Incorporating Social Issues of Computing in a Small, Liberal Arts College: A Case Study

Janet Davis
Department of Computer Science
Grinnell College
Grinnell, IA 50112
+1 641-269-4306
davisjan@cs.grinnell.edu

Henry M. Walker
Department of Computer Science
Grinnell College
Grinnell, IA 50112
+1 641-269-4208
walker@cs.grinnell.edu

ABSTRACT
CC2001[1] and CC2008[5] recommend that an undergraduate computing curriculum include 16 hours related to social and professional issues. An ITiCSE 2010 Working Group discussed approaches for incorporating this material in the curriculum and outlined seven contrasting implementation cases [8]. Also, Baldwin *et al* discuss the implementation of computing curricula at five different liberal arts colleges [4]. However, none of these provides specific implementation details for addressing social issues in a liberal arts computing curriculum. This paper identifies successful strategies from one college and begins a general discussion of teaching social issues of computing in a liberal arts setting.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education, curriculum*

General Terms
Design, Economics, Human Factors, Theory.

Keywords
Liberal arts, social issues, ethics, teaching methods.

1. INTRODUCTION
National curricular guidelines consistently identify the need for undergraduate computing curricula to address social issues. For example, CC 2008 [5] quotes CC 2001 [1] in stating

> Students … 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? [5, p. 92.]

In particular, CC 2008 specifies 16 hours devoted to social and ethical issues [5, pp. 92-96].

Similarly, the first curricular goal in the 2007 Model Curriculum for a Liberal Arts Degree in Computer Science is "To enable understanding the capabilities, limitations, and ramifications (technical, ethical, and social)." The Model Curriculum continues, "In particular, students in the type of program described here should … Recognize the importance of social and ethical issues of computing" [9, sec. 3].

But even when faculty have strong motivation and fine intentions, challenges arise when incorporating social issues into already crowded curricula. An ITiCSE 2010 Working Group addressed such implementation issues; discussions highlighted that diverse educational settings may require different pedagogies. The subsequent report [8] provides seven distinct, high-level implementation cases that outline possible approaches for incorporating social issues within curricula. Due to space limitations, each case was less than one page—enough to compare approaches, but too short to provide extensive guidance for any one approach. These Working Group discussions also highlighted that assumptions, opportunities, and challenges for liberal arts colleges may differ from those encountered in other types of schools [8].

In parallel work, the Liberal Arts Computer Science Consortium compiled five case studies describing computing programs in liberal arts settings. While providing a fine review of approaches for technical materials, space considerations limited discussion of the inclusion of social issues [4].

This paper describes in some detail how social issues can be incorporated in a liberal arts computing curriculum, and in particular, considers ways to build on traditional strengths of the liberal arts environment. Although this paper focuses on a specific environment, presenting these practices can be useful to others in at least two ways:

- the liberal arts environment is an integral part of hundreds of colleges in the United States and elsewhere;
- many of the ideas discussed here may apply in other settings as well.

Of course, each department and school will need to consider what approaches work locally. We hope this discussion can promote brainstorming and provide pragmatic suggestions for many colleges and universities.

The remainder of this paper focuses on challenges and opportunities of the liberal arts setting, and down-to-earth strategies that experience has shown to work—at least in some settings. Section 2 describes the small, liberal arts college setting, and Section 3 considers how computing fits within this context. As these sections indicate, assumptions and emphases may be different for liberal arts programs than for more technically-oriented curricula; these differences both impose limitations and enable possible synergies.

SIGCSE'11, March 9–12, 2011, Dallas, Texas, USA.

Section 4 focuses on courses for non-majors, both to promote computer fluency among future citizens and as an opportunity to recruit additional majors. Section 5 provides a parallel discussion for potential majors. Although this paper reports on a successful program, opportunities for refinement and expanded assessment remain; likely next steps are described in Section 6. Section 7 concludes the paper with an invitation to continue the discussion.

2. THE SMALL, LIBERAL ARTS COLLEGE

A liberal arts college fosters breadth of education and interdisciplinary connections, and students typically earn a Bachelor of Arts degree. Students usually have a choice of major, but depth of study in one field is balanced by developing broader interests in the humanities, social studies, and sciences. Thus, in a typical liberal arts program, students devote about 40% of their course work to their major and related subjects (e.g., computer science and supporting mathematics) and the remainder in outside areas. In contrast, a typical Bachelor of Science or Bachelor of Engineering program requires that students devote 60% or more of their course work to the major [9, 13].

Many students are drawn to a liberal arts college because they have a variety of interests. Even a student who wants to pursue a science major, for example, may have serious interests in fine arts, foreign languages, social studies, or other areas. Thus, the vast majority of students in a liberal arts setting feel comfortable studying social issues. For example, Grinnell's *Catalog* identifies the following six "Elements of a Liberal Education" [2009, pp. 13-15:

1. Communication skills: the ability to read, analyze, write, and speak effectively
2. "Study of a language other than one's own"
3. Lab-based science (particularly the scientific method in the natural sciences)
4. Quantitative reasoning
5. The study of human behavior and society
6. "Study of creating expression" and work in the humanities

Within this framework, social and ethical issues are core topics for all students (element 5 in the list, with elements of 1, 2, and 6).

Often, the greater challenge is to encourage students to study computing (or science more generally). This orientation may be different from what is found in many technical programs, where students focus on computing or science and may not have as strong an affinity to social issues.

2.1 Breadth in a Liberal Arts Program

Typical of many liberal arts colleges, the Mission Statement of Grinnell College states that the college seeks to educate students "for the different professions and for the honorable discharge of the duties of life." Those professions include the fine arts, literature, philosophy, social studies, mathematics, physical and life sciences, and even computer science. After undergraduate work, those professions expand to include law, medicine, engineering, social service, politics, teaching, and many others.

Liberal arts institutions employ various strategies to help students achieve a broad program of study. One common approach involves a core curriculum or distribution requirements; students must take courses in different departments or subject ares to meet these requirements. At one extreme, a college may specify one-third or even one-half of an undergraduate degree program. If an introductory computing course satisfies a requirement, then students have an incentive to try it out, and instructors have the chance to excite these students about the discipline. However, this approach also has the disadvantage that students may lack motivation—they may take courses only to complete requirements, with little real interest or commitment.

Grinnell College represents the other extreme, in which advisers are charged to work with students to achieve breadth. Grinnell's only requirements for graduation are completion of a First-Year Tutorial (a writing-intensive course discussed further in Section 5.2), a major (8 primary courses plus a few supporting courses), and 124 credits (no more than 48 in a department, 92 in a division) with a GPA of 2.0 or higher. Under such a system, advisers bear the primary responsibility for enforcing a breadth of education for each student. Since specific courses are not required (outside the major), students usually are highly motivated for each course—after all, they could have chosen something else. But, without requirements, it can be a challenge to entice students to take courses in unfamiliar fields.

In addition to formal course work, many liberal arts colleges encourage independent projects and close student-teacher relationships. For example, at Grinnell, the Mentored Advanced Project (MAP) program provides an opportunity for students to conduct research or undertake other advanced, student-driven work. MAP students work alone or in small groups under the close supervision of a faculty member, typically in their major or an interdisciplinary concentration.

2.2 Typical Teaching Methods

A small school typically touts small classes, close student-faculty interactions, and personal attention. In this environment, large lectures are rare—at the very least, lectures have relatively few students. Instead, classes of 30 are considered large, and pedagogy often emphasizes active learning. Some examples from computing courses at Grinnell College typify this general environment.

- Only two courses in the entire computing curriculum (algorithms and theory of computation) meet in a traditional classroom; all other courses meet exclusively in a lab setting.
- Pedagogy typically changes class-by-class, according to the topic at hand. Activities may include computer-based lab exercises, small group activities, student presentations, class discussions comparing algorithms or addressing students' questions, problem solving on paper or on a whiteboard, etc.
- Even within a class session, activities may blend fluidly together. A class session might start with a 3-minute introductory talk and then move to a lab exercise. As questions arise, the class might reconvene for clarification or brainstorming, after which the lab activity might resume.
- Introductory computing courses emphasize collaboration and hands-on problem solving. A typical CS1 course might involve 45 lab sessions; one author estimates he lectures only about 4 hours per month when teaching CS1.

A common theme involves the engagement of students in the topic at hand. At Grinnell, it is often said that "students don't learn about science; rather students *do* science." In addressing social issues, faculty seek ways to involve students in discussion. Students should not be passive listeners, but rather active participants.

3. COMPUTING IN A SMALL, LIBERAL ARTS COLLEGE SETTING

At a liberal arts college, faculty across all disciplines understand the importance of social and ethical issues. Thus, the campus culture likely accepts the relevance of discussions of social and ethical issues of computing. At such schools, the more common issue is how computing fits into an overall education.

3.1 Where Computing Fits

At many liberal arts colleges, the computing faculty make the case that computer science fits within core areas, such as those listed earlier, by focusing on multiple approaches to problem solving, emphasizing underlying principles and underlying concepts, articulating precision of exposition (by writing in a precise [programming] language), addressing social and scientific problems raised in other disciplines, and demonstrating intellectual challenge and depth.

Further, computer science curricular recommendations (e.g., [1, 5, 9]) highlight theory, logical rigor, foundations, experimentation, analysis, and other intellectual skills that resonate in an academic community. At some schools, faculty must argue that computer science is different from vocational training, such as required for nursing, plumbing, or auto maintenance. Increasingly, however, colleges understand that computing connects well with the liberal arts; discussions such as that put forth by Walker and Keleman [13] can help educate any skeptics. Overall, in the liberal arts setting, faculty are comfortable discussing social and ethical issues of computing, once they agree that computing itself is worthy of study.

For a more extensive treatment of computing within the framework of a liberal arts institution, see [13].

3.2 Challenges

The small, liberal arts college setting poses some challenges for teaching computer science in general and for the inclusion of social issues in particular. Some challenges arise from institutional and departmental constraints:

- Majors are limited in the number of courses required, and so can require few specialized courses (e.g., on social issues).
- Schools expect students to graduate in four years, so required courses must be offered regularly.
- Faculties are small, so faculty scheduling must often focus on required courses.
- Offering courses in alternate years can provide options, but scheduling is delicate if such courses are required.

Although such constraints yield many consequences, one result is that very few computing departments in liberal arts colleges offer a separate course on "Social issues in computing." None are listed in [4], although courses on technology in society might be offered in a social studies or philosophy department.

Another challenge is to attract non-majors to computing courses. Liberal arts students typically have broad interests, but even students interested in science often have little or no knowledge of computer science. Schools with general education requirements may use them to channel non-majors into computing courses [4, 8]. At other schools, such as Grinnell, faculty must enhance the appeal of courses for non-majors, and advertise them to students and advisers. A discussion of social issues can provide one type of general appeal to this audience.

3.3 Opportunities

Within a small, liberal-arts setting, both faculty and students typically have a broad range of interests. Further, departments are small, and inter-departmental connections flourish. This framework provides numerous opportunities.

Many students who matriculate at liberal arts colleges wish to explore a wide range of disciplines before deciding on a major. Indeed, less than half of Grinnell's computer science majors entered Grinnell thinking they would major in computer science. The others try an introductory computing course and fall in love with the subject. With this breadth, liberal arts students often welcome discussions of social and ethical issues, and they bring knowledge and methods of inquiry from other disciplines. This greatly enhances discussions for all computing students.

Faculty, too, often have a broad range of interests and form close connections with colleagues in other disciplines. Interdisciplinary collaborations are not uncommon, and courses outside computing (e.g., in philosophy, sociology, or psychology) may build on these collaborations to discuss social and ethical issues of computing.

Since active learning techniques and discussion pervade many courses, both faculty and students are used to engaged discussion regarding both technical and social issues. In small classes, even more reserved students are drawn into discussions.

Moreover, faculty at small colleges often have a keen interest in teaching and may work together to develop their skills. For example, at Grinnell College, faculty from all science departments engage in a Science Teaching and Learning Group that has met biweekly for over a decade. This and other forums encourage exploration of creative pedagogy.

Finally, the atmosphere of many small, liberal arts colleges favor the consideration of social and ethical issues in all contexts. Recall Grinnell's mission to prepare students “for the honorable discharge of the duties of life." Grinnell, as one example, is noted for its commitment to social justice, and this commitment is expressed not only in students' extracurricular activities but also in courses across the college's entire curriculum. When social and technical topics are woven together through the careful selection of examples [8], faculty are more likely to be supported than censured by their colleagues for making this choice.

4. ENGAGING NON-MAJORS

As is typical at many liberal arts colleges, students at Grinnell do not have to declare their majors until their fourth semester. Thus, many first- and second-year students in introductory-level courses can be considered as potential majors. If students become excited about the subject, they might take more courses in the field and possibly declare a major. Thus, Grinnell's normal first course in computing emphasizes (functional) problem solving with applications to image processing. The course serves non-majors, but also helps recruit students to the major.

In addition, Grinnell's computer science department offers two courses that serve the general population: *The Digital Age* and sections of the First-Year Tutorial.

4.1 *The Digital Age*

Open to all students, *The Digital Age* provides background and insights for future citizens. In the early 2000s, the content of the course was strongly influenced by the 1999 report, "Being Fluent with Information Technology" [6]. This report identifies intellectual capacities, concepts, and skills that are vital to citizens in today's society. Over the years, the content and framing of the course has evolved, but the aim is to provide a fundamental background for future citizens. Social issues are addressed regularly (roughly weekly), alongside computer literacy topics. Instructors' approaches to addressing social issues have varied.

When teaching *The Digital Age*, Henry Walker connects topics in computing with common questions regarding technology, its applications, and consequences of its use. Social and ethical questions often can motivate discussions of computing principles, algorithms, and perspectives. For example, Web pages may refer to the user's city, and e-commerce sites may suggest related items after the user makes a purchase. These experiences raise important technical questions, such as, "How does the Web server know where I am?" and "How did the e-commerce site remember I had made that purchase?" Walker notes this question/answer approach is the basis for his book, *The Tao of Computing* [12], which he uses as an assigned text. This style of course naturally integrates technical material with discussions of social and ethical issues.

Marge Coahran and Jerod Weinman have used topical readings, discussions, and short essays to address social issues. In one offering, Coahran allowed students to choose topics from Sara Baase's computer ethics textbook, *A Gift of Fire* [3]; similarly, Weinman has assigned readings from Michael J. Quinn's *Ethics for the Information Age* [10]. Coahran has also selected accessible articles from the *Communications of the ACM* on perennial topics such as malware and privacy, less familiar topics such as e-waste and artifact preservation through 3D modeling, and timely topics such as the role of social media in Iran's recent election riots. Students are required to read selections before class and write one-page reflection papers. The instructor poses discussion questions either when the reading is assigned or at the beginning of class; students consider these questions in small groups.

Coahran also has assigned “tech news” presentations. Each week, a different team of 2-3 students chooses a current event in computing technology and presents this news to the class. Although discussions are not as in depth as for the assigned readings, this activity gives students a broad sense of the computing industry and its role in society.

4.2 Tutorial

Incoming students at Grinnell College are required to take a First-Year Tutorial. First instituted in 1971 to replace the writing and composition course formerly offered by the English Department, the First-Year Tutorial is now offered by faculty in each academic department. Although the subject matter may be anything of interest to the instructor, the course must include writing, oral presentations, and library research. Some recent titles offered by computing faculty include "Computers: Facts, Misconceptions, and Ethical Issues" (taught by Walker), “Virtue in Animal and Machine” (Weinman), "Technology and Place" (Janet Davis), "Free software, free culture" (John David Stone), "Owning the Intangible: Possession, theft, and (Mis)Appropriation of Ideas" (Samuel Rebelsky), and "Outward from the middle of the maze: Selected plays of Tom Stoppard" (Stone).

"Computers: Facts, Misconceptions, and Ethical Issues" was the most recent iteration of a tutorial that Walker has taught over a number of years. This course has faced several challenges.

- Incoming students have had increasing contact with computers, but most have had little formal study of computing. They usually accept viewpoints about computing from the popular press (often wrong), and they have little systematic foundation to discuss social and ethical issues.
- For years, incoming students have heard high-level ethical principles, such as "love your neighbor" and "do not steal". Walker notes that even fourth-graders are good at articulating moral imperatives and giving expected responses about good behavior. By the time they enter college, students are often quite adept at repeating platitudes. However, they often do not consider what those principles might mean in practice.
- Initial attempts to integrate isolated articles into the Tutorial readings met with mixed success. Students could read and discuss, but the readings did not seem to capture students' attention and interest.

The most recent iteration of this Tutorial addressed these issues directly in at least three ways:

- Quinn's textbook, *Ethics for the Information Age*, presents multiple ethical philosophies, and these viewpoints are then applied to a range of situations [10]. This textbook provided both a foundation for discussions of ethics and an organization for later class activities.
- Many classes were devoted to student-led discussions. At first, each student presented one ethical perspective from Quinn's textbook [10]. Later they led discussions of chapters, articles, and news stories. All of this helped students take ownership of the material and formulate their own responses.
- Over the semester, each student was asked to present two news reports that suggested something of popular image of computing (what computer science is, what computing professionals do, or how computers are used). Each presentation was to describe the basic facts of a story, but then also suggest 2-4 questions to guide student discussions.

While this Tutorial continues to evolve, the new format seemed significantly more successful than past formats regarding discussions of social and ethical issues. The textbook provided a foundation, student-led discussions encouraged ownership of ideas, and student selection of news stories ensured students were talking about topics they were interested in.

One possible shortcoming of a computing-focused Tutorial is limited appeal to students who are culturally distanced from computing, notably women. Topics that include computing as one topic among many, such as “Virtue in Animal and Machine,” “Technology and Place,” and “Owning the Intangible,” seem to attract more gender-balanced groups of students.

4.3 Assessment

Both *The Digital Age* and the First-Year Tutorial are stand-alone courses; they do not serve as prerequisites for later courses. Assessment, therefore, is self-contained within the course. Further, both courses emphasize oral and written communication skills; students are expected to participate in discussions, present to the class, and write papers based on class readings and outside research. In each course, students must delve into new topics,

synthesize ideas, formulate hypotheses, and justify their conclusions. Since *The Digital Age* and some Tutorial sections explore social issues, evaluation of oral presentations and written work allows assessment of the development of ideas about social and ethical issues. However, neither course utilizes in-class tests, so a strict quantitative assessment is not expected.

5. ENGAGING MAJORS

Majors in liberal arts programs can require relatively few courses for a computer science major: usually just 8-12 CS courses plus 2-4 mathematics courses, and rarely more than 12-14 major-related courses in total. Consequently, none of the programs discussed in [4] has a computer science course dedicated to social and ethical issues, and only Calvin College has an explicit requirement for the CS major regarding social issues [4]. Rather, social and ethical issues typically are discussed in other contexts.

- Computing may be addressed in courses outside of the computer science department, e.g., in a Philosophy of Science or a Technology Studies course.
- Themes or modules may be part of computing courses.
- Informal discussions may arise in a variety of settings.
- Advanced projects may include consideration of the work's social impact, or social issues may frame the overall project.

5.1 Regularly-Offered Courses

A typical approach for incorporating topics into existing courses is to construct a module or sequence of discussions on a specified topic. For example, Davis has students in *Computer Networks* read about the architectural principles that drove the design of the Internet forty years ago. In class, students consider the evolving uses of the internet, and technical implications of supporting (or thwarting) these uses.

Another approach is to inject questions about social and ethical issues as they arise. For example, in the context of *Computer Networks*, Rebelsky might ask his students, "What does it mean that we can gather this data?" or "Hmmm...we can forge email *this easily*. What implications does that have?" Walker finds that social and ethical issues arise naturally in *Databases and Web Application Design, Artificial Intelligence,* and the *Theory of Computation*. For example, in Web applications, it is natural to ask what consequences there might be to gather certain types of data, how that data might be stored (plain text, encrypted), what access there should be for that data, and what consequences might arise if the data were accidentally released. In *AI*, many techniques involve statistics or heuristics, and Walker asks students to consider possible procedures, legal responsibilities, and social impacts when results might be wrong for an individual. In the *Theory of Computation*, the Halting Problem demonstrates that some problems are unsolvable, and it is natural to ask about possible consequences for government policies and managerial decision making.

Faculty may also require students to do independent research on a socio-technical issue and present their findings to the class. For example, in *Software Design*, Rebelsky focuses on the potential impacts of the code one writes, particularly the danger of hidden bugs. In some offerings, he has had the students pick an incident (such as the THERAC-25 incident) and present it to the class. In *Computer Networks*, Davis has had students read and present research papers on network protocols designed to accomplish socially significant ends, such as onion routing, designed to provide anonymity to Internet users [11].

One course in particular is intimately related to the questions regarding quality of life and impacts on individuals and society posed by CC 2008 [5] and CC 2001 [1]: *Human-Computer Interaction*. Topics of special social relevance have included accessibility for people with disabilities, groupware, computers as social actors, persuasive technology, and ubiquitous computing. A focus on methodologies such as User-Centered Design, Value Sensitive Design, and Participatory Design lets students see how researchers and designers systematically consider stakeholders in the design of new technology. The course includes a project component involving system design and evaluation; to be successful, students must interact with stakeholders and consider the real-world use of the system they are working with. Moreover, students must consider ethical treatment of user study participants through through writing a proposal for work with human subjects and gaining approval from the Institutional Review Board.

Several computing faculty argue that social issues may arise in any course. Stone reports that he tries to teach by example, infusing throughout his courses his philosophy on social issues in computing, particularly with respect to intellectual freedom. Rebelsky writes, “I try to at least mention social issues in all of my courses; I think that considering social issues is an important part of the discipline.”

5.2 Reading Groups and Informal Learning

Small-group study and informal settings also can be important in raising social issues of computing. For example, in Fall 2005, Walker collaborated with a faculty member in education to organize an independent project to explore the literature on women in computing. In Fall 2007, three computing faculty responded to student concerns about the status of women in computing by organizing a weekly guided reading course. This fall, Davis is offering a course on *Socio-Technical Issues in Computer Networks*, similarly in response to student interest. The latter two courses were offered in the context of our informal “CS Table” lunchtime discussions, to which all students are welcome, regardless of whether they are enrolled in the course or even computer science majors. CS Table often touches on social issues even when there is not a special topic offering.

A small school can take advantage of its size and close interpersonal relationships by organizing such small group and community-building activities. This often requires some faculty time beyond officially recognized teaching duties. However, with relatively modest effort, social issues can be an ongoing theme both inside and out of the computing program.

5.3 Experiential Learning/Mentored Advanced Projects (MAPs)

Computer science students often undertake a range of independent projects under the supervision of a faculty member. In many cases, these projects provide excellent opportunities to raise social and ethical issues—not as abstract concepts but as points that have direct application to the work at hand. For example, Walker has led selected teams in developing software projects for record keeping for Grinnell's recruiting of student athletes, for Grinnell's local food cooperative, for an interactive campus map, and for placing incoming students in computer science, math, and

statistics courses. For each of these projects, the development team met periodically with clients to learn needs, to present prototype systems, to receive feedback, and to understand user priorities. Since each system was to support clients, discussions within the team regularly considered social and ethical issues, such as social impact, privacy, security, and authentication. In each case, social and ethical considerations had a significant impact upon software design and implementation.

Also through the Grinnell's program of Mentored Advanced Projects (MAPs), Davis has involved students in her research on applying participatory design methods to the development of persuasive technology: computing systems intended to change attitudes or behaviors. Persuasive technology raises a number of ethical issues relating to the intentions of the designer/persuader, the means of persuasion, and the desirability of outcomes, while participatory design is intended to help address these issues by involving potential users in deciding what behaviors the system will target and how the system will interact with users. Students contributed to developing and implementing participatory design approaches, and to considering the social and ethical implications of the resulting systems.

5.4 Assessment

Although this section has identified many ways that social issues can be addressed in courses, readings, and projects, formal assessment poses a challenge. Each individual course focuses on specific technical topics (e.g., operating systems, algorithms, networks), and formal evaluation (e.g., exams) typically focuses on those main topics. Framing an exam question related to social issues can seem awkward. Anecdotally, faculty can observe that students in one course are raising issues of privacy, security, intellectual property, etc. that have been raised in other courses, but assessment of these topics within individual courses is a challenge at best.

6. FUTURE WORK

The discipline of computing continues to develop at a rapid pace, and computing curricula and individual courses evolve quickly to keep up. Part of this evolution involves ongoing efforts to find creative ways to include social issues within courses and discussions. This paper has identified several approaches that work within a liberal arts framework, but the efforts will need to continue as both the discipline and society progress.

One particular area for future work involves assessment. This paper has documented problems in determining the effectiveness of individual courses, in which social issues form a background-level, on-going theme. However, one way to approach effectiveness might be to survey or interview graduating seniors and alumni regarding their exposure to social and ethical issues through the Grinnell computer science curriculum, their attitudes towards such issues, and their perceived ability to address such issues in their work and everyday life.

7. CONCLUSION

This paper has described the environment of the small, liberal-arts college and presented a range of approaches for introducing social and ethical issues of computing. These approaches largely entail readings and discussions that fit into study of traditional topics, but also include opportunities for study and discussion beyond the regular curriculum.

All of this work, however, is meant to provide a foundation. The comments here can provide a starting point for discussion and experimentation. The authors hope that this base leads to new conversations, as the start of expanded collaborations.

8. ACKNOWLEDGMENTS

This paper builds upon numerous conversations and activities of colleagues throughout the computing community. Many thanks to our colleagues, John David Stone, Samuel Rebelsky, and Jerod Weinman, for their on-going contributions in developing Grinnell's program. We also thank Marge Coahran, now at Dickinson College, for her insights, particularly in helping to develop *The Digital Age.* Beyond Grinnell, we gratefully acknowledge the contributions made over the years by members of the Liberal Arts Computer Science Consortium. We also want to extend our appreciation to Donald Joyce, Unitec, for proposing the ITiCSE 2010 Working Group on social issues, and to the members of that group for their encouragement and insights.

REFERENCES

[1] ACM/IEEE Joint Task Force 2001. *Computing Curricula 2001: Computer Science.*

[2] Alt, A, Astrachan, O., Forbes, J., Lucic, R., and Rodger, S. 2006. Social Networks Generate Interest in Computer Science, *SIGCSE 2006 Proceedings*, pp. 438-442.

[3] Baase, S. 2008. *Ethical Issues for Computing and the Internet Second Edition.* Prentice-Hall.

[4] Baldwin, D., Brady, A., Danyluk, A., Adams, J., and Lawrence, A. 2010. Case studies of liberal arts computer science programs, *ACM Transactions on Computing Education (TOCE)*, Vol. 10, Issue 1 (March 2010), Article 4.

[5] Cassel, L., et al 2008, *Computer science curriculum 2008: An interim revision of CS 2001. Report from the interim review task force.*

[6] Computer Science Telecommunications Board 1999. *Being fluent with Information Technology*, National Research Council.

[7] Guzdial, M. and Erickson, B. 2010. *Introduction to Computing and Programming in Python: A Multimedia Approach*, Second Edition, Addison-Wesley / Prentice Hall.

[8] ITiCSE 2010 Working Group 8 2010. *Enhancing the Social Issues Components in our Computing Curriculum: Computing for the Social Good,* Report in preparation.

[9] Liberal Arts Computer Science Consortium 2007. A 2007 model curriculum for a liberal arts degree in computer science, *Journal on Educational Resources in Computing (JERIC)*, Vol. 7, Issue 2 (June 2007), Article 2.

[10] Quinn, M. J. 2009. *Ethics for the Information Age, Third Edition,* Pearson / Addison-Wesley.

[11] Reed, M. G., Syverson, P. F., and Goldschlag, D. M. 1998.Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications,* Vol. 16, Issue 4, May 1998, pp. 482-494

[12] Walker, H. M. 2005. *The Tao of Computing,* Jones and Bartlett.

[13] Walker, H. M. and Kelemen, C. 2010. Computer science and the liberal arts, *ACM Transactions on Computing Education (TOCE)*, Vol. 10, Issue 1 (March 2010), Article 2.