3-1996

Teaching Ethical Issues in Computer Science: What Worked and What Didn't

Kay G. Schulze

Frances Grodzinsky
Sacred Heart University, grodzinskyf@sacredheart.edu

TEACHING ETHICAL ISSUES IN COMPUTER SCIENCE: WHAT WORKED AND WHAT DIDN'T

Kay G. Schulze
Computer Science Department
United States Naval Academy
572 Holloway Road
Annapolis, MD 21402-5000
schulze@nadn.navy.mil

Frances S. Grodzinsky
Computer Science & Information Technology Department
Sacred Heart University
5151 Park Avenue
Fairfield, CT 06432
grodzin@grodzinsky.sacredheart.edu

INTRODUCTION

During the summers of 1994 and 95, the National Science Foundation sponsored a week long workshop at Rensselaer Polytechnic Institute on teaching computer ethics[1]. These workshops were led by Deborah Johnson, a philosopher at Rensselaer, and Keith Miller, a computer scientist at Sangamon University. The goal of the workshop was to educate undergraduate computer science teachers in some of the relevant ethical issues in computer science and to train them in how to teach these issues to their students. Twenty-five computer scientists, philosophers and social scientists from undergraduate institutions attended the first summer and twenty-six attended the second. Eighteen of the first group met again in the summer of 1995 to assess and discuss their experiences. As members of the first group of educators, we would like to report on which activities were particularly successful, which ones were not and why.

The materials presented in this paper were developed by participants in the 1994 workshop, and used in colleges and universities of varying sizes and populations. Some were used in stand-alone courses in computer ethics while others were modules in various traditional, computer science courses. Prior to the workshop, none of the professors who developed the material had had any more education or experience in philosophy or computer ethics than the average computer science professor. Our goal was to have our students develop the skills to 1) argue from example, analogy and counter-example, 2) identify ethical principles and stake holders in concrete situations, 3) identify and evaluate alternative courses of action, and 4) apply ethical codes to concrete situations [11]. Ultimately these skills,

SIGCSE '96 2/96 Philadelphia, PA USA

[1] Information on the 1996 Workshop can be obtained from Deborah Johnson at johnsd@rpi.edu.

we hoped, would give our students the confidence and ability to establish their own set of ethical principles.

Before students can start discussing ethical issues that arise in computer science, such as privacy, intellectual property rights, reliability and responsibility, it is imperative that they be given an understanding of the fundamentals of the major ethical theories. Most computer ethics texts [8] [10] [12] offer brief descriptions of utilitarianism, relativism, and Kant's deontological theory. These are illustrated by presenting varying solutions to a problem within a scenario as argued by a utilitarian, Kantian or relativist. It often helps to give outside readings on Mill and Kant, either as primary or secondary sources, to further explicate the theory behind the solutions. While philosophical readings can be complex, they can be clarified by classroom discussions and interactive activities.

It is crucial that students understand that irrespective of the viewpoint, ethical solutions must be consistent, coherent, and defended with reason rather than emotion or intuition. There is a fine line between building a solid ethical framework and confounding the students with so much philosophy that they lose interest. By recognizing the limitations of only using a "quandary" approach to ethics, and drawing on theorists ranging from Aristotle to Bellah for richer notions of the self, virtue, and the common good, we assist students in appropriating a more compelling set of ethical convictions. Students who understand the difference between solving a quandary by applying ethical principles to it and having ethical principles that can be applied to all problems will be well on the road to becoming ethical people who by extension will become ethical computer scientists.

SUCCESSFUL PEDAGOGICAL METHODOLOGIES

Seven major pedagogical methodologies evolved as the most successful. 1) Present the material using an interactive discussion or dialogue. This usually works best when a specific scenario or reading is used as the focal point. The students can be divided into two groups where one group argues the *pro* position and the other the *con* position. If the topic is controversial, the students may

naturally separate themselves into groups. It is critical that the students be made to articulate and defend their positions. If the students are permitted to simply state what they "feel" without being made to defend it, the dialogue will degrade into simple vote taking. One scenario that generally produces a high level of discussion is "The Case of the Killer Robot" [2].

2) The use of analogies is successful in clarifying ethical and social issues. For example, a student might claim it is acceptable to "snoop" in someone's computer files if they are not protected, particularly if the intruder doesn't disturb anything. At this point, suggesting to the student that with that line of reasoning it would be permissible to walk down the street, try the front door of every house, and upon finding one unlocked, walk in and browse around as long as nothing is disturbed [8]. Using this type of analogy normally provokes heated discussions. Drawing a parallel between ethical issues and "real life" situations can make the students understand the importance of critically considering all the issues.

3) Critical thinking and writing assignments are also effective. If the students are asked to write a short paper supporting their position or debating the pros and cons of an issue, they begin to develop the skills to analyze an ethical issue. These assignments can be impromptu writing or homework assignments and are best focused on a scenario or journal article. Another effective individual assignment in a stand-alone computer ethics course is a student journal [3]. The student can be asked to include in the journal one or two newspaper or magazine articles a week that present, in their mind, an ethical problem. They would present the problem and then evaluate the solution if one is given, or present a possible well-reasoned solution of their own.

An effective group writing assignment is to require the students to do a social impact statement on a real system on campus. The basic idea of a social impact statement is modeled on the environmental impact statement, and is an organized and coherent look at the social and ethical implications of a particular computer system [7][14]. Developing the policies and rules for a computer laboratory or campus-wide computer system is another group project. Students can investigate the policies at other universities via the WWW and divide into groups which can write up student responsibilities, system operator responsibilities and university responsibilities. This can be done in an on-line forum or in class.

A critical thinking assignment might consist of the student conducting a field interview and then analyzing the results. Another critical thinking activity is an ethics quiz. The class is presented with various situations and is asked to decide whether the situation poses an ethical question. If this is done early in the term, it provides the instructor with guidance on what needs to be covered and it helps the student learn the first step in the process of making ethical decisions.

4) The use of an on-line forum is another method. It can be used to discuss articles from newspapers and magazines or to continue in-class discussions that simply ran out of time. Most students are very comfortable with this method and many students who may be too shy to actively participate in a classroom discussion are usually more willing to participate on-line. This activity can take the form of a bulletin board, newsgroups or simply email. Creating a forum between two different institutions is also enlightening and exciting for the students.

5) Another method is classroom presentations and they can take two different forms. One is to have the students act out scenarios and develop a set of discussion questions. After the "playlets," the class can be broken into small groups to discuss the questions and one member of the cast can serve as a consultant for each group. The other form is to assign the students the responsibility of teaching one hour of the class. If this is made competitive by requiring each student to submit his/her lesson plan and then selecting the very best, one can get some ingenious ideas. For example, one student developed a mock trial on computer liability - who is responsible when something goes wrong? He had members of the class play the judge, prosecuting and defending attorneys, the jury and the accused and had made up cue cards so that each player knew what to say. Another student created a video presentation on virtual reality.

6) Inviting a guest lecturer to class is also effective. For example, when discussing licensing of computer professionals, one might invite a member of another licensed profession, such as an architect. This provides the students with an outside view of why licensing is important, the responsibilities accompanying it, and allows them to discuss whether it should be applied to the computer profession.

7) The most subtle method is the use of a programming assignment. Students are generally so eager to begin coding that it is difficult to convince them to spend sufficient time on design. Therefore, they seldom, if ever, consider the ethical or social implications of the program they are writing. But once the program is written, it can become the focus of an ethical discussion. For example, assigning a beginning programmer the task of practicing *for loops* by writing a simulation of the positions of the arm of an x-ray machine provides an excellent situation in which to discuss reliability and responsibility [5]. This is particularly true if their code did not check to see whether the patient was off the x-ray table before their code lowered the arm to the table. Another is to have students design interfaces for some "critical" applications. This leads to discussions of responsibility vis a vis safety issues, layout, ease of access, gender bias, etc.

Clearly some of these methodologies lend themselves to individual activities while others lend themselves to group activities. The dialogues and on-line discussions, and role playing are more effective with a group. Writing

assignments, field interviews, class presentations, and programming assignments can be either group or individual efforts.

WHAT DIDN'T WORK

One problem area is writing assignments, particularly early in the term. Many of us discovered that students had difficulty determining whether an ethical situation existed and many did not know how to construct a logical argument. One suggestion is that class time be used early in the course to do practical syllogisms and exercises and to dissect the arguments. Fulda's [4] article can be used to dissect each sentence/paragraph looking for positions that were and weren't well-supported and where inconsistencies existed. Students can then model their own analyses after this approach. Many of us found that major term papers did not work well because students showed very little ingenuity and viewed them as a chore. Writing assignments of 1-2 pages were much more successful.

A problem that can easily arise is what to do when the discussion dies. This usually happens because the students have few opinions, no passion about the subject, or haven't done the reading. One suggestion is to pass out questions before the assigned readings and use these as a springboard for the discussion. Another suggestion is to require each student to arrive at class with an opening question. Then go around the class and have each student read his/her question. Generally there will be enough interesting questions to spark discussion. The last suggestion, which is possibly the hardest for an educator to do, is to do nothing. Students generally cannot tolerate silence and if the instructor just silently counts to one hundred, someone will speak up. However, keep in mind that the discussion may have died because the students are having trouble connecting the class discussion with the material in the reading or to real life and are looking to the instructor to help them make the connection.

Another problem can occur when the entire class takes the same side of an issue. The best solution to this is to be well-prepared with arguments on both sides and to use exaggerated situations. For example, in discussing intellectual property rights, if the best one can get from the student who makes only one copy is an admission of theft, suggest that 100 copies were made. This might spur them to consider the effects of the action. A similar situation can arise when the students are not all majors. Be careful not to create groups where majors and non-majors are polarized. The majors can infuse technical details that leave the non-majors helpless.

The bright, articulate student who dominates the discussion and intimidates the rest of the class can cause another problem. This can be difficult to handle, but one can simply articulate other positions or elicit positions from the rest of the students. When a student makes an absolutely outrageous statement, a very effective way to handle it is to pause, and then ask the student to either rescind his statement or defend it by giving reasons. If it is truly outrageous, he generally can't defend it.

International students with limited English-speaking skills, different learning styles, and/or non-Western philosophical backgrounds can pose unique problems. They are often intimidated in interactive classroom situations. However, in discussing philosophical theories, these students, if they are coaxed to participate, can offer their own sets of beliefs which are usually drawn from their religious backgrounds. American students are generally amazed to discover the similarities in the teachings of the Eastern philosophers and the Western ones. With their interest piqued, these students often encourage the international student by asking more questions. International students who have learned never to disagree in public with the teacher will find it difficult to adapt to a vociferous classroom argument especially if the professor is involved. Gentle soliciting of opinion through round-robin questioning will often make the student more comfortable and eventually s/he may become involved in the argument.

Limited English often makes writing assignments difficult for international students, especially when they are asked to offer an opinion or defend one. Poorly written papers can be misinterpreted by the professor as a lack of understanding or interest on the part of the student when they may be neither. When written assignments are given, students who need language help should be encouraged to seek it. If the class is small, the professor may offer to review a first draft. Papers with unacceptable English should be returned to the student with a note to seek appropriate language help and to rewrite the draft. It is more difficult for students of some cultures to adapt to the American style of education than others. It is a challenge to the professor to involve these students by offering encouragement and helpful criticism.

FEARS AND ANXIETIES

Three major anxieties and inadequacies appeared in most of the workshop participants. The first concerned pedagogical style. The consensus of the group was that traditional lecture style presentations, common in the computer science discipline, are inadequate for presenting this material. As a result, much of the anxiety felt by the participants was due to the need to explore more interactive classroom techniques. The only thing that will remove this anxiety is practice. Leading a discussion is no harder than giving your first lecture, and it does get easier with experience.

The second major problem was the inadequacy the participants felt in grading students' written work. The best approach is to articulate the purpose of the writing assignment and to base the grade on that aspect of the paper. For example, if the students are asked to write analogies, then evaluate them on the success of that task. Although it is important to mark the grammatical errors, it is also important to get beyond the syntax and place a

heavier concentration of the grade on the specific skills you want to develop. Often short weekly papers can be evaluated on an acceptable/unacceptable basis. By focusing on the task at hand, and insisting that the students cite from their readings, professors can easily evaluate whether the student has done a good job. Longer papers can be assigned letter grades. Writing Across the Curriculum workshops offer support to faculty wishing to incorporate writing into their disciplines.

The last anxiety expressed by participants at the workshop was the lack of training we had in ethical theories. One advantage computer scientists have is that our discipline is one in which we are constantly learning new material. Hence, this inadequacy can easily be overcome. There are several good textbooks [6][8][9][10][12] that can be used in classes or as a starting point for self-education. In addition, the WWW [13] is rapidly becoming an excellent resource as more and more people are sharing their modules and syllabi. If your university will allow it, team teaching with a professor of philosophy, ethics or religious studies who is interested in applied ethics is invaluable.

CONCLUSIONS

Ethical issues are not going to disappear from the computer science discipline. Both Computing Curricula '91 [15] and CSAC/CSAB accreditation [1] recommend or require the inclusion of ethical and social implications in computer science in the undergraduate curriculum. We as educators can no longer pay lip-service to this area of study. It is far better that we ensure that our students grapple with ethical dilemmas while they are learning the discipline rather than take the chance that they will not be able to identify, let alone resolve, ethical problems once they are in the field. If students develop their own set of values as well as the skills necessary to recognize, analyze, and solve ethical problems, it is more likely they will not compartmentalize computer ethics as something that is irrelevant to their profession. Although it is not an area in which we are experts, nor a pedagogical style with which we are particularly comfortable, the teaching of computer ethics is nevertheless our responsibility. It is our role as computer science educators, to ensure that our students have a firm foundation in the social and ethical issues of the discipline.

ACKNOWLEDGMENTS

The authors wish to acknowledge Deborah Johnson, Keith Miller and their fellow participants in EPIC94. Particular thanks are extended to the participants in the follow-up workshop for providing many of the methodologies and examples incorporated in this paper: Peggy Eaton, Ed Gehringer, Steve Grodzinsky, Linda Herndon, Chuck Huff, Laurie King, Joe Kizza, Pat Lapczynski, Debra Lelever, Dee Medley, Russell Mills, Ralph Morelli, Bob Riser, Tony Robbi, Becky Tidwell, Deb Trytten, and Laurie Werth.

BIBLIOGRAPHY

1. Computer Sciences Accreditation Commission, "CSAC Accreditation Criteria", Stamford, CT, 1995.

2. Epstein, Richard G., "The Case of the Killer Robot," Computers & Society 24(3): 20-28, September 1994.

3. Fedor, John, Personal communication, 1995.

4. Fulda, Joseph S., "Technological Advances: Who Benefits Most?," *Computers & Society*, 24(4), 6, 1994.

5. Gotterbarn, Don, Personal communication, 1995.

6. Huff, Chuck and Finholt, Thomas (Eds.), Social Issues in Computing: Putting Computing in Its Place, McGraw-Hill, New York, NY, 1994.

7. Huff, C. W. and Jawer, B., "Toward a Design Ethic for Computing Professionals. In C. W. Huff and T. Finholt (Eds.) Social Issues in Computing: Putting Computing in Its Place, New York, NY: McGraw-Hill, 691-713, 1994.

8. Johnson, Deborah G., Computer Ethics, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1994.

9. Johnson, Deborah G. and Nissenbaum, Helen (Eds.), Computers, Ethics & Social Values, Prentice Hall, Englewood Cliffs, NJ, 1995.

10. Kallman, E. A. and Grillo, J. P. Ethical Decision Making and Information Technology, McGraw-Hill, New York, NY, 1993.

11. Martin, C. W., Huff, C. W., et al., "Computers in Context: A Framework for Presenting the Social and Ethical Impact of Computing", A Report from the ImpactCS Steering Committee, May, 1995.

12. Oz, Effy, Ethics for the Information Age, Business and Educational Technologies, 1994.

13. Schulze, Kay G., "Computer Ethics," http: //www.scs.usna.navy.mil/users/schulze/ethics/ethics.html

14. Shneiderman, B. "Humans Values and the Future of Technology: A Declaration of Empowerment. *Computers & Society*, 20(3):1-6., October, 1990.

15. Turner, A. Joe, "A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report: Computing Curricula 1991," Communications of the ACM, 34(6), 69-84, 1991.