Viewing Computing Ethics through an OPEN SOURCE Lens

Andrew A. Anda
Computer Science Department
St. Cloud State University
St. Cloud, MN 56301
aanda@stcloudstate.edu

9 March 2007

Abstract

In the context of teaching an upper-division undergraduate Computing Ethics course, this paper describes how the OPEN SOURCE model may be studied by students, and presented by an instructor, as a unifying methodology for the concomitant exploration of the ten topical units specified and prescribed by the *2001 ACM Computer Science Computing Curricula*.

Andrew A. Anda
Computer Science Department
St. Cloud State University
St. Cloud, MN 56301
aanda@stcloudstate.edu

1 Introduction

In the context of computer software, the OPEN SOURCE model (arguably a *paradigm*) consists of the professional practice whereby source code is published under a specific license that defines a set of intellectual property rights and obligations for the use of that code. The primary right (and benefit) is that one (and often many) may peruse the code free from any legal obligation. Under many common licensing conditions, OPEN SOURCE enables and facilitates collaborative user-generated software development as well. There are two primary distinct definitions of OPEN SOURCE software: FREE [10] and OPEN SOURCE [23]. The differences between them are essentially philosophical and ideological. Actually, one can identify a rich variety of OPEN SOURCE definitions [11, 5]. However, OPEN SOURCE is not synonymous with *public domain*.

The popularity and impact of the OPEN SOURCE software model has been burgeoning. The OPEN SOURCE model, in a broader context, has been propagating broadly and rapidly to disparate disciplines and professions in the arts, sciences, engineering, and business in part to exploit the social networking benefits inherent in the OPEN SOURCE model. Surely, the OPEN SOURCE model will be used pervasively and profoundly in many contexts of students' future lives.

The discipline which the OPEN SOURCE model has affected earliest and most significantly has been computing. Computing educators should therefore integrate an introduction to the OPEN SOURCE model into the computing curriculum. For students in any of the computing disciplines, a preliminary examination of the OPEN SOURCE model in the context of computer software can serve as a gateway to an understanding of the broader applications of the model. An appropriate course in which to present to students the OPEN SOURCE model is the *Computing Ethics* (sometimes labeled *Computing and Society*) course, as this course focuses primarily on issues relating to the bidirectional interaction of computing and society.

The *Final Report of the Joint ACM/IEEE-CS Task Force on Computing Curricula 2001 for Computer Science* [41] provides a ten-unit template for a course [`CS280T`] syllabus for a recommended 40 hours (*minimum*) instructional focus on social and professional issues related to computing. Some currently popular texts for the Computing Ethics course, e.g. M.J. Quinn's *Ethics for the Information Age* [28], are careful to cover comprehensively this ten-unit template.

I will outline how the OPEN SOURCE model may be exploited as a unifying context for the presentation, examination, and discussion of most, if not all, of the ten units in the `CS280T` Computing Ethics course template. Of course, it should be left to an instructor's discretion the degree to which OPEN SOURCE is discussed in the service of each unit, as well as the degree to which OPEN SOURCE is discussed when integrating those units in recapitulation. Examples I provide within each unit are meant to be exemplary. Additional, and perhaps more recent, examples for each unit should be located and disseminated by both the instructor as well as the students.

At St. Cloud State University (SCSU), our Computer Science program expects and requires students taking our Computing Ethics course (CSCI 332) to be at the upper division level having had experience programming in teams/groups. I doubt that the utility, methodologies, and nuances of the OPEN SOURCE software model would be as accessible to the

average lower division computer science student having less collaborative programming experience. Currently, shortly after my Computing Ethics course begins, I assign to the students the task of writing a term paper which discusses the OPEN SOURCE model in the context of the ten units as described below. They are allowed the rest of the course time to research and write the paper.

2 Computing Ethics Unit Outline

Discussion of the OPEN SOURCE model will be more comprehensive in some of the following units than in others. As well, there will be much more OPEN SOURCE model related teaching and research materials available for some of the following units than for others.

2.1 [SP1] History of Computing

In this unit, historical antecedents for OPEN SOURCE both inside and outside of the context of computing are explored. One could argue that the evolution of professions from their guild or guild-like origins, wherein profession specific information and data were considered to be proprietary trade secrets, to the modern professional model, constitutes a migration to a model of behavior which exhibits many of the characteristics of the OPEN SOURCE model. More specifically, science evolved to rely upon *peer review*. As well, mathematics now relies on the peer review of proofs of mathematical assertions.
Depending on one's criteria, one can identify at least two formal origins of the OPEN SOURCE model in computing, both establishing a nomenclature. The first is the establishment of Richard Stallman's *Free Software Foundation* [38], and the second is the OPEN SOURCE *Initiative* [15, 23]. In computing, one can identify numerous precursors and antecedents to the formal origins of the OPEN SOURCE model, most early examples relying on the process of review, publication and comment. In other contexts closer to the contemporary model see [22], [23, pages 50–53, 90], [44, 1 History], and the May 2007 special issue of the *Communications of the ACM (CACM)* on the *History of the ACM* [not yet published]. In my research specialty area of numerical computation (scientific computing) there are many examples of the OPEN SOURCE peer review model going back to its mathematical origins, in both the contexts of individual algorithms and code (e.g. *ACM Transactions on Mathematical Software [TOMS]*) as well as libraries (e.g. EISPACK, BLAS, LINPACK, SLATEC, and more generally, NETLIB). See [14].

2.2 [SP2] Social Context of Computing

OPEN SOURCE has utility only in the context of a social network. Ridley [30] provides an anthropological foundation for explanations of why the OPEN SOURCE model has proven effective. Sunstein [39] discusses current research into the relative effectiveness of a variety of collaborative models in various contexts. In addition to the OPEN SOURCE model, applied to not just software, he examines the relative strengths and weaknesses of *surveys*, *deliberative groups*, *prediction markets*, and *Wikis & blogs*. Sunstein also discusses some hybrids of the above different models. Numerous sources in this paper's bibliography

discuss the social aspect of OPEN SOURCE software model. One could argue that all of the other nine units in this outline are specific foci of this unit. Additionally, one could easily proceed to explore other social contexts of OPEN SOURCE outside the software domain [35] as an increasing number of other domains are experimenting and implementing the OPEN SOURCE model. See, for example, [9, epilogue: pages 483–488], [5], and [39, Chapter 5]
International issues represent one topic emphasized by this unit.

> All countries want the benefits of localized software, but some cannot afford the expensive licenses. With FOSS, this problem is solved. A combination of policies that encourage software localization/translation into different languages and the ready availability of FOSS, presents an ideal means for expanding computer use worldwide. [34]

International issues and higher education have been addressed by MIT with its *OpenCourseWare* [21] initiative. High quality undergraduate and graduate courses at MIT are deliverable to anyone anywhere in the world for free. This program has now spread to a consortium of universities around the world [24].

2.3 [SP3] Methods and Tools of Analysis & [SP10] Philosophical Frameworks

I have combined these two units in this outline because they are so interrelated. [SP3] concerns:

- Making and evaluating ethical arguments
- Identifying and evaluating ethical choices
- Understanding the social context of design
- Identifying assumptions and values

And [SP10] concerns:

- Philosophical frameworks, particularly utilitarianism and deontological theories
- Problems of ethical relativism
- Scientific ethics in historical perspective
- Differences in scientific and philosophical approaches

Quinn [28, Section 4.8: pages 188–198] devotes a section to a discussion of OPEN SOURCE primarily in the context of intellectual property (IP) [SP6]. Quinn critiques OPEN SOURCE and IP in the context of *rights-based/social contract* and *consequentialist/utilitarian* arguments.

Berry [3] explores the use of the OPEN SOURCE model in collaborative ethical analysis in a specific domain (Internet research). Sunstein's research [39] concerning the appropriate contexts for the efficacy of various popular collaborative models could certainly be exploited in the service of collaborative ethical analysis. The crucial distinction is that ethical situations tend to be gray and fuzzy without a demonstrably clear result that appears evident once revealed.

2.4 [SP4] Professional and Ethical Responsibilities

Again, this is a unit wherein more specific topics are more comprehensively covered in other units with respect to the OPEN SOURCE model. However, the instructor and/or the students can certainly proceed through the tenets of various applicable professional *codes of ethics* related to computing to argue for the applicability of the OPEN SOURCE model in each case. E.g., consider that the act of *disclosure* in various contexts often serves to satisfy one or more ethical obligations. The primary attribute of the OPEN SOURCE model *is* disclosure. Additionally, one can discuss the obligation of maintaining professional competence and currency. In reading other's code, maintaining code, and originating code, one is developing, honing, and enhancing one's acumen, knowledge, and abilities. Kasper Edwards [7] analyzes this learning process.
One of the primary topical concerns of this unit is *the role of the professional in public policy*. As the OPEN SOURCE model increasingly mediates and hosts public policy debate and analysis, those having a more sound understanding of and facility with the OPEN SOURCE model will better address their professional obligations in the context of public policy. We have witnessed the profound effect the *blogosphere* has produced, just in the last decade, in the context of public policy debate. See Sunstein [39, pages 180–191] for a discussion of the influence of *blogs* on public policy.

2.5 [SP5] Risks and Liabilities of Computer-Based Systems

Failures of software can be harmful in the contexts of any of the rest of the units of this outline, [SP6] intellectual property, [SP7] privacy & civil liberties, [SP8] computer crime, and [SP9] economic issues in computing.
One of the strongest justifications for the adoption of the OPEN SOURCE software model is the model's effectiveness in assessing and mitigating risk.
[SP5] concerns:

1. Historical examples of software risks (such as the Therac-25 case)
2. Implications of software complexity
3. Risk assessment and management

With a nod to Norbert Wiener, a cybernetic definition of computer-based systems includes people as an integral component of the system, able to influence and be influenced by the hardware and software.

With respect to item 1, instructors and/or their students could proceed to identify how the the use of the OPEN SOURCE model could have prevented or mitigated classic failure cases. Evan & Manion [8] provide a comprehensive and multifaceted presentation of factors that either promote or mitigate risk and its analysis. This text could be used in conjunction with this exercise.

Each successive major and minor programming paradigm has been developed and adopted because of its success, to varying degrees, of handling the burgeoning size of applications and the code necessary to implement them. Advocates of the object oriented programming paradigm (OOP) claim that well-designed OOP code is safer code throughout its development lifetime. One target of OOP code is high functional cohesion, i.e., one function – one task. Because collaborative OPEN SOURCE projects often have individuals or small teams modifying or augmenting relatively narrowly specified subsystems of an OPEN SOURCE software project, there seems to be a natural and organic tendency to maintain a higher degree of functional cohesion. Bauer & Pizka [2] and Crowston & Howison [6] discuss the evolution of OPEN SOURCE software projects functionally and socially and the implications for risk.

Another noted benefit of the OPEN SOURCE software model is that there are *many eyes* examining and testing the software as a whole or in part. Within hours of the OPEN SOURCE release of the Netscape Navigator code, "fixes and enhancements begin pouring in off the net." [15]. However, it should be noted that just because a specific code is available for perusal, there should be no *a priori* expectation that a sufficient number of pairs of eyes will appear in a timely fashion to peruse it, thus leaving the code in a higher state of risk and therefore potentially unsafe for an indeterminate time.

2.6 [SP6] Intellectual Property (IP)

An exploration and examination of OPEN SOURCE *licensing* requires concomitant exploration and examination of the tools and analysis of *intellectual property* (IP) in general. The primary stated motivation for the state to recognize IP rights is to encourage (and hopefully spur) innovation (which should spur economic development and overall welfare). The traditional expectation was that one could (*should?*) profit from the exercise of one's IP rights. This is still possible within even Stallman's definitions of the OPEN SOURCE model. But, the economic *carrot* is no longer sufficient to explain the degree of participation in OPEN SOURCE processes.

Ridley [30] provides some anthropological bases for the seemingly altruistic behavior of OPEN SOURCE model participants. Ridley explains the cooperative drive as an evolutionary one whereby the exchange of favors confers mutual benefit. This explanation explores issues of trust, reputation, gifts, trade, and moral sentiment. (Two chapters are devoted to simulations and optimal solutions to the *prisoner's dilemma* and parallels to our and other animals' behavior)

It is important to emphasize the OPEN SOURCE is **not** *public domain*, *shareware*, or *freeware* [26, pages 159–160]. OPEN SOURCE (or FREE) software licensing should reserve the following rights:

1. freedom to *use* the code

2. freedom to *examine* and change the source code

3. freedom to *distribute* the program and any *changes*

[26, page 161]
Some licenses are close but have further restrictions such as *free for only non-commercial use*. The LaTeXlicense prohibits code modification. A major attribute of some licenses, e.g. the popular GPL, is that the are *sticky*, i.e. they behave *virally* like an infection wherein a segment of code confers the license to code that uses it.
For a listing of some significant free/OPEN SOURCE software (FOSS) licenses, see Wikipedia, the OSI website, or St. Laurent [36]. A variety of licenses with a nifty selection applet are available through Creative Commons [5].
Discontent with the current state of the patent process in the United States has been rising for quite a while. Recently, the US Patent and Trademark Office has decided to use a *Community Patent Review* [27] OPEN SOURCE model peer review process in an attempt to improve its actual and perceived performance. The League for Programming Freedom is an organization that opposes software patents and user interface copyrights. [19] Sunstein [39, page 178] states that IP has become such a problem in the biotech domain that a strong OPEN SOURCE movement is developing [4, 40] in this domain as well.

2.7 [SP7] Privacy and Civil Liberties

The visibility of the code in OPEN SOURCE software helps to ensure that no hidden *trapdoors* should lurk, unknown but to a few, in applications which protect privacy or perform a critical or trusted function, subject to the qualification that there are enough eyes of sufficient quality on the code. Cryptographic and other types of security systems which are not OPEN SOURCE are now considered highly suspect. A famous example of a trapdoor being revealed when a source becomes Open occurred in the context of the *Interbase* server database. The principle of *security through obscurity* (relying on information hiding) is no longer considered viable. [33] There is currently national activism regarding public dissatisfaction with electronic closed-source voting systems. The Electronic Frontier Foundation (EFF.org) has issued the following *white paper* [1] regarding e-voting and OPEN SOURCE advocacy. The EFF is also sponsoring and advocating the use of an OPEN SOURCE Internet anonymizer project, *Tor* [42].

2.8 [SP8] Computer Crime

In the context of *computer crime*, OPEN SOURCE software has proven to be essentially a double edged sword. A *predator-prey* dynamic can be recognized operating on the Internet mediated by OPEN SOURCE and disclosure on both sides involving the *White Hats* (*Hackers*) who detect, publicize, and repair vulnerabilities, versus the malicious *Black Hats* (*Crackers*) and parasitic *Script Kiddies*. Both the White Hats and the Black Hats are creating and inspecting OPEN SOURCE software. Ordinary users will utilize the White Hat products and services for their own protection and safety, whereas Script Kiddies will rush to exploit the malware developed by the Black Hats. A current intense security policy

debate concerns the utility of vulnerability disclosure, and more critically, *immediate* vulnerability disclosure.

2.9 [SP9] Economic Issues in Computing

This is the unit that may interest students the most, as it concerns the apparent conundrum of OPEN SOURCE economics and motivation. The chief challenge for many in trying to grok the OPEN SOURCE model economics. Ridley [30] is again appropriate reading for helping to understand the OPEN SOURCE economy explained by Eric S. Raymond [29, Homesteading the Noosphere] as a *gift culture*. I find it useful to adopt and present the following economic classification of software products:

monopoly value the value you gain not just from having the use of a program but from having it be *unavailable to your competitors*.

market value its value as a *salable commodity*.

use value its economic value as a *tool*. ($> 75\%$ overall)

[16] See also [25, 12, 18]. A more mathematical economic discourse is offered by Johnson [17].

3 Conclusion

In the context of teaching an upper-division undergraduate Computing Ethics course, this paper described how the OPEN SOURCE model may be studied by students, and presented by an instructor, as a unifying methodology for the concomitant exploration of the ten topical units specified and prescribed by the *2001 ACM Computer Science Computing Curricula*.

References

[1] *EFF White Paper: Accessibility and Auditability in Electronic Voting*. http://www.eff.org/Activism/E-voting/e-vote_white_paper_20040517.php%.

[2] BAUER, A., AND PIZKA, M. *The Contribution of Free Software to Software Evolution*, 2003. http://citeseer.ist.psu.edu/bauer03contribution.html.

[3] BERRY, D. M. Internet research: Privacy, ethics and alienation: an open source approach. *Internet Research 14*, 4 (2004), 323–332.

[4] *BiOS, Biological Innovation for Open Society*. http://www.bios.net/daisy/bios/home.html.

[5] *Creative Commons*. http://creativecommons.org/.

[6] CROWSTON, K., AND HOWISON, J. *hierarchy and centralization in Free and Open Source Software team communications, 2005*. *http://citeseer.ist.psu.edu/736094.html*.

[7] EDWARDS, K. Epistemic Communities, Situated Learning and Open Source Software Development*, 2001*. *http://citeseer.ist.psu.edu/edwards01epistemic.html*.

[8] EVAN, W. M., AND MANION, M. Minding the Machines. *Prentice Hall, Upper Saddle River, NJ, USA, 2002*.

[9] FELLER, J., FITZGERALD, B., HISSAM, S. A., AND LAKHANI, K. R.*, Eds.* Perspectives on Free and Open Source Software. *MIT Press, Cambridge, MA, USA, 2005. Foreword By-Michael Cusumano.*

[10] The Free Software Definition. *http://www.fsf.org/licensing/essays/free-sw.html*.

[11] GACEK, C., LAWRIE, T., AND ARIEF, B. the many meanings of Open Source, 2002. http://citeseer.ist.psu.edu/gacek02many.html.

[12] GARZARELLI, G. *Open Source Software and the Economics of Organization*, April 2002. http://opensource.mit.edu/papers/garzarelli.pdf.

[13] GODFREY, M. W., AND TU, Q. Evolution in open source software: A case study. In *ICSM* (2000), pp. 131–142. http://citeseer.ist.psu.edu/300079.html.

[14] HAIGH, T. The evolution of software: A european perspective. *SIAM News 39*, 1 (January 2006). Talk: "Free As In Software: The Forgotten Mathematical Origins of Open Source Software.".

[15] *History of the OSI*. http://www.opensource.org/docs/history.php.

[16] *OSI: Jobs for Hackers: Yes, You Can Eat Open Source*. http://www.opensource.org/advocacy/jobs.php.

[17] JOHNSON, J. P. *Economics of Open Source Software*. http://citeseer.ist.psu.edu/480333.html.

[18] KHALAK, A. *Economic Model for Impact of Open Source Software*. http://citeseer.ist.psu.edu/401671.html.

[19] *League for Programming Freedom*. http://lpf.ai.mit.edu/index.html.

[20] MILLER, K., BOWYER, K., AND GOTERBARN, D. *Open Source Software: Intellectual Challenges to the Status Quo*, 2002. http://www.cstc.org/cgi-bin/show_abstract.pl?number=254.

[21] *MIT OpenCourseWare*. *http://ocw.mit.edu/index.html*.

[22] NEWMAN, N. The Origins and Future of Open Source Software*, 1999*. *http://www.netaction.org/opensrc/future/*.

[23] The Open Source Definition. *http://www.opensource.org/docs/definition.php*.

[24] OpenCourseWare Consortium. http://ocwconsortium.org/.

[25] PARENS, B. *The Emerging Economic Paradigm of Open Source*, February 2005. http://perens.com/Articles/Economic.html.

[26] PAVLICEK, R. C. *Embracing Insanity: Open Source Software Development*. Sams Publishing, Indianapolis, IN, USA, 2000. Foreword By-Robin Miller.

[27] *The Peer to Patent Project - Community Patent Review*. http://dotank.nyls.edu/communitypatent/.

[28] QUINN, M. J. *Ethics for the Information Age (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[29] RAYMOND, E. S. *Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001. http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/.

[30] RIDLEY, M. *The Origins of Virtue*. Viking, Penguin Books, London, 1996.

[31] ROTHFUSS, G. J. *A Framework for Open Source [Master's thesis]*, 2002. http://citeseer.ist.psu.edu/563355.html.

[32] ROY, C. K., AND CORDY, J. R. *Evaluating the Evolution of Small Scale Open Source Software Systems*. 2006, http://citeseer.ist.psu.edu/roy06evaluating.html.

[33] SCHNEIER, B. *Schneier: Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'*, January 2007. http://www.schneier.com/essay-146.html.

[34] SOUPHAVANH, A., AND KAROONBOONYANAN, T. *Free/Open Source Software Localization*. Elsevier, New Delhi, India, 2005. http://en.wikibooks.org/wiki/FOSS_Localization.

[35] *SourceForge.net: What is SourceForge.net?* http://sourceforge.net/docs/about.

[36] ST. LAURENT, A. M. *Understanding Open Source and Free Software Licensing*. O'Reilly Media, Sebastopol, CA, USA, 2004.

[37] ST. LAURENT, A. M. *Open Sources 2.0*. O'Reilly Media, Sebastopol, CA, USA, 2005.

[38] STALLMAN, R. *About the GNU Project*. http://www.gnu.org/gnu/thegnuproject.html.

[39] SUNSTEIN, C. R. *Infotopia: How Many Minds Produce Knowledge*. Oxford University Press, Inc., New York, NY, USA, 2006.

[40] THACKER, E. *Open Source DNA?* http://www.mikro.org/Events/OS/text/Eugene-Thacker_OSDNA.htm.

[41] THE JOINT TASK FORCE ON COMPUTING CURRICULA – IEEE COMPUTER SOCIETY, ACM. *Computing curricula 2001; Computer Science; Final Report*. IEEE Computer Society, Los Alamitos, CA, 1991. http://www.sigcse.org/cc2001/280.html.

[42] *tor: anonymity online. http://tor.eff.org/.*

[43] WEBER, S. The Success of Open Source. *Harvard University Press, Cambridge, MA, USA, 2004.*

[44] WIKIPEDIA. Open source — Wikipedia, The Free Encyclopedia*, 2007. http://en.wikipedia.org/w/index.php?title=Open_source&oldid=113743610.*

[45] WOLF, M. J., BOWYER, K., GOTTERBARN, D., AND MILLER, K. *Open source software: intellectual challenges to the status quo. In* SIGCSE '02: Proceedings of the 33rd SIGCSE technical symposium on Computer science education *(New York, NY, USA, 2002), ACM Press, pp. 317–318. http://doi.acm.org/10.1145/563340.563464.*