Melbourne), Joe Mertz (Carnegie Mellon), and Linda Seiter (John Carroll University) for providing the implementation cases found in Appendix B. Additionally, we wish to acknowledge Randy Connelly (Mount Royal University) who also provided an implementation write-up, which due to space constraints could not be included in the Appendix. We also thank Lisa C. Kaczmarczyk for her assistance with designing the survey. Finally, we wish to acknowledge Donald Joyce (Unitec) who initially proposed this working group and without whose dedication to the ideals expressed in this report it would not have been possible.

7. REFERENCES

- Little, J.C., et al., *Integrating professionalism and workplace issues into the computing and information technology curriculum: report of the ITiCSE'99 working group on professionalism*, in *Working group reports from ITiCSE on Innovation and technology in computer science education*. 1999, ACM: Cracow, Poland. p. 106-120.
- von Kinsky, B.R., *Defining the ICT Profession: A partnership of stakeholders, keynote address*, in *Proceedings of the 21st Annual Conference of the National Advisory Committee on Computing Qualifications*. 2008: Auckland New Zealand. p. 15-22.
- Tucker, A.B., et al., *Computing curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force*. 1991.
- Rosenberg, R.S., *The social impact of computers*. 3rd ed. 2004: Elsevier Academic Press.
- Raghupathi, W. and J. Tan, *Strategic IT applications in health care*. Commun. ACM, 2002. **45**(12): p. 56-61.
- Arduini, F. and V. Morabito, *Business continuity and the banking industry*. Commun. ACM, 2010. **53**(3): p. 121-125.
- Dawes, S.S., et al., *Some assembly required: building a digital government for the 21st century*, in *Proceedings of the 2002 annual national conference on Digital government research*. 2002, Digital Government Society of North America: Los Angeles, California. p. 1-39.
- Zittrain, J. and B. Edelman, *Internet filtering in China*. IEEE Internet Computing, 2003. **7**(2): p. 70-77.
- Crandall, J., et al., *Conceptdoppler: A weather tracker for internet censorship*, in *14th Conference on Computer and Communications Security*. 2007.
- Mustafaraj, E. and P. Metaxas, *From Obscurity to Prominence in Minutes: Political Speech and Real-Time Search*, in *WebSci 10: Extending the Frontiers of Society On-Line*. 2010: Raleigh, NC USA.
- Boldrin, M. and D.K. Levine, *Against intellectual monopoly*. 2008: Cambridge University Press.
- Williams, S., *Free as in Freedom: Richard Stallman's crusade for free software*. 2002, Sebastopol, CA: O'Reilly & Associates.
- Giles, J., *Internet encyclopedias go head to head*. Nature, 2005. **438**(7070): p. 900-901.
- Spradling, C., L.-K. Soh, and C.J. Ansoorge, *A comprehensive survey on the status of social and professional issues in United States undergraduate computer science programs and recommendations*. Computer science education, 2009. **19**(3): p. 137-153.
- Buckley, M., J. Nordlinger, and D. Subramanian, *Socially relevant computing*, in *Proceedings of the 39th SIGCSE technical symposium on computer science education*. 2008, ACM: Portland, OR, USA. p. 347-351.
- Huff, C. and C.D. Martin, *Computing consequences: a framework for teaching ethical computing*. Commun. ACM, 1995. **38**(12): p. 75-84.
- Martin, C.D., et al., *Implementing a tenth strand in the CS curriculum*. Commun. ACM, 1996. **39**(12): p. 75-84.
- Granger, M.J., et al., *Using information technology to integrate social and ethical issues into the computer science and information systems curriculum. Report of the ITiCSE '97 working group on social and ethical issue in computing curricula in ITiCSE'97*. 1997, ACM SIGCSE: Uppsala, Sweden. p. 38-50.
- Goldweber, M., et al., *Historical perspectives on the computing curriculum (report of the ITiCSE '97 working group on historical perspectives in computing education)*, in *The supplemental proceedings of the conference on Integrating technology into computer science education: working group reports and supplemental proceedings*. 1997, ACM: Uppsala, Sweden. p. 94-111.
- Little, J.C., et al., *Integrating cultural issues into the computer and information technology curriculum (ITiCSE 2000 working group reports)*. SIGCSE Bull., 2001. **33**(2): p. 136-154.
- IP3. *International Professional Practice Partnership*. 2010 [cited 2010 26 June 2010]; Available from: <http://www.ipthree.org/>.
- IP3. *Strategic outcomes and directions*. 2010 [cited 2010 26 June 2010]; Available from: <http://www.ipthree.org/about-ip3/strategicobjectives>.
- Seoul Accord. *Seoul Accord*. 2010 June 4 2010]; Available from: http://www.abeek.or.kr/accord/contents.jsp?menu_l=85.
- Seoul Accord. *Seoul Accord Foundation Documents: Section D (Graduate Attributes)*. 2010 June 4 2010]; Available from: http://www.abeek.or.kr/accord/contents.jsp?menu_l=144&menu_m=199.
- Seoul Accord. *Seoul Accord Signatories*. 2010 June 4 2010]; Available from: http://www.abeek.or.kr/accord/contents.jsp?menu_l=154&menu_m=185.
- Gregor, S., et al., *The ICT profession and the ICT Body of Knowledge (Vers. 5.0)*, Australian Computer Society. 2008: Sydney, Australia.
- ACM/IEEE Joint Task Force, *Computing Curricula 2005: The overview report covering undergraduate degree programs in Computer Engineering, Computer*

- Science, Information Systems, Information Technology, Software Engineering. 2005.
28. ACM/IEEE Joint Task Force, *Computing Curricula 2001: Computer Science*. 2001.
 29. Cassel, L., et al., *Computer science curriculum 2008: An interim revision of CS 2001. Report from the interim review task force*. 2008.
 30. Lunt, B.M., et al., *Information Technology 2008: Curriculum guidelines for undergraduate degree programs in informaiton technology*. 2008, Association for Computing Machinery and IEEE Computer Society.
 31. Topi, H., et al., *Curriculum guidelines for undergraduate degree programs in informaiton systems*. 2010.
 32. Le Blanc, R., et al., *Software Engineering 2004: Curriculum guidelines for undergraduae degree programs in software engineering*, J.L. Diaz-Herrera and T.B. Hilburn, Editors. 2004.
 33. ACM/IEEE Joint Task Force, *Computer Engineering 2004: Curriculum guidelines for undergraduate degree programs in computer engineering*. 2004.
 34. Homkes, R. and R.A. Strikwerda, *Meeting the ABET program outcome for issues and responsibilities: an evaluation of CS, IS, and IT programs*, in *Proceedings of the 10th ACM conference on SIG-information technology education*. 2009, ACM: Fairfax, Virginia, USA. p. 133-137.
 35. Barroso, P. and G. Melara, *Teaching of computer ethics at the State of California's Universities and other countries*. The Ethicomp E-Journal, 2004.
 36. Martin, C.D., *The case for integrating ethical and social impact into the computer science curriculum*, in *The supplemental proceedings of the conference on Integrating technology into computer science education: working group reports and supplemental proceedings*. 1997, ACM: Uppsala, Sweden. p. 114-120.
 37. Spradling, C., L.-K. Soh, and C. Ansorge, *Ethics training and decision-making: do computer science programs need help?*, in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 2008, ACM: Portland, OR, USA. p. 153-157.
 38. Eisenberg, T., et al., *The Cornell commission: on Morris and the worm*. Commun. ACM, 1989. **32**(6): p. 706-709.
 39. *Morris worm - Wikipedia, the free encyclopedia*. [cited 2010, 2 August]; Available from: http://en.wikipedia.org/wiki/Morris_worm.
 40. H-FOSS. *The Humanitarian FOSS Project - Home*. 2010 [cited 2010 30 June 2010]; Available from: <http://www.hfoss.org/>.
 41. Gotterbarn, D. and T. Clear, *Using SoDIS™ as a risk analysis process: a teaching perspective*, in *Proceedings of the sixth conference on Australasian computing education - Volume 30*. 2004, Australian Computer Society, Inc.: Dunedin, New Zealand. p. 83-90.
 42. Vascellaro, J.E. *Google to build great wall in wake of cyber attack*. The Wall Street Journal 13 January 2010 [cited 2010 1 August]; Available from: <http://www.theaustralian.com.au/business/news/google-to-build-great-wall-in-wake-of-cyber-attack/story-e6frg90x-1225818914737>.
 43. Lewis, L. *Military's killer robots must learn warrior code*. The Sunday Times 16 February 2009 [cited 2010; Available from: http://technology.timesonline.co.uk/tol/news/tech_and_web/article5741334.ece.
 44. *Facebook's security flaws exposed*. The Australian IT [cited 2010 1 August]; Available from: <http://www.theaustralian.com.au/australian-it/facebooks-security-flaws-exposed/story-e6frgax-1225898699575>.
 45. Yongxi, C., *Crisis of personal data protection during the construction of e-government: reaction and limits of Hong Kong privacy laws*, in *Proceedings of the 1st international conference on Theory and practice of electronic governance*. 2007, ACM: Macao, China. p. 155-164.
 46. Good, N., et al., *Stopping spyware at the gate: a user study of privacy, notice and spyware*, in *Proceedings of the 2005 symposium on Usable privacy and security*. 2005, ACM: Pittsburgh, Pennsylvania. p. 43-52.
 47. Davtyan, S., et al., *Taking total control of voting systems: firmware manipulations on an optical scan voting terminal*, in *Proceedings of the 2009 ACM symposium on Applied Computing*. 2009, ACM: Honolulu, Hawaii. p. 2049-2053.
 48. SIGCAS. *SIGCAS Mission - SIGCAS - Computers & Society*. 2010 [cited 2010 30 June 2010]; Available from: <http://www.sigcas.org/>.
 49. Ellis, H.J.C., et al. *Can humanitarian open-source software development draw new students to CS?* . in *38th SIGCSE Technical Symposium on Computer Science Education* 2007. Covington, Kentucky, USA.
 50. Morelli, R.A., et al., *Revitalizing Computing Education by Building Free and Open Source Software for Humanity*. CACM, 2009. **52**(8): p. 67-75.
 51. SRC. *Socially Relevant Computing*. 2010 [cited 2010 30 June 2010]; Available from: <http://src.cse.buffalo.edu/index.html>.
 52. CPSR. *CPSR - Computer Professionals for Social Responsibility*. 2010 [cited 2010, 30 June]; Available from: <http://cpsr.org/>.
 53. CPSR. *CPSR - CPSR History*. 2010 [cited 2010, 2 August]; Available from: <http://cpsr.org/about/history/>.
 54. Moderated by Neumann , P.G. *The Risks Digest*. [cited 2010, 2 August]; Available from: <http://catless.ncl.ac.uk/risks>.
 55. *IEEE Technology and Society Magazine*. 2010 [cited 2010, 2 August]; Available from: http://ieeessit.org/technology_and_society/.
 56. Taylor, B. and S. Azadegan, *Moving beyond security tracks: integrating security in cs0 and cs1*, in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 2008, ACM: Portland, OR, USA. p. 320-324.
 57. Bailey, T. and J. Forbes, *Computers and Society in CS0: An interactive approach*, in *ASEE/IEEE Frontiers in Education Conference*. 2004: Savannah, GA. p. S2F 19-23.

58. Riser, R. and D. Gotterbarn, *Ethics activities in computer science courses*. SIGCAS Comput. Soc., 1996. **26**(3): p. 13-18.
59. von Kinsky, B.R., J. Ivins, and S.J. Gribble, *Engaging undergraduates in discussions about ethics in computing*, in *Proceedings of the ninth Australasian conference on Computing education - Volume 66*. 2007, Australian Computer Society, Inc.: Ballarat, Victoria, Australia. p. 163-169.
60. Clear, T., et al., *Resources for instructors of capstone courses in computing*. SIGCSE Bull., 2001. **33**(4): p. 93-113.

Appendix A. Social and Professional Issues Elaboration

There are eleven components of social and professional issues that are recognized by the ACM/IEEE Body of Knowledge. This list should not be considered comprehensive or static, but is representative of what is commonly encountered in education and in practice. This section contains an elaboration of what these components mean. They are identified as either core or elective, and include the CS-2008 recommended minimum number of class-time hours if they are core. Not included in these hours is time for homework and other outside activities.

1. *History of Computing* is a core component with a minimum class time of 1 hour. Learning about the pioneers, current leaders, and evolution of computer technologies provides students with a context for understanding the increasing impacts and roles of computation today and in the future. Included are the historical roots and confluence of theory, industrial automation, and communications in the development of computers; the driving forces from shipping, war, and economic competition; and continuing trends of rapid evolution with implications of various sectors being out of synchronization.
2. *Social Context* is a core component with a minimum class time of 3 hours. It deals with the use and misuse of computers in society, and effects on communication and culture. The focus is on the effect of computing on how people interact between themselves, governments, business, and other social institutions. It also considers the other direction, namely the effect of culture and human institutions on the acceptance and uses of computing. Included are the effects on gender and cultural disparities such as the gender gap in computing education, understanding and acceptance of computing permeating all aspects of life as in electronic voting and robotic servants, and the effect on governments in monitoring and controlling their populations and use of computers in cyberwarfare.
3. *Analytical Tools* is a core component with a minimum class time of 2 hours. Rather than pertaining directly to social impacts of computing, this component addresses the ability to reason about these issues, to form arguments and analyze them for fallacies, to consider the motivations of stakeholders, and evaluate ethical tradeoffs. It is fundamentally the logical analysis of ethical issues.
4. *Professional Ethics* is a core component with a minimum class time of 3 hours. As one of the most accepted and taught components, this introduces students to professional codes of ethics and practice, and to other aspects of the role of a professional in industry and academia. Considerations of whistle-blowing, ergonomics, discrimination, ethical concerns of computing system development are examples of topics. Also included are understanding what the law says about ethical issues, the ability to analyze the intent, uses of, effectiveness, and “rightness” of the codes and laws, and ways of dealing with ethical violations.
5. *Risks* is a core component with a minimum class time of 2 hours. Awareness of historical system failures, limitations of testing, approaches to managing risks, and the role of risk management are topics that are covered by this component. An important skill is the ability to analyze costs and benefits of differing levels of risk acceptance.
6. *Security Operations* is an elective component. It deals with physical security including physical and personnel access, recovery, etc. Finding weak points, designing security measures, and intrusion detection are some of the topics that can be taught.
7. *Intellectual Property* is a core component with a minimum class time of 3 hours. Rights to “intellectual property,” i.e. copyright, patent, trademarks, including legal aspects and international considerations, licensing of software, problems of license violations such that unauthorized copies or inclusion of copyleft software in proprietary products including impact on software development, and the historical context are aspects of this component. Additional aspects include understanding patents, trade secrets, and the open source software movement with its impacts and implications for continued software and hardware development. Students should also understand the various licensing options and their implications.
8. *Privacy and Civil Liberties* is a core component with a minimum class time of 2 hours. Understanding the legal bases, recognizing where the threats may arise, dealing with tradeoffs between access to information and possible loss of privacy, and technological ways of dealing with the problems are topics that may arise in this component. Additional topics are the extension of civil rights to cyberspace including international differences and impacts, the effects of technology on the ability to regulate such rights, and likely future trends.
9. *Computer Crime* is an elective component. In addition to the expected topics of hacking, viruses and their ilk, identity theft, and techniques for their detection, prevention, and recovery, all of which non-computing professionals should be cognizant of, it is important that computing professionals consider their roles and responsibilities when developing systems. Also included are the relative roles of government, industry, and the individual in managing the risk and assuming responsibility for losses.
10. *Economics of Computing* is an elective component. Outsourcing, skilled labor supply, access to computing facilities and the Internet including geographical and cultural differences, software pricing strategies, monopolies, and the concept of green computing would all fall under this category.
11. *Philosophical Frameworks* is an elective component. There are various ethical theories that philosophers deal with, including utilitarianism and deontological theories. They identify different ways that people can measure “good” and “bad”. Understanding the differences in scientific and philosophical approaches would be part of this component.

Appendix B – Implementation Cases

Implementation cases demonstrating how societal impact issues have been addressed at various institutions are presented below. This includes a liberal arts school (Grinnell College: integrated modules and embedded), a private university (John Carroll University and Carnegie Mellon: experiential), large public universities (Curtin University and Towson University: standalone courses; The University of Melbourne: integrated modules) and finally a program (graduate) whose whole curriculum is formed around the coverage of social issues (University of Bologna: embedded). Each description lists a set of *contributors*. Interested readers are invited to contact a contributor with questions.

B1. Towson University

Contributor: Joyce Currie Little

Location: Baltimore, Maryland, USA

Institution type: Metropolitan

Program being documented: Bachelor's Degree

Curricular model: ACM Computer Science (CS)

Accreditation body: Accreditation Board for Engineering and Technology (ABET), using Computing Accreditation Commission (CAC), lead society Computing Sciences Accreditation Board (CSAB), for the Computer Science program; IS and IT programs not yet accredited but applications for both are being planned, to the same accreditation body. Institutional program review is done every five years.

At what level is the impact of CS/IT on society addressed? Undergraduate bachelor's degree in Computer Science (CS), Information Systems (IS), and Information Technology (IT).

Is the content spread across many courses or concentrated in one or two? There are two courses in the undergraduate CS major: COSC418 Societal and Ethical Concerns for Computer Scientists (three credits); and COSC480 Senior Seminar (one credit). To complement those, some required computing courses include modules of topics that provide social issues and impact using examples and cases appropriate to the course. This integration of topics provides additional enhancement that connects the content in more depth, and to the appropriate subject matter.

Are the courses compulsory or elective? All computing majors (CS, IS, and IT) must take either the three-credit course or the one-credit course. The course is available also for students in other majors, for whom their home department recognizes it as a substitute for their own course in ethical and social issues.

How is the learning of social impacts facilitated in the course(s)? In the three-credit course, students participate in a team research project to study one area of impact of computing on society, such as GPS, mobile phone, medical databases, for example. They prepare a professional paper and give a team presentation. Assignments during the semester include study followed by online and in-class discussion groups, using topics such as the USA PATRIOT Act, for example. Short written assignments are required, with citations and analysis. Students give presentations of historic cases (such as the Therac-25 Radiation Machine Case, for example). Two exams are given.

The one-credit course is taught in a hybrid manner, meeting both in-classroom and online. This course is moving to become an online course. It emphasizes the use of scenarios and cases and requires students to perform case analysis, resulting in the writing of summaries of actions taken, or actions that should be taken, by stakeholders.

What content is prescribed; in particular, what learning outcomes are desired? Topics of study in the course include: introduction to evolution of computing devices and machines, intellectual property, electronic theft and piracy, electronic privacy, malicious software such as viruses and worms, evaluation of risks and probabilities of error and failure, professional responsibility, professional concerns of computing and information technology workers, codes of ethics of behavior, standards for good practice, computer forensics and security, qualifications of workers, and certification and licensing of personnel.

The goals in this course are designed to help the student be able to: describe some of the ethical and societal problems that have arisen due to computer usage in several different disciplines, including privacy, equity of access, and intellectual property; describe some of the societal problems and the risks that may arise from the design, development, and implementation of software; describe and interpret some of the major ethical theories for classifying cases and events; explain some of the effects of computerization on the human condition in society; identify some major issues related to intellectual property rights involving hardware and software; identify some major concerns related to the individual right to privacy; analyze a workplace or societal situation to establish its ethical boundaries and impact; explain the purpose of, and give examples of the use of, professional codes of ethics in the workplace; explain some of the non-traditional ways computer professionals prepare for the workplace by means of certification examinations and trade/vocational schools; express some arguments for, and against, professionalism of computer/computing professionals; explain how to protect against, and describe some ways to prevent, some of the types of malicious behavior that interferes with responsible computer usage; describe some of the ethical and societal problems that have arisen due to computer usage in several different disciplines, including privacy, equity of access, and intellectual property; describe some of the societal problems and the risks that may arise from the design, development, and implementation of software; describe and interpret some of the major ethical theories for classifying cases and events; explain some of the effects of computerization on the human condition in society; identify some major issues related to intellectual property rights involving hardware and software; identify some major concerns related to the individual right to privacy; analyze a workplace or societal situation to establish its ethical boundaries and impact; explain the purpose of, and give examples of the use of, professional codes of ethics in the workplace; explain some of the non-traditional ways computer professionals prepare for the workplace by means of certification examinations and trade/vocational schools; express some arguments for, and against, professionalism of computer/computing professionals; explain how to protect against, and describe some ways to prevent, some of the types of malicious behavior that interferes with responsible computer usage.

B2. Curtin University

Contributors: Raj Gopalan, Tele Tan, and Brian R. von Konsky

Condensed by: Mikey Goldweber

Location: Perth, Western Australia

Institution type: Metropolitan

Program being documented: Bachelor's Degree

Curricular model: ACM Computer Science (CS)

Accreditation body: Australian Computer Society

At what level is the impact of CS/IT on society addressed?

Professionalism, societal impact, and ethics are addressed in all undergraduate programs: the three-year Bachelor of Science degree in Computer Science (CS), Information Technology (IT), and Software Engineering (SE). The Australian Computer Society has accredited all of these programs.

Is the content spread across many courses or concentrated in one or two? Social and ethical issues are concentrated in Project Design and Management (PDM351). Professionalism in ICT is concentrated in PDM351 and Software Engineering Project capstone units (SEP401/402 and 451/452).

Are the courses compulsory or elective? PDM 351 is compulsory for all CS majors, for IT, and SE majors. The Software Engineering Projects units are compulsory for all SE majors.

How is the learning of social impacts facilitated in the course(s)? PD351 has three guest lectures of one hour each done by the University Chaplain. Assessments include a student debate and an essay.

PDM 351 also introduces students to professionalism and the roles and responsibilities of IT professionals working on software teams. This is facilitated through additional lectures that introduce topics on the Personal Software Process (PSP), risk management, and earned value. Assessments associated with this portion of the unit include the design, planning, and implementation of a proof of concept project. This is undertaken as a team, and has traditionally been sponsored by the University's IT Services area.

In addition, Software Engineering students are assigned a leadership role as a member of a software development team on a large project of significant scope in the Software Engineering Project capstone units. These roles include Team Leader, Development Manager, Quality Manager, and Support Manager. Collectively, students manage project risks, quality, scope, time and schedule, each in the context of defined leadership goals and responsibilities (von Konsky & Ivins, 2008). Outcomes from project management roles are presented in an end-of-semester poster presentation.

What content is prescribed; in particular, what learning outcomes are desired? Learning Outcomes for PDM351 include the ability to appraise relevant professional and ethical issues as they relate to the computing and software industries. Professionalism outcomes from PDM351 include an ability to demonstrate excellent time management skills and understand the application of the personal software process for time management and effort estimation purposes.

Professionalism outcomes from Software Engineering Project capstone units include the ability to manage a project team on a software project of significant scope, with the ability to set and

achieve defined goals and the means to measure their attainment; and to be a cooperative and contributing team member, planning and tracking personal work that leads to the delivery of a quality product. The capstone project units are team-based projects of significant scope, undertaken in conjunction with an industry partner, and therefore offer great work placement experience.

Reference:

Curtin University. (2010). Teaching and Learning – Curtin Graduate Attributes. Retrieved 18 June 2010 from https://otl.curtin.edu.au/teaching_learning/attributes.cfm.

von Konsky, B.R., and Ivins, J. (2008) Assessing the Capability and Maturity of Capstone Software Engineering Projects, 10th Australasian Computing Education Conference (ACE 2008), in conjunction with the Australasian Computer Science Week (ACSW 2008), Wollongong, 22-25 January, 2008, Conferences in Research and Practice in Information Technology, Vol. 78, Simon and Margaret Hamilton, Eds.,

B3. Grinnell College

Contributors: Janet Davis, Henry Walker

Condensed by: Joyce Currie Little

Location: Grinnell, Iowa, USA

Institution type: Liberal Arts College

Program being documented: Bachelor of Arts

Curricular model: ACM/IEEE Computer Science (CS) Curricula 2001, 2008, & Liberal Arts Computer Science Consortium (Bruce et.al. 2010, Liberal Arts Consort. 2007, Walker and Keleman 2010)

CS/IT Accreditation Body: None

At what level is the impact of CS/IT on society addressed?

Undergraduate, both CS majors and non-majors

Is the content spread across many courses or concentrated in one or two? The content is spread over a number of courses.

For non-majors, the course CSC105 The Digital Age provides both technical content and social issues. All incoming students take a First Year Tutorial; when offered by CS faculty, many sections include social and ethical issues of computing. Computer Science majors may take courses that include aspects of societal issues of computing: CSC232 Human-Computer Interaction; CSC 261 Artificial Intelligence; CSC323 Software Design, CSC325 Databases and Web Application Design; CSC341 Theory of Computation, and CSC364 Computer Networks. Special Topics courses often are offered, such as one on Women in Computing, and another in Socio-Technical Issues in Computer Networks.

Are the courses compulsory or elective? The Computer Science major requires two of the courses: CSC341, CSC323 or CSC325.

How is the learning of social impacts facilitated in the course(s)? For non-majors, some of the content in CSC105 The Digital Age offered for non-majors was influenced by the "information fluency" movement (Comm. Information Tech. 1999). Some of the sections of the course use articles from the literature; others may introduce new technical topics and discuss their impact; some faculty have developed materials for social

and ethical issues work (Walker 2005). Majors are exposed to social issues as an on-going theme through numerous courses, including CSC 341, CSC 323, and CSC 325. For example, lectures include social impact as part of the discussion of applications, readings cover social and ethical issues, discussions of design and implementation include consequences of decisions, and theoretical limitations are connected with potential policy considerations and algorithmic problem solving.

What content is prescribed: in particular, what learning outcomes are desired? Social issues are presented in a variety of ways, with a liberal arts approach for all students in computing courses, including the Computer Science majors. This approach often emphasizes assigned readings, interpretations, discussions, specific writing assignments, and independent research. Topics of particular relevance include: accessibility for people with disabilities, persuasive technology, privacy, security, and intellectual property. Learning outcomes include these: understand roles of computers in today's society; have ability to apply relevant principles in the solving of some common problems; collaborate on team-based projects; understand the importance of attribution when using the works and ideas of others; understand that programs don't exist in isolation but have an impact on society; analyze social and ethical implications of both existing and potential technologies.

For a more extensive treatment of computing within the framework of a liberal arts institution, see (Walker & Kelemen 2010).

References:

Bruce, Kim B. Cupper, Robert D., Drysdale, Robert L. Scott. ACM Transactions on Computing Education (TOCE) , A History of the Liberal Arts Curriculum and its Model Curricula, Volume 10, Issue 1, Article 3, March 2010.

Liberal Arts Computer Science Consortium, A 2007 Model Curriculum for a Liberal Arts Degree in Computer Science, JERIC, Volume 7, Issue 2, Article 2, June 2007.

Committee on Information Technology Literacy, National Research Council, National Academies Press.. Being fluent in Information Technology 1999.

Walker, Henry M., The Tao of Computing, Jones & Bartlett, 2005.

Walker, Henry M. and Charles Kelemen, "Computer Science and the Liberal Arts: A Philosophical Examination", ACM Transactions on Computing Education (TOCE), volume 10, Issue 1, Article 2, March 2010.

B4. John Carroll

Contributor: Linda Seiter

Condensed by: Mikey Goldweber

Location: Cleveland, Ohio, USA

Institution type: Comprehensive University

Highest degree offered in CS/IT: Bachelor

Curricular model employed/followed: ACM models consulted

CS/IT Accreditation body: none

At what level(s) is the impact of CS/IT on society addressed? Bachelor's

Is the content spread across many courses or concentrated in one or two? The program is supported by three faculty members. One faculty member incorporates social issues content in all her courses. The other two faculty members limit their coverage to their software engineering courses.

Are the courses compulsory or elective? Yes. All students must take two software engineering courses in addition to a Web Design course. The Web Design course is always taught by the faculty member who incorporates social issues in her courses.

How is the learning of social impacts facilitated in the course(s)? Each course with a social impacts component requires interconnected components to facilitate content/learning.

- Required readings. This can include books as well as articles.

- Experiential learning. In the introductory CS courses, students are initially taught Alice/Scratch. Students, then, working in pairs, go off into the community and teach disadvantaged students Alice/Scratch computer programming. The target audiences are 5-9 grade students working in local community centers.

For the Web Design and Software Engineering courses, student groups work on projects for local community non-profit organizations/partners.

- Journals. All students are required to keep a journal of their experiences when they either act as programming instructors or as a "consultant" for a community partner (Web Design and/or Software Engineering)

What content is prescribed; in particular, what learning outcomes are desired?

Introductory CS course(s): The primary content is a focus on our digital society and the digital divide. In particular there is a focus on the impediments some communities face entering technology fields, social inequities that technology creates, maintains or exasperates, and the concept of social justice. Not only does teaching students programming help them learn programming, but it also allows students to focus on goes into programming/technology fields and who is discouraged from entering these fields.

Web Design/Software Engineering: While one key learning outcome is the value of technology to the operation of non-profits, the primary learning outcome is the importance of the open source movement to their success. The hope is that even though most students will have the opportunity to earn substantial incomes, post graduation, that they should consider continuing to volunteer some time to the open source movement.

B5. The University of Melbourne

Contributors: Brian von Konsky, Alistair Moffat

Condensed by: Mikey Goldweber

Location: Melbourne, Australia

Institution type: University

Highest degree offered in CS/IT: Ph.D.

Curricular model employed/followed: Melbourne model

CS/IT Accreditation body: Engineers Australia and the

At what level(s) is the impact of CS/IT on society addressed?

This implementation case describes how professionalism, the societal impact of software-based systems, and ethical issues are addressed in the Master of Engineering (Software) degree at the University of Melbourne (University of Melbourne, 2010b).

This degree was introduced in 2008 and has received provisional accreditation at the professional level from Engineers Australia. This is the highest level of accreditation available to a new program. Following completion of the first sizable cohort, Engineers Australia will be invited to review the program once again and confirm full professional accreditation. The degrees are also accredited by the Australian Computer Society.

The program is based on the “Melbourne Model”, which incorporates a three-year generalist undergraduate degree, potentially followed by a two-year postgraduate specialization. The undergraduate degree structure includes a major to provide depth from a structured sequence of subjects, plus breadth subjects from complementary disciplines. Students completing the “new generation” Bachelor of Science with a major in Software Systems are eligible for direct entry into the second year of the three year Masters of Engineering (Software) (University of Melbourne, 2010b). Students from other undergraduate majors may also be eligible to enter the ME(Software), and will be granted between zero and one and a half years of credit. An alternative two-year specialization is provided by the MSc(Computer Science), a research-training pathway.

Prior to 2010, students wanting an engineering qualification in software engineering undertook the four-year Bachelor of Engineering (Software) degree. As of 2011, the BE will be fully replaced by the combination of the three-year Bachelor of Science degree with a major in Software Systems followed by the last two years of the Master of Engineer (Software) program.

Is the content spread across many courses or concentrated in one or two? Students entering the Masters of Engineering (Software) via the Bachelor of Science (Software Systems) pathway have societal impact and ethics concentrated in two undergraduate subjects entitled Engineering Systems Design 1 and 2. Students from non-major backgrounds entering the Master of Engineering (Software) see equivalent material in a subject entitled Engineering Communication.

Professionalism in ICT is a key focus of two year-long projects that occupy a quarter of both the second and third year of the Masters of Engineering (Software).

Are the courses compulsory or elective? These subjects are compulsory.

How was the “what issues to address” question(s) decided? Issues to be addressed are determined at the discretion of the lecturer.

How is the learning of social impacts facilitated in the course(s)? Learning is facilitated through a mix of design projects, interactive workshops and lectures. Wherever possible, guest lectures from industry practitioners are used – for example, by experts who have given evidence in court cases.

Describe any special efforts to attract/involve/retain “techie” students? Generalist undergraduate degrees with breadth

subjects that complement a major are intended to provide students with broad educational choices. The goal is to provide a solid foundation for subsequent employment or further study leading to a professional qualification in a chosen discipline. Effectively, the Melbourne Model caters to two different kinds of students: those that want technical depth in a specialized area via a five year “3+2” degree sequence, and those that want a broad interdisciplinary learning experience. Additionally, subjects are designed to be engaging in order to retain students throughout the program. Students in four of the five other Melbourne Model undergraduate degrees are able to take IT subjects as their “breadth” study.

How do you persuade colleagues and committees that societal impacts should be addressed in “regular computing courses”? Engineers Australia accredits engineering programs in Australia. To qualify for accreditation, an institution must demonstrate that the program produces graduates that have attained the stage 1 competencies of a professional engineer. In particular, these competencies include an:

- ... *Understanding of social, cultural, global, and environmental responsibilities and the need to employ principles of sustainable development*
- a. *Appreciation of the interactions between technical systems and the social, cultural, environmental, economic and political context in which they operate, and the relationships between these factors*
- b. *Appreciation of the imperatives of safety and of sustainability, and approaches to developing and maintaining safe and sustainable systems*
- c. *Ability to interact with people in other disciplines and professions to broaden knowledge, achieve multidisciplinary outcomes, and ensure that the engineering contribution is properly integrated into the total project*
- d. *Appreciation of the nature of risk, both of a technical kind and in relation to clients, users, the community and the environment*

(Engineers Australia, 2010)

The Stage 1 competencies further include an expectation that professional engineers demonstrate an:

- ... *Understanding of professional and ethical responsibilities, and commitment to them*
- a. *Familiarity with Engineers Australia’s Code of Ethics, and any other compatible codes of ethics relevant to the engineering discipline and field of practice, and commitment to their tenets*
- b. *Awareness of legislation and statutory requirements relevant to the discipline and field of practice*
- c. *Awareness of standards and codes of practice relevant to the discipline and field of practice*

(Engineers Australia, 2010)

The University of Melbourne expects its programs to produce graduates that work sustainably in their discipline, and in a culturally aware, socially responsible, and ethical manner (University of Melbourne, 2010a).

Moreover, Department of Computer Science and Software Engineering staff members are known to have a shared sense of responsibility and to be advocates for the safeguarding of digital privacy and the security of on-line data.

What content is prescribed; in particular, what learning outcomes are desired? Learning outcomes include:

- Understanding of social, cultural, global and environmental responsibilities and the need to employ principles of sustainable development
- Explain the importance of engineers and engineering in society;
- Discuss the differences between the key engineering disciplines;
- Explain the importance and principles of sustainable development and safety;
- Identify problems and formulation solution strategies.

Describe the experiential component(s) if any. Professionalism in Software Engineering is a key aspect of two year long projects in the MEng(Software).

References

Engineers Australia. (2010). Appendix B - Guide to assessment of eligibility for membership (Stage 1 competency). Retrieved 18 June 2010, 2010, from

http://www.engineersaustralia.org.au/shadomx/apps/fms/fmsdownload.cfm?file_uid=0446A3D5-B812-B0F4-4B66-8AF85D4C337B&siteName=ieaust

University of Melbourne. (2010a). Graduate attributes - The University of Melbourne. Retrieved 18 June 2010, 2010, from <http://www.unimelb.edu.au/about/attributes.html>

University of Melbourne. (2010b). Master of Engineering (Software): The University of Melbourne. Retrieved 18 June 2010, 2010, from http://www.cs.mu.oz.au/future/me_software.html

B6. Carnegie Mellon

Contributor: Joe Mertz

Condensed by: Mikey Goldweber

Location: Pittsburgh, Pennsylvania, USA

Institution type: University

Highest degree offered in CS/IT: Ph.D.

Curricular model employed/followed: ACM models consulted
CS/IT Accreditation body: none

At what level(s) is the impact of CS/IT on society addressed?

At all levels. At the bachelor's and master's level there are courses students can take. At the doctorate level there are a few faculty and research groups that are doing research specifically targeted toward impacting society with CS/IT.

Is the content spread across many courses or concentrated in one or two? There is no single concentration; rather, there are courses that address the impact of CS/IT on society. Some include:

Technology Consulting in the Community (TCinC)

Technology and Global Development (TGD)

Human-Computer Interaction in the Developing World

Are the courses compulsory or elective? They are not compulsory. They count as computer science electives.

How is the learning of social impacts facilitated in the course(s)? In TCinC, each student is assigned to work one-on-one in a local nonprofit organization, school, or municipal agency. They travel to the site each week for 3 hours to work with staff. In class sessions, the instructors teach a capacity-building model of IT consulting, which the students play out each week with their client. Most of the learning of the social impact comes from this experience. The course teaches that all IT systems live within social systems. We use an excellent documentary, *The Waters of Ayole* to get this idea across well. We teach a consulting model that has students broadly investigate the organization, and they consider the impacts of alternative solutions to the problems they find when planning their scope of work. They are required to document the outcomes of their work in terms of these impacts, and at the end of the semester we hold a reflective session in which we discuss more general societal impacts.

In TGD, students look at examples of what has and has not worked when employing IT for development.

What content is prescribed; in particular, what learning outcomes are desired?

TCinC teaches students how to:

- Establish a professional working relationship
- Quickly assess a complex technical environment and identify problem areas
- Systematically bring structure to unstructured problems
- Communicate technical ideas to a non-technical audience
- Negotiate with a client acceptable deliverables for the consulting period
- Develop and execute a work plan
- Use writing skills to maintain working documents that describe, plan, persuade, and coordinate work with others
- Reflect and learn from their experience as well as the experience of their colleagues
- Broaden their understanding of the relevance of computer science and information systems

Various instances of TGD have had the general learning outcomes of:

- Build affective interest in the topic of technology for developing communities.

Students will be encouraged to break away from us/them dichotomies to see the process of introducing technology into developing communities as a partnership dynamic, not a one-way street.

- Be aware of the problems in developing communities.

Students will be aware of the breadth of problems, can critically define why a condition is a problem, are aware of baseline measures, and can argue how a development intervention would impact that baseline state.

Students understand strategies for building partnerships in developing communities in order to develop a richer understanding of the problems and together work toward locally-appropriate, sustainable solutions.

- Be literate in the ways that development has changed over the past several decades

Students are aware of the historical path development has taken, they can make comparative statements of a development program in terms of past development programs, they understand best practice principles and can critically evaluate a development program.

- Be literate in the critical (both positive and negative) literature on technology based development

Students understand that technology by itself cannot solve social problems such as poverty, inequality, hunger or disease; that the introduction of new technologies often require simultaneous adjustments in social, cultural, political, economic and legal realms; and that technology is not neutral and is imbued with cultural meaning.

- Be able to perform economic analysis of development projects

Students can itemize the types of economic benefits a development program can provide and can do simple economic analysis of the costs and benefits of a development program.

- Be aware of technology trends

Students are aware of technology trends and can use past and current innovations in order to propose new applications to development problems (E.g. in fields such as robotics, artificial intelligence, human-computer interaction and user interface design, networking, natural language processing, sensing, medical diagnostics, biotechnology, water filtration technologies, and energy production.)

- Be able to design, propose, and implement a pilot development project

Students can synthesize their understanding of problems, development best practices, economic analysis, and technology trends in order to design, propose and implement a pilot development project.

Students observe the challenges of development-project deployment.

Students understand the pitfalls of trying to do development without strong partnerships, and have an idea of how to go about building working partnerships with developing communities.

- Develop cross-cultural competency

Students will learn about attitudes to technology in developing communities and why they may affect success or failure of projects

Students will understand that not all developing communities are the same--some are eager to gain access to new technologies, some are not.

Students will be able to critically analyze dominant assumptions about, and approaches to, the role of technology in improving lives of people in developing communities.

Students will be aware of the political dimensions of development, especially by organizations like USAID and Gates Foundation.

- Understand related meta-issues

Students are aware of the issues of intellectual property, privacy, and open source vs. proprietary software and can discuss their relevance to development projects.

Students understand issues of intellectual property: what are advantages/disadvantages of using proprietary technology (e.g., Monsanto seeds or Microsoft windows vs. "open source" technology).

B7. University of Bologna

Contributor: Renzo Davoli

Condensed by: Mikey Goldweber

Location: Bologna, Italy

Institution type: University

Highest degree offered in CS/IT: Ph.D.

Curricular model employed/followed: Bologna agreement, with additional inspiration from the ACM/CS models

CS/IT Accreditation body: none

At what level(s) is the impact of CS/IT on society addressed?

Bachelors and Masters. The Department offers a professional Master in Science and Technology of Free Software.

Is the content spread across many courses or concentrated in one or two? There are no specific courses on social impact, but the topic is included to some extent in all of the traditional technical courses.

Are the courses compulsory or elective? The social impact of computing is discussed in many of the compulsory courses. (e.g. operating systems, software engineering, database systems)

Additionally, in both the bachelor's program and the Master's of Free Software, there are compulsory courses on copyright issues. There are also extra curricular short courses on copyright(copyleft), free software, netiquette, and the department's "Computer Use" rules. This last course is compulsory whose final web-test is a prerequisite for access to any of the departmental computing facilities.

How is the learning of social impacts facilitated in the course(s)? The Department has a strong commitment to the use of free software in education. All of the department's computing facilities are based on free software – a "de facto" lecture itself regarding social issues.

The Free Software movement represents an important philosophical perspective in the debate on the ownership of information. As with public free-lending libraries, this movement advocates that all information should be freely accessible to all. Hence the University's computing programs are based around the philosophy, and possibly more important, the culture, that all the software that the students interact with (e.g. operating systems, compilers, networks) are part an open laboratory for students to learn in/from.

Furthermore, the Master's of Free Software focuses on the process of creating free software.

What content is prescribed; in particular, what learning outcomes are desired? The primary learning outcome is to create good (Net) citizens who will strive to preserve their intellectual freedoms and hopefully use their creative talents to increase the body of free knowledge. A secondary learning

outcome is for students to become proficient in understanding that information is power and how the control of information can lead to unethical behavior. This includes understanding that software is information and the unethical consequences when that software is controlled (i.e. proprietary software).