Don Berkich

Texas A&M University-Corpus Christi

April 5, 2007

Ethics Online: A Plea for Open Source

Abstract

Should the Humanities embrace Open Source software? In this paper I explain the concept of Open Source Software and argue for its use in the Humanities by presenting a case study of improved functionality in the code used to implement asynchronous and anonymous online moral discussions.

Introduction

In the Spring of 2001 my school surprised larger and better-known institutions like Indiana University, Villanova, and the University of Washington by winning the U.S. National Ethics Bowl Championship.

Unlike other college bowls, the Ethics Bowl is not designed to be a debate so much as a deliberative exploration of the moral dimensions of actual cases and ethical dilemmas by teams of students and judges (Borrego, 2004). Since the cases are published a mere six to eight weeks prior to the Bowl, preparation is intense. Two faculty coaches put together a team of between five and eight students and meet with them two to three times each week for up to three hours at a time. The meetings are intense exercises in moral deliberation. Each case is carefully analyzed by the group. Arguments are witheringly examined in light of student-led research. Moral analogies are drawn, refined, and, gradually, the process of serious moral deliberation leads to the discovery of morally defensible courses of action.

Students invariably begin by basing their arguments about a case on largely emotional reactions to the case, which is no fault of their own. Unreflective moral judgment—which, in hard cases at least, is almost always mistaken—is all they've ever known to apply. Students are apt to be as certain of their moral judgments as they are quick to make them. Yet moral certainty cannot long withstand the atmosphere of rigorous moral deliberation in an Ethics Bowl meeting. Students (and coaches) almost always end up repudiating formerly-held positions in favor of more argumentatively sophisticated and ethically nuanced positions; a sense develops on the team that they are gradually uncovering the moral truth of the matter.

I take it that a large part of the value of moral deliberation in moral education is teaching students the value of reflective moral judgment. The process of moral deliberation so successfully employed in preparation for the Ethics Bowl teaches students the ability to withhold judgment, to *examine* a position without either *endorsing* or *condemning* it, and to humbly search for moral truth instead of declaring it.

The Ethics Bowl program is an extracurricular off-shoot of the University's Professional Ethics course, a first course on applied ethics required of everyone. The course is organized around a lecture + discussion section model. Each lecture enrolls approximately 200 students—there are typically three lecture times each semester—and each discussion section has between 30 and 50 students. The idea is that students attend lecture for instruction on ethical theory which they then apply in discussion sections roughly organized according to their professional disciplines. Thus students from the College of Business will enroll in Business Discussion Sections to examine specific cases and issues in Business Ethics, while students from the College of Nursing will enroll in Bioethics Discussion Sections to examine specific cases and issues in Bioethics.

The success of the Ethics Bowl meetings in teaching students moral deliberation does not, unfortunately, translate to the Professional Ethics course. The problem is scale. Learning the shape of moral deliberation by reasoning about particular ethical dilemmas requires that each student engage in moral deliberation, yet this is only possible if a class is sufficiently small to allow every student the opportunity to contribute to discussion. The number of students in discussion section makes it almost impossible to engage every student in serious moral deliberation. The most discussion section leaders can usually hope to achieve is to summarize and explain key points from lecture to help students better grasp the material. Students rarely have the opportunity to wrestle with complex moral dilemmas and real-world moral issues.

In this paper I explain how we used information technology to solve the problem. In particular, I argue using our example that the Humanities should embrace Open Source Software.

I proceed as follows. I first examine some of the special features of the Ethics Bowl meetings missing in large lecture which make it possible to teach moral deliberation. I then describe how information technology can be used to capture these special features in a much larger scale. Examining alternative solutions, I explain the distinction between Open and Closed Source Software and describe how using Open Source Software gave us crucial flexibility to improve functionality and our pedagogic effectiveness.

Learning Moral Deliberation

What is it about the Ethics Bowl meetings that makes learning moral deliberation so successful? There are at least three answers to this question.

1. The Ethics Bowl meeting has an extremely favorable time per student ratio. There is sufficient time for each student to develop and offer his or her insights and arguments.
2. Students who would otherwise dominate discussion are more easily restrained by the faculty coaches and, importantly, fellow teammates: Ethics Bowl meetings are better at self-regulating.
3. Teammates get to know one another and the faculty coaches very well. They enjoy a greater comfort level with one another, trust one another, and, in general, feel *safer* in discussion. It is not unusual to see teammates become fast friends.

I submit that the third answer is the most important, for it is only when students feel safe from the condemnation or scorn of their peers and the professor that they will freely explore new positions and, most importantly, learn how to stand back from their initial positions to critically assess their own arguments. This is particularly true in applied ethics, where topics are often of a highly sensitive nature--e.g., abortion rights, euthanasia, homosexual rights, etc.

Students tend to feel safer in preparing for the Ethics Bowl precisely because they get to know each other so well. There is safety in an environment of friendship and mutual respect, yet friendship and mutual respect are much more easily fostered in a small group. A class of 200 students poses a formidable obstacle to a student's gaining a sense of safety in the class; some students report experiencing terror at the mere thought of being called-upon in lecture.

To be sure, my point is not to disparage large lectures—they can be invaluable if done well. Yet as a comedian quipped about surveys which show that people fear public speaking more than death, it appears we would rather be in the casket than giving the eulogy. This fear is a significant obstacle for students who are being asked to do the sensitive and challenging work of developing a sophisticated understanding of controversial and often polarizing moral issues.

As long as I focused on the process of gaining a sense of safety by learning *more* about one another, however, I could not see a way to meet the challenge. In the short time we have each semester, there is no practical way to get 200 or more students to the level of familiarity and trust an Ethics Bowl team enjoys. If a sense of safety in discussion is essential to useful discussion, and a sense of safety is gained by having greater knowledge of one another, then the possibility of teaching anything other than a superficial applied ethics course on our scale is called into question.

Yet if it is impractical to have more knowledge of one another, then perhaps having *less* knowledge would help meet the challenge. There is, after all, a sense of safety in anonymity. There is nothing particularly groundbreaking in this insight, as anonymity is already employed by professor and student alike. Thus,

- Professors do not generally require that students sign their course evaluations. Instead, anonymity is used to encourage the student to comment openly and honestly.

- Professors sometimes separate student names from student papers so as to grade the papers blindly--that is, without risk of having their personal attitudes towards the student color the grading process.

- Shy students will often sit to the side or back of their class to avoid being called upon. The larger the class, the more the shy student can cloak herself in anonymity by fading into the crowd.

- Students sometimes leave anonymous notes under the professor's door to make a point about the class or complain about a fellow student.

Anonymity is a useful pedagogic tool. Yet preserving student anonymity is as simple as having them contribute to online discussions under pseudonyms. To maximize student engagement, new topics for discussion would have to be continuously added and the discussions themselves would have to be asynchronous, so that students could reply to one another at any time they chose. To ensure rigor and improve sophistication, student contributions to discussions would have to be scored and commented upon by faculty and teaching assistants. Adding an online component to the course, it seemed, could bring the three features crucial to the ethics bowl team to the class at large.

We have not blazed a new trail by moving ethics discussions online, but almost. Painter-Morland et al (2003) evaluate their use of *synchronous* online discussions to teach business ethics against theories of moral development and conclude that such discussions are effective tools for teaching ethics. Closer to our case, Ellenchild Pinch and Graves (2000) describe and assess the use of asynchronous online discussions to teach bioethics, concluding that

> On-line conferencing through a WebBoard succeeded in stimulating valuable bioethical discussion and debate among class members. Both reticent speakers and distance learners were on equal footing with the more outspoken and on-site attendees. Limitations, especially that of time, were present but not insurmountable. The WebBoard success encourages even further initiation and development of on-line strategies to link students with each other while meeting their educational goals. (pp. 710-711)

To be sure, there are pitfalls. Anonymity can encourage students to engage in abusive behavior (Frankel, 1999), claims that students gain analytical skills at least as well in asynchronous online discussion as in the traditional, in-class venue are open to criticism (Miller, 1999), and, in light of the complexity of moral normative theory, understanding how theory applies to practical situations presents its own special challenges for online discussions (Schrag, 2005). Nevertheless,

> ...it is now widely recognized that active learning is essential to achieve many educational goals. In the case of research ethics, this means that students are engaged in discussion about contrived or real cases, challenging thought questions, relevant fictional or non-fictional selections from print or video, or surveys about topics in research ethics. With appropriate guidelines and monitoring of such discussion, the result can be an active learning experience that will reinforce the learning of information, develop improved skills for ethical reasoning, and foster a positive attitude toward research ethics. (Kalichman, 2005, p. 343)

A Technical Solution

Information technology exists that makes it possible to create an anonymous learning community of the sort suggested above which can be used to augment, though perhaps not replace, an existing applied ethics course. That is, our solution is to supplement the Professional Ethics course by deploying a sophisticated asynchronous web-based discussion board I describe in greater detail below. In this system, students register for an account by entering a nickname and their email address. An account is created and a password is emailed to them. The student's nickname becomes her *nom de plume*--or, for the more argumentative student, her *nom de guerre*--under which she contributes to new discussions on the website. Since no one except site administrators have access to her email address, her anonymity is assured.

In addition to anonymity, the success of an anonymous online deliberative community depends on students' sense of ownership (Sieber, 2005). Thus provisions exist for students to contribute their own essays or news for discussion. Students can initiate their own discussions and keep online journals if they choose. Moreover, the scoring system, whereby individual discussion contributions are scored on a scale of -1 to 5, randomly selects active and high-scoring community members and awards them a fixed number of points to use scoring their peers' contributions as they see fit. The scoring of contributions is, therefore, a function of the entire community, although faculty do the lion's share of scoring and always have the final say on scoring disputes.

A student's scores on discussion contributions are summed to give a total, called 'Karma', which roughly represents the student's overall contribution to the anonymous learning community. Karma is then used as part of the student's grade for the course: It typically counts for 30-50%, with the usual examinations and other assignments making up the rest. Given the relationship between individual discussion contribution scores, Karma, and course grades, the scoring serves to reinforce good behavior and extinguish, or at least minimize, bad behavior. Students who are vicious to one another, for instance, are likely to lose Karma and see disapproving comments from their peers. Students who write clearly, rigorously, and respectfully receive higher scores on their contributions. Since contributions are available for all to see, the higher scoring contributions become a model for others to follow. Our experience suggests that consistent use of discussion contribution scoring can, indeed, be a powerful tool for encouraging good habits of moral analysis, exploration, and deliberation.

Open Source Software

The kind of functionality we required, including anonymity and scoring, was not to be found in the usual web-based course management tools like WebCT and Blackboard. Researching alternatives led us to an alternative to WebCT and Blackboard, Open Source Software.

Since humans cannot usually read the strings of 1's and 0's that make up a computer program, human readable and writable computer programming languages are used to write the program's *source code*. Source code is like a recipe for making, or compiling, the program that actually runs on a computer. Corporations that write programs for profit jealously guard their source code so as not to aid competitors in writing their own software.

The Open Source development model eschews the corporate “Closed Source” development model for a community approach wherein anyone can read, comment upon, and make changes to a program's source code (Raymond, 2001). The resulting Open Source development model is richly academic in the sense that it emphasizes peer review and collaboration. Open Source advocates like to say that buying a piece of closed-source software is like buying a car with the hood welded shut (Stephenson, 1999). They are the shade-tree mechanics who like to pop the hood and make adjustments. Indeed, the ability to make changes to the software turned out to be crucial for us, as I will explain shortly. Better, Open Source software is often, though not always, provided free of charge.

SLASH, the Slashdot Like Automated Storytelling Homepage, is the Open Source software that runs the popular news and discussion website slashdot.org (Chromatic, 2002). SLASH provides a number of tools that make it ideal for creating an anonymous learning community and, indeed, a course portal.

- Continuously Updated Homepage
- Polls with Attached Discussions
- Lecture Notes and Discussion
- Threaded Discussions on Articles and Essays
- Student Essay Publication
- Announcements and Student Questions
- Student Submissions for Discussion
- Student Journals
- Scoring of Discussion Contributions

SLASH and all of the software upon which it depends is Open Source and free for use and modification. To be sure, other software exists for augmenting courses. The aforementioned WebCT and Blackboard are common Closed Source examples, and Moodle is a popular Open Source counterpart to them. Why use SLASH, in particular?

WebCT, Blackboard, and Moodle are course management systems that supply a broad range of functionality for building online courses. Yet because developers attempt to provide all things to all people, individual functions are often lacking in sophistication. SLASH does one thing—anonymous asynchronous discussion boards with scoring—very, very well. Indeed, SLASH's functionality in that regard far outstrips anything the other software can provide.

Installing SLASH

Installing SLASH is non-trivial, in part because SLASH is just a single component of a much larger array of necessary software:

1. The Linux Operating System in a standard web-server configuration.
2. Perl, which is usually installed when Linux installs.
3. The MySQL Database Server with a 'slash' user account for the eventual SLASH site.
4. The Apache Web Server compiled with *mod_perl*.
5. Perl modules required by SLASH, which are provided by CPAN (the Comprehensive Perl Archive Network)
6. SLASH modules and applets supplied in the SLASH package.

Chapter 2 of the freely available manual “Running Weblogs with Slash” (Chromatic, 2002) has a detailed description of the steps and specific commands necessary for steps 2-6, but it assumes familiarity with the Linux Operating System and the Linux command line interface.

That said, I advise those whose sole experience with computers is using a pre-installed copy of Microsoft Word on a desktop computer to seek help in installation. Nearly every campus enjoys an Open Source subculture of faculty, Information Technology staff, and, especially, students who are eager to help. For example, many of the servers we use use the Linux operating system. Our server administrator was more than willing to help install SLASH and provide support.

SLASH, however, installs with a somewhat dreary default color scheme and generic layout. An administrative program called “BackSlash” provides a convenient web-interface for changing the look and feel of the site. The following options are especially useful for customizing the look of the site:

Colors: The Color Editor, which allows one to preview and set different color schemes.

Templates: The Template Editor, which controls the layout and content of the HTML pages generated by SLASH.

Blocks: The Block Editor, which controls the content and order of the Blocks appearing in the right column of the homepage.

Vars: The Variable Editor, which sets a number of site variables.

Topics: The Topics Editor, which allows the site administrator to create, edit, and delete the topics under which individual discussions are organized.

Customizing the look and feel of a new SLASH site is a matter of

1. Settling on a color scheme for the site;
2. Creating or adapting graphics for the site header and topics;
3. Editing the order and content of the blocks; and,
4. Creating and editing the topics for the site.

The default variable settings and default templates are satisfactory for most applications, and BackSlash makes (1)-(4) relatively straightforward: The appearance of a new SLASH site can be changed in an hour or so.

BackSlash also provides web-based facilities for editing and managing a SLASH site by creating new discussions, creating new polls, and reviewing student suggestions for new discussions. I've found “Lectures”, “Announcements”, and “Exam Review” topics especially useful, since, for example, lecture materials and lecture synopses may be posted as discussions with the extremely useful result that students can use the discussion to ask questions or raise criticisms. Discussions are also permanently available, so students can be directed to prior discussions for clarification, which tends to save the instructor quite a bit of time. SLASH thus forms the basis for a comprehensive and interactive course portal. The changes I've described thus far, however, are anticipated modifications which require no changes to the underlying source code. The important changes, ones that add significant functionality, come with being able to tinker with the source code itself.

Modifying SLASH code: An Argument for Open Source

The Closed Source development model used by course-ware like the ubiquitous WebCT has a serious drawback: The user, who generally knows best what she needs, cannot make changes to the basic functionality of the software. Companies welcome suggestions, of course, but one then has to wait until the next upgrade to see whether or how a suggestion has been implemented. Improvements and changes cannot be made each semester according to student and instructor evaluations. The Closed Source development model suffocates experimentation and innovation because it is necessary for commercial reasons to protect the source code by restricting access. It makes customization to meet the specific needs of its users all but impossible.

The Open Source development model permits changes and improvements in functionality on an as-needed/as-wanted basis. SLASH in particular is designed to accommodate further development. A few of the more superficial changes I've made include

- Assigning discussion section leaders to be principal scorers of discussions so as to more fairly distribute the workload of scoring discussion contributions.
- Providing color cues on discussions for principal scorers to visually distinguish between contributions which have been scored and those which have not.
- Manual locking and unlocking of discussions to permit principal scorers to close discussions to further contributions and changes in scoring.

- Reporting student activity and scores to discussion section leaders for purposes of tracking student development and retrieving grades.
- Changing the kinds of comments scorers may make about discussion contributions that get reported back to the student-contributor.

These were all relatively trivial changes to make, although in some cases time was spent staring at Perl scripts to try to understand how SLASH manages its variables and the flow of information.

Over time, however, the need for a less trivial change became apparent. Instructors could score a given contribution and attach a comment to the contribution from a menu of preselected comments. Examples include “Well Argued”, “Deserves Replies”, “Grammar and/or Spelling Errors”, etc. The combination of score and comment provides some feedback to the student, but it gradually became clear that a wider channel of communication from instructor to student was needed. Instructors always have the option of making their own contribution to a discussion, but we quickly discovered that such contributions were too intrusive on the student discussions. Students tended to “shut-down” when they saw instructors posting comments and criticisms, although they were eager to see comments and criticisms from fellow students.

The solution I hit upon was to provide a text field, viewable only to instructors logged-on as such, beneath each contribution. The instructor can enter comments of any sort and length into the text field and, upon scoring the discussion, the instructor's comments are automatically emailed to the author of the contribution. A copy of the instructor's comments are emailed to the instructor for record-keeping purposes. Anonymity is preserved at all times, and the instructor's comments are privately available to the student by email—private, at least, as any email exchange can be. The point is that the instructor's comments do not clutter and stifle discussions. Yet the ability to comment at length about a student's contribution and provide those comments directly to the student has proven to be an extremely useful tool for improving our students' efforts. We can quickly catch problems, extinguish bad habits, and offer guidance and direction on difficult arguments. We can also applaud and encourage students who discover a novel argument or propose an especially defensible solution.

Conclusion

It is important to emphasize that the new functionality I describe above was only possible because I could study, analyze, and modify the SLASH source code. That is, it was only possible to tailor the software to our needs because it was Open Source software. No such modifications could have been made to Closed Source software without going through the software's corporate programmers, and unless there is overwhelming demand for some new function, corporations will not spend the money to customize their software. Given our experience, the argument for Open Source software in the Humanities is clear.

Finally, it bears pointing out that other programs in the Humanities should not find the use and modification of Open Source software daunting. The virtue of Open Source software is that the work of previous programmers is available for viewing to serve as a model for making changes. One need not be an expert programmer to read and understand how a particular function has been implemented. All one requires is a clear understanding of the problem that needs to be solved and an idea of how the technology can be harnessed to solve the problem.

REFERENCES

Borrego, A.M. (2004). Ethics bowls exercise students' moral muscles [Electronic version]. *The Chronicle of Higher Education, 50*(26), A31.

Chromatic, Aker, B., & Krieger, D. (2002). *Running weblogs with slash.* Sebastopol, CA: O'Reilly and Associates, Inc. Retrieved April 4, 2007, from http://safari.oreilly.com/0596001002

Ellenchild Pinch, W.J., & Graves, J.K. (2000). Using web-based discussion as a teaching strategy: Bioethics as an exemplar. *Journal of Advanced Nursing, 32*(3), 704-712.

Frankel, M.S. (1999). Anonymous communications on the internet. *Science and Engineering Ethics, 5*(3), 395-402.

Kalichman, M. (2005). Use and abuse of the internet for teaching research ethics. *Science and Engineering Ethics, 11*(3), 341-345.

Miller, J. (1999). Critical thinking and asynchronous discussion. *Inquiry: Critical Thinking across the Disciplines, 19*(1), 18-27.

Painter-Morland, M., Fontrodona, J., Hoffman, M.W., & Rowe, M. (2003). Conversations across continents: Teaching business ethics online. *Journal of Business Ethics, 48*, 75-88.

Raymond, E.S. (2001). *The cathedral & the bazaar: Musings on linux and open source by an accidental revolutionary.* Sebastopol, CA: O'Reilly and Associates, Inc. Retrieved April 4, 2007, from http://safari.oreilly.com/0596001088

Schrag, B. (2005). Teaching research ethics: Can web-based instruction satisfy appropriate pedagogical objectives? *Science and Engineering Ethics, 11*(3), 347-366.

Sieber, J.E. (2005). Misconceptions and realities about teaching online. *Science and Engineering Ethics, 11*(3), 329-340.

Stephenson, N. (1999). *In the beginning... was the command line.* New York: Avon Books. Retrieved April 4, 2007, from http://www.cryptonomicon.com/beginning.html