Reshaping the Image of Computer Science in Only Fifteen Minutes a Week

Sara Sprenkle
Washington & Lee University
Lexington, VA 24450
sprenkles@wlu.edu

Shannon Duvall
Elon University
Elon, NC
sduvall2@elon.edu

ABSTRACT

Undergraduate enrollments in computer science have declined since the dot-com bust. Some reasons for the decline are computer science's nerdy image, lack of understanding of the field, and low motivation for learning programming. We propose to change the image of computer science by exposing students to applications of computing and its impact on their lives through reading and discussing recent news articles in 15 minutes of class. We call this component of our courses the *Broader Issues* in computer science.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Literacy*; K.4.0 [**Computers and Society**]: General

General Terms

Human Factors

Keywords

Current events, Engaged learning, Computers and society

1. INTRODUCTION

Enrollments of undergraduates in computer science have been declining since the dot-com bust [15, 4]. Declining enrollment is partially attributed to the image of computer science as nerdy and limited to programming [8, 10]. Women in particular are more interested in computer science when they see the field in terms of real-world applications that benefit mankind [10]. Some groups are working to change computer science's image, with initiatives aimed at motivating the study of computer science and busting the nerd stereotype [5], but these approaches have not necessarily been integrated into the computer science curriculum. Often the real-world issues are only discussed in an ethics unit rather than integrated into the core topics of programming and algorithms. We propose to change the image of computer science by exposing students to applications of computing and its impact on their lives through reading and discussing recent news articles in 15 minutes of classtime. We call this component of our courses the *Broader Issues* in computer science.

The Broader Issues component has several goals. First, **the articles should help students put abstract class material into real-world context.** Reading and discussing news articles is an easy way to introduce the human element and relevance to society in an established course. We are careful to choose articles that directly connect to the class material so that the articles help the students see the large-scale application of their small-scale assignments.

Secondly, **articles should show the broad range of computer science applications and opportunities**. In a study of high school girls, Jepson and Perl found that two reasons girls do not choose to study computer science is that there are not enough computer scientist role models and that the girls do not know much about the field of computing [11]. One way to appeal to women in computer science classes is to "humanize" the course [6]. That is, women are more engaged in material which has an obvious connection with people's lives because "they are not interested in computing for the sake of computing" [10]. News articles are an easy and unintimidating way to learn about the field and its major players.

Thirdly, the addition of the Broader Issues component should **improve students' written and oral communication skills**. Low-stakes writing assignments allow students to practice focused, analytical writing—without high grading costs [9]. Furthermore, Waite et al. advocate a "Conversational Classroom," citing that their course was more successful when classroom discussion was encouraged [17]. Students learned to engage one another's opinions and ideas as well as reflect on their own. Including the Broader Issues component in class allowed us to create a conversational classroom. Students have time to read the article, research more information if they are interested, write about their opinions on the issue, and read the opinions of others before coming to class to discuss the article. As a result, class discussions are a lively exchange of ideas and a time for students to draw parallels from the reading to their other course work and articles. The discussion atmosphere also encouraged students to open up to one another and gel as a group, warding off the typical competitive and intimidating atmosphere.

The last and perhaps most important goal for the Broader

Issues class component is **to expose students to how a computer scientist thinks and feels about current issues**. The articles we use in the class highlight the issues we are naturally interested in as computer scientists. In class discussions we can model the thinking process we use to evaluate the information in the article. Learning "disciplinary thinking" has been argued to be more important in a course than the memorization of facts or the explanation of concepts [7, 12]. The course content is only one aspect of class, and students "should learn to think as a member of a particular disciplinary community and to be familiar with the types of debates in which members engage" [7].

2. COURSE ACTIVITIES AND LOGISTICS

The Broader Issues component of the class consisted of three parts: reading, writing analyses, and discussing the articles in class. These activities happened on a weekly basis. We discuss the approach as we applied it in CS0 and CS1 courses, but we believe a similar approach can be taken in other courses as well.

2.1 Articles

We selected articles that were relevant to the course, were easy to read and short (generally taking less than fifteen minutes to read), covered a variety of topics, were not narrowly focused, and that we believed students would find interesting.

The articles frequently came from articles that we read on Wired, ACM Communications, or ACM TechNews. After colleagues learned about our use of articles, they sent articles that they liked and/or had discussed in class. For some topics, we chose several articles, which had different angles on the week's topic and may interest different students. For example, students could choose between two articles on the topic of environmental monitoring: a more technical one on volcanoes and a less technical one on zebras. Students looking for a more challenging read or were more interested in volcanoes than zebras could read the more technical article.

We scheduled reading articles when they were timely (e.g., electronic voting near election days) and near relevant course material. For example, in CS1, we read a DARPA Urban Challenge article after covering `if` statements. Students are thrilled to read from a real-world researcher, "We were just one IF [statement] away from success" [16]. The article also emphasizes the importance of testing and debugging, supporting our requirement for demonstrating good testing in their submitted assignments.

Examples of our articles can be found at `http://computersciencenews.blogspot.com/`.

2.2 Short Writing Assignments

Students write summaries as comments on a blog so that other students can read their entries and students get experience using a blog. The summaries must contain the following information for full credit:

- How interesting this article is on a scale of 0 to 9
- The three most important points in the article(s)
- How the article affected your understanding of CS and how computing relates to other areas
- How the article relates to our course
- One question for class discussion

Writing serves several purposes. First, we can verify that the student read the article. Second, students practice writing, which many colleges emphasize, especially in the sciences. Third, we can see if the students understood the relationship between the article and the class. Fourth, we get input from students on what would make for interesting discussion. Fifth, if discussion lulls, we can ask specific students about points in their blog.

2.3 In-class Discussions

In class discussions, we posed several questions, students volunteered answers, and discussion usually flowed from there. To facilitate discussion, we tried to maintain a balance so that no student dominated the conversation. We posed new questions when discussion waned, sometimes played devil's advocate, and always tried to challenge the students' thinking.

The articles spark discussion on issues traditionally discussed in introductory computer courses, such as efficiency and cost tradeoffs and testing challenges, but also unexpected topics, such as as usability, privacy, security, ethics, and economic and social policies. Students do not expect to discuss such issues in a computer science course, and they can relate their experiences and expertise from other areas to their computer science course.

In the CS1 class, we also had small group (3-4 students) discussions. The purpose of the small groups was to encourage shy students to participate. The instructor chose the small groups before class, arranging different groups each week, such as randomly, using the modulo operator, by gender, by year, by article read, or by opinion.

2.4 Grading

Students were evaluated on their blog entry and their participation in class discussion. The blog entry was weighted more than participation because students can participate in discussion without reading the article. We graded blog entries based on their clarity, understanding, thoughtfulness and English grammar and the entry's completeness. The blogs are quick to read and to grade, and it was usually evident if the students were making connections between the articles and the course content. For the discussion grading, students were usually given full credit for participation if they said something during discussion. Even though the Broader Issues were worth less than 10% of the students' course grade, the students still took the assignments seriously.

2.5 Extra Credit

We gave students extra credit opportunities based on the Broader Issues component. Students could find new articles that interested them, summarize them, and write about how they found the article and why it interested them. While some of the articles students found were too narrow to be used for the class, many of these extra credit articles can be used in the current or future classes.

3. EVALUATION AND OBSERVATIONS

In this section, we describe our methodology for evaluation and provide quantitative and anecdotal evidence of the effectiveness of including the Broader Issues in our courses.

3.1 Methodology

We evaluated our approach in one CS0 course and two CS1 courses at two small liberal arts universities. All classes were small and populated mostly with students who were not planning to major in computer science but rather were completing a general education requirement or elective.

We used two surveys to evaluate our approach. At the beginning of the semester, students filled out a survey about their backgrounds, their perceptions of computer science, what computer science majors do after graduation, and how computer science related to their lives, if at all. At the end of the semester, the students completed an anonymous evaluation, where they rated various parts of the Broader Issues component of the course and reflected on their changed perceptions of computing. In addition, part of the students' written assignment was to rate their interest in that week's article. We used this information to help identify trends for what types of articles would be most engaging in future courses.

3.2 Results

We gathered both statistical and anecdotal evidence to see if our goals for the Broader Issues component were being met.

3.2.1 Logistics of Readings and Discussions

The majority of students rated the quality of readings and large class discussions as "very good" or "excellent". However, students found that small-group discussions were mediocre. We believe that the classes were too small to merit breakout groups, and smaller group sizes may have made some students feel pressured to carry the conversation.

All students reported spending on average an hour or less reading and summarizing the articles; CS0 students all reported spending less than 30 minutes. Some of the CS1 articles were more technical and, therefore, more challenging to read and summarize. Students reported that they would like more emphasis on the discussion in class as well as more weight on this component in their final grades.

Women comprised half of the CS0 students but less than one-third of the CS1 students. In both courses women responded well to the Broader Issues component, writing thoughtful blog entries and participating in discussion. In personal communication, one woman math major in CS1 credited the Broader Issues as her reason for continuing on to the next computer science course.

The following quotes from students further illustrate their enthusiasm for the Broader Issues component:

- "This was my favorite part of the class!"
- "I thought the readings were always good and pretty interesting and they were a nice break from the programming."
- "It was good that we kept up to date with the latest news in the computing world."

3.2.2 Article Interest

Most students rated their overall interest in the articles as either "very good" or "excellent". No individual article in CS1 scored on average less than 5.5 on a scale of 0 to 9. CS1 students found the articles on Privacy and Security in Facebook [2], the One Laptop Per Child project [14], and the DARPA Urban Challenge [16] the most interesting, averaging over 8. CS0 students were most interested in the articles on Electronic Voting [1], the One Laptop Per Child project [3], and Steganography and Terrorism [13]. Even if students were not interested in an article, they often recognized its value, as evidenced in a student's blog entry: "While the article is worth reading, I did not find it to be interesting." The following student quotes show that most did in fact find the articles interesting as well as worthwhile.

- "The articles regarding steganography and the XO laptop were really relevant and interesting, a rare combination."
- "I really enjoyed the articles. It was interesting to see how computing really effects our lives in current times, and the discussions were also a nice change from the regular lecture."

3.2.3 Communication and Writing Skills

While it is not clear that the students improved in terms of grammar, the students' quality of answers to discussion questions improved over the course of the semester. They were better able to compare articles and draw conclusions at the end of the semester. In addition, the students were able to more clearly and precisely construct their answers.

Students often wrote thoughtful entries that revealed more about themselves than we would typically expect in a class. However, students did not appreciate being evaluated on English grammar and clarity in a computer science course.

While some articles generated more interest than others, each discussion was lively and there was never a problem getting most students to participate. Since some students are naturally more opinionated and outspoken than others, ensuring that everyone had a chance to be heard in 15 minutes often presented a challenge. Small-group discussions were employed to include everyone but were only mildly successful. Shy students still often hung back from the discussion, while the other students preferred the larger group setting. Small group discussions worked best when the groups felt safer (such as grouping by gender) and when the small group discussion had an agenda (e.g., they had to prepare to debate other small groups). We were pleased to find that students had civil discussions: even when there were dissenting opinions, the students were never rude.

3.2.4 Impact on Understanding of Computer Science

The CS0 and CS1 student responses differed most widely about how the articles impacted their understanding of computer science. In the CS1 classes, 62% rated the articles' impact as "excellent" or "very good" and only 16.7% rated it as "fair" or "poor". In CS0, however, only 19.1% of students rated the articles' impact as "excellent" or "very good" while 47.6% rated it as "fair" or "poor". We believe there are several reasons for this. First, the CS0 course is a non-programming breadth course: most topics are fairly easy to understand and the course is designed to show the variety of topics in computing. The CS1 course, however, is an introductory programming course and the Broader Issues component is the only connection in the class between the programming topics and their wider applications. It is not surprising that the articles make a bigger impact on the CS1 students' understanding of computer science.

However, students did not always make the expected connections to the articles. For example, late in the semester some CS1 students thought the answer to the "relevance" question was "they probably used this week's data structure to implement this program." While that may be true, we were often looking for a more specific connection to our class's concepts. For example, the Facebook article discussed how Facebook uses frequencies in its algorithms [2], and we had calculated frequencies in a recent assignment. Furthermore, some students talked about how complicated Facebook's algorithms are. We wanted to take the mystery and difficulty out of these applications, hoping they could see that an algorithm in a recent assignment could be scaled up to do something similar to Facebook's algorithm. In addition, some, students did not critically analyze the article and could not draw conclusions beyond what the author concluded, making their grade on the writing portion lower. We used class discussion to address these misunderstandings.

In the following quotes students comment on the impact of the articles and discussions on their understanding of computer science.

- "I love this part of the course. It's nice how this part allows you to see real-life examples of what you do in class."
- "I thought it was a great element because it made the material we are learning more real to me."
- "I truly enjoyed this component of the class and it definitely served to put some perspective on the things we were doing in lab and learning about in class."

3.2.5 Disciplinary Thinking

Perhaps the most difficult goal to assess was students' learning to think like computer scientists. Certainly the goal of disciplinary thinking is beyond the grasp of many non-majors after one course in computing. We did, however, find some students in each class that seemed especially intrigued by the Broader Issues component of class. Sometimes, students would write long blog entries on everything the article made them think about. Some students would get interested and follow links or find related articles to the given topic. In two separate classes, a student who stopped coming to class and turning in assignments still read articles and posted blog entries. Sometimes, we would walk into class and students were already discussing the week's article, or the students would leave class talking about it, or they mentioned talking to other students not in the class about the article.

In general, even if a CS1 student was struggling with the course content, the readings and participating in class discussion would bring them back into their comfort zone. They enjoyed making connections between course content and an article. For example, many CS1 students are thrilled that the `If` statement is mentioned in the DARPA Urban Challenge article [16], that Excel 2007 was released with a bug involving binary numbers, and that the One Laptop Per Child project uses Linux and Python.

Perhaps most compelling are the insights of some students. For example, after reading articles about the One Laptop Per Child (OLPC) project and gender differences in computing, one student drew a parallel between the two that amazed us. He stated that the OLPC project's biggest challenge is that the local leaders fear the change it represents—a shift in power. He then hypothesized a similar fear in traditional computer scientists that, consciously or not, may keep them from greater efforts to diversify computing.

Other survey responses lead us to believe that students are learning more about what problems interest computer scientists and what the field of computing is really like.

- "Computer scientists aren't sitting around programming machine language but are instead involved in a wide variety of topics solving problems in just about every field."
- "I can now identify the reaches of computer science in society, and I also found computer science to be fascinating since it is in essence another way of thinking, something which is beneficial in any field."
- "In some sense, computer science is perhaps the most pure of the sciences, simply asking 'what is the most efficient, most precise, overall best way to solve any given problem?' regardless of what that problem actually is."

3.2.6 Perceptions of Computer Science

As expected, at the beginning of the course, students had a variety of perceptions about computer science. Many mentioned that they use computers and applications daily, that computer scientists program, and that computer science majors become programmers. A few who had taken computer science courses before mentioned algorithms and problem-solving. Some advanced math and science majors mentioned how programming would help them in their research or coursework. At the end of the course, however, it was clear that the Broader Issues component of class had indeed broadened their perceptions of the field, as evidenced by the following quotes.

- "Throughout the course I have become aware of the many possibilities and applications that computer science and programming provide. I never would have known that I could have a basic understanding of the programming behind automated cars or wireless sensor networks just by learning basic for loops and if statements in Python."
- On Voting: "I think that this article changed my views on computing. I entered this class with little to no knowledge of computing and the informational age. I am shocked to find that such programs can be uploaded so quickly into a supposedly 'airtight' system."
- "The Broader Issues portion of our course, however, has brought me to the realization that there is much more to the Computer Scientist than meets the eye... I typically think of Computer Scientists as people who create programs for their own profit, but the 'One Laptop Per Child' article proved otherwise."

It is clear to us that the Broader Issues component of class met the goals we had set. In the next section, we provide more details on lessons learned from our experience and possible pitfalls of our approach.

4. LESSONS LEARNED AND CONCLUSION

We would recommend that others introduce reading and discussing news articles into their computer science courses. To help others apply a similar technique, we have compiled some advice based on our experiences.

Begin keeping a log of interesting articles. When you come across an interesting news event, write the link as well as a few keywords of how the article connects to your course. Soon you will have plenty of articles to get started. You do not need to have all the articles in place at the beginning of a course—you or your students may find new ones during the semester.

We graded blog entries using a points rubric. While grading was done quickly, it can be even faster using a "check" system. Give checks for satisfactory/typical work, a check-minus for unsatisfactory work, a check-plus for exemplary work, and a 0 for incomplete. In addition, not every blog needs to be graded for every student every week. Random sampling for grading tends to work well since the blog posting is due each week. It is also unclear whether the discussion portion of the component needs to be graded. In the past we have graded based simply on attendance or participation. Since neither of these have been a problem in class, that part of the grading can probably be dropped.

As discussion facilitator, you have to balance letting students lead the discussion and preventing tangents. You also have to play devil's advocate sometimes and support opinions that you don't believe. It is helpful for the students if they know that your contribution to the conversation is not what you personally believe but rather fodder for discussion. Many students feel uncomfortable contradicting a professor unless they are specifically instructed that it is the correct thing to do.

Small group discussions were not as effective as class discussions in some cases, which may be caused by our small class size. Larger classes may find the small group forum effective. Small groups worked best for preparing students to take specific roles or argue certain opinions within the class discussion. This is also an easy way to start a discussion when students are reluctant to talk. Assigning a small group of students a role or opinion also forces a student to think and argue from a new perspective.

In conclusion, the Broader Issues component takes little time to prepare, carry out, and grade. It has proven effective in exposing students to a wider perspective on computing, getting them to practice their oral and written communication skills, and sparking their interest in computer science. With only a small change in our existing classes, we hope to make a great change in the image of computer science.

5. REFERENCES

[1] Princeton scientists create vote-stealing program. http://www.truthout.org/article/princeton-scientists-create-vote-stealing-program-diebold-accuvote-ts, Sept. 2006.

[2] Facebook's news feed knows what you did last summer. http://www.insidefacebook.com/2007/10/29/facebook's-news-feed-knows-what-you-did-last-summer/, Oct. 2007.

[3] Little laptop a hit in rural Peru. http://www.cnn.com/2007/WORLD/americas/12/24/laptop.village.ap/, Dec. 2007.

[4] Bachelor's and associate's degrees awarded in computer sciences, by sex, 1985–2005. http://www.nsf.gov/statistics/wmpd/figc-2.htm, 2008.

[5] Nerd girls: Breaking the stigmas and stereotypes of women in engineering. http://www.nerdgirls.org/public/index.html, 2008.

[6] D. Blank and D. Kumar. Patterns of curriculum design. In L. Cassel and R. Reis, editors, *Informatics, Curricula, and Teaching Methods*, pages 77 – 86. Kluwer Academic Publishers, 2003.

[7] R. Boyle and M. Clark. CS++: Content is not enough. In *Proceedings of the Technical Symposium on Computer Science Education (SIGCSE)*, volume 36, pages 422–426, March 2004.

[8] C. Dean. Computer science takes steps to bring women to the fold. http://www.nytimes.com/2007/04/17/science/17comp.html, Apr. 2007.

[9] P. Elbow. *High Stakes and Low Stakes in Assigning and Responding to Writing*. Jossey-Bass, 1997.

[10] S. Graham and C. Latulipe. CS girls rock: sparking interest in computer science and debunking the stereotypes. In *SIGCSE Bulletin*, pages 322–326, 2003.

[11] A. Jepson and T. Perl. Priming the pipeline. *SIGCSE Bulletin*, 34(2):36–39, 2002.

[12] J. A. Langer, C. Confer, and M. Sawyer. Teaching disciplinary thinking in academic coursework. In J. Mangineri and C. Block, editors, *Creating Powerful Thinking in Teachers and Students*, pages 81 – 109. Harcourt Brace, 1994.

[13] K. Maney. Bin Laden's messages could be hiding in plain sight. http://www.usatoday.com/tech/columnist/2001/12/19/maney.htm, Dec. 2001.

[14] D. Pogue. Laptop with a mission widens its audience. http://www.nytimes.com/2007/10/04/technology/circuits/04pogue.html?_r=1&oref=slogin, Oct. 2007.

[15] J. Vesgo. Enrollments and degree production at US CS departments drop further in 2006-07. *Computing Research News*, Mar. 2008.

[16] J. Voelcker. Autonomous vehicles complete DARPA urban challenge. http://www.spectrum.ieee.org/nov07/5717, Nov. 2007.

[17] W. M. Waite, M. H. Jackson, and A. Diwan. The conversational classroom. *SIGCSE Bulletin*, 35(1):127–131, 2003.