A Comparative Ethical Assessment of Free Software Licensing Schemes

S. Chopra and S. Dexter
Department of Computer and Information Science
Brooklyn College of the City University of New York
2900 Bedford Avenue
Brooklyn, NY 11210 USA
Telephone: +1 718 951 5657
Fax: +1 718 951 4842
Email: {schopra,sdexter}@sci.brooklyn.cuny.edu

Abstract

Software is much more than sequences of instructions for a computing machine: it can be an enabler (or disabler) of political imperatives and policies. Hence, it is subject to the same assessment in a normative dimension as other political and social phenomena. The core distinction between free software and its proprietary counterpart is that free software makes available to its user the knowledge and innovation contributed by the creator(s) of the software, in the form of the created source code. From an ethical perspective, one of the most pressing questions raised by this form of collaboration is the question of the rights, and the restrictions on them, that are passed on to users and collaborators by the creators of programs. That is, what freedoms do software users deserve, and how can they best be protected? In this study we analyze free software licensing schemes in order to determine which most effectively protects such freedoms. We conclude that so-called copyleft licensing schemes are the morally superior alternative.

Keywords: free software, open source software, copyleft, copyright, freedom, social implications of technology

INTRODUCTION

Software is much more than sequences of instructions for a computing machine: it can be an enabler (or disabler) of political imperatives and policies. Hence, it is subject to the same assessment in a normative dimension as other political and social phenomena. Recently, the potential for radical change inherent in free software has seized the attention of commentators on business, law, politics, technology, and culture (Stallman 2002; Lessig 2000; Raymond 2001; Weber 2004; Vaidhyanathan 2003; DiBona 1999).

The core distinction between free software and its proprietary counterpart is that free software makes available to its user the knowledge and innovation contributed by the creator(s) of the software, in the form of the created source code. This not only permits but encourages interested programmers to become involved with the continuing development of the software, disseminates knowledge about the inner workings of computing artifacts, and breeds independence in the community of software users. It has also been argued that free software may be the only viable source of software in developing nations, where programming talent is available but prices for proprietary software licenses are prohibitive; that free software is a threat to the corporate status quo;[1] and that it preserves the objectivity of computer science *qua* science (Chopra & Dexter 2005).

From an ethical perspective, one of the most pressing questions raised by this form of collaboration

[1] For online discussions, see: http://www.linuxjournal.com/article/6049 or http://www.bytesforall.org

is the question of the rights, and the restrictions on them, that are passed on to users and collaborators by the creators of programs. That is, what freedoms do software users deserve, and how can they best be protected? These are the questions we set out to answer in this study.

Free and open source software can be classified according to the terms under which it is licensed. Richard Stallman's Free Software Foundation (FSF) provided the first definition of this class of software, the Free Software Definition (FSD)[2]. In the late 1990's, the Open Source Initiative drafted an alternative, the Open Source Definition (OSD)[3]. These definitions provide lists of conditions that software licenses must meet to be classified as `free' or `open source,' respectively. Most licenses meeting one definition meet the other as well, but, as we shall see, the language and ideology of the definitions are radically different.

The Free Software Foundation's ideology is most clearly reflected by the notion of *copyleft*, a particular characteristic of some software licenses. Copyleft software licenses require all modified versions of the software to be released under the *same* licensing terms as the original. Hence, copylefted free software is required to remain free in all future versions. Non-copylefted software is not subject to this requirement; thus non-copylefted free software may become proprietary in future versions. (In what follows, we use the term 'non-copylefted software' to refer to non-copylefted free software.)

Both the FSD and the OSD allow copyleft and non-copyleft licenses[4]. The GNU[5] General Public License (GPL) is an example of a copyleft license that meets the FSD; the Berkeley Standard Distribution (BSD) licenses[6] are non-copyleft licenses. So strong is the association of the GPL with free software that it is commonly assumed that 'free software' is synonymous with 'software released under the GPL'. However, it is possible for free software to be released under a non-copyleft license, as the BSD licenses exemplify.

There are crucial distinctions in the spirit of these definitions. The FSF makes clear that its motivation in creating the FSD is the preservation of a freedom *granted by society* rather than by a particular programmer[7]. The preservation of the freedom of all software users, now and in the future, is a moral and social imperative. This concern for the freedom of future users is reflected by the use of copyleft in the GPL.

The OSD emerges from a desire for technical efficiency (acknowledging the technical strength of free software[8]) and neo-liberal business pragmatism (Berry 2004). Indeed, the stated motivation for the promulgation of the OSD is widespread corporate adoption of the free software development model. The text of the FSD deploys an explicit language of freedom and community, while the terms of the OSD are presented in dialog with explicitly business-oriented pragmatic 'rationales' attached to each clause of the Definition. For example, Clause 6 of the OSD ("No Discrimination Against Fields of Endeavor") reads:

> The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

This statement is accompanied by the following 'rationale':

[2] http://www.fsf.org/licensing/essays/free-sw.html

[3] http://www.opensource.org/docs/definition.php

[4] There is a broad array of licenses that meet the free software definition. See http://www.gnu.org/licenses/license-list.html for a partial list, and http://www.opensource.org/licenses/ for a list of licenses that meet the Open Source Definition. Obviously the two overlap.

[5] GNU is a "recursive acronym" which stands for GNU's Not Unix. It is a project to create an operating system made entirely of free software.

[6] There are two BSD licenses: the original and the revised. In the revised version, a clause requiring mention of the University of California as a contributor has been removed. Sometimes 'BSD-style licenses' is used as a descriptive term for non-copyleft software licenses.

[7] This is brought out in the Preamble to the GPL.

[8] http://www.catb.org/~esr/writings/cathedral-bazaar/ is a good source for readings/critique on this subject.

> The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.

The FSD states the same requirements with this language:

> [Users must have the] freedom to run the program, for any purpose…. The freedom to use a program means the freedom for any kind of person or organization to use it on any kind of computer system, for any kind of…job, and without being required to communicate subsequently with the developer or any other specific entity.

Through its language and strong identification with copyleft licensing, the FSD appears to hold the interests of society paramount, while the OSD, with its concern for constraints on the individual (copyleft is one such constraint, although it is not explicitly identified as such in the OSD), seems to prioritize the interests of the individual licensee. A copyleft license need not use the ideological language of the FSD; it might simply state its requirements in pure legalese. That the GPL does borrow language and ideology from the FSD will be significant in our subsequent analysis.

The difference between the two definitions delineates a classical opposition between community welfare and the protection of individual choice. We find ourselves faced with that oldest of political questions: can society restrict the rights of the individual? If so, when, how, and to what end?

TOWARDS AN ETHICAL ANALYSIS

Non-copyleft licenses allow licensees the choice to restrict the freedom of *others*: while the source code is open to the licensee, he or she is free to make modifications and subsequently distribute the software without source code (thus undermining the right of future users to modify the software). In this way, a licensee can create an ostensible commercial advantage for himself by keeping his modifications to the source code proprietary. A copyleft license such as the GPL restricts *licensees'* freedom in that it forbids anyone to make the software, *or any of its future derivatives*, proprietary. In so doing, it preserves the freedom of future licensees of the source code. As the GPL's Preamble states:

> To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

Richard Stallman, the author of the GPL, claims that it enforces its restrictions to ensure the greatest good – freedom – for all. In so doing, it is willing to restrict the freedom of individual licensees. Conversely, non-copyleft licenses may be viewed as deploying the principle that it is obligatory not to restrict anyone's freedom.

We will examine these licensing schemes and attempt a comparative analysis to determine which provides a morally superior choice. We will identify the FSD with copyleft licenses, regarding the OSD's failure to mention copyleft as an implicit endorsement of non-copyleft licensing. While the OSD and FSD both sanction non-copyleft licenses, the change in language of the OSD gives little reason to use copyleft licenses (the FSD includes a discussion of the advantages of copyleft over non-copyleft). In what follows, we will inquire into how these licensing schemes facilitate the provision of one moral good, freedom.

Freedom

Though Hobbes famously defined freedom as the absence of restriction, political thought going back to Aristotle (and going forward to Heidegger), has been at pains to distinguish freedom from license. Lawrence Crocker (1980), building on Isaiah Berlin's (1969) distinction between positive and negative freedoms, characterizes positive freedom as the presence of alternatives and negative freedom as the absence of restriction. That is, for Crocker, a positive freedom may be framed as

x may choose y' as an alternative to y

while negative freedom is framed as

x is not restricted by z

In Berlin's original formulation, z is a person or some institution. Hobbes, then, identified freedom with negative freedoms (his simpler notion of physical restriction included restraint by non-human actors). Harbingers of Berlin's distinction may be found, too, in Maurice Cranston's (1954) analysis, which argues that in speaking of freedom we must ask "free *to do* what?", and "freedom *from* what?"

Much philosophical debate has centered on whether freedom is properly analyzed purely in terms of positive freedoms alone, or only negative freedoms, or as a combination of the two (in order to determine how best to protect these freedoms). Crocker and Charles Taylor (1979) have both argued that positive freedom is the only kind of freedom – that talk of negative freedoms is redundant at best. There are critiques aplenty of positive liberty, an example being Benn and Peters' (1959) contention that it overextends the notion of freedom to the point of meaninglessness – if every social good can be described as the freedom from its opposite, then freedom loses its descriptive meaning.

But our task here is not to provide an analysis of freedom. We contend (with Cranston and MacCallum (1967)) that the distinction between providing alternatives and removing restrictions is a slippery one. We can plausibly view the removal of restrictions as the provision of alternatives; the presence of alternatives can be understood as the absence of restrictions. But if freedom is properly analyzed as the absence of restrictions, yet we concentrate only on the alternatives engendered by an action, then we might incorrectly assess the moral worth of that action. To reduce the risk of an overly narrow analysis we treat freedom as consisting in both the provision of alternatives and the removal of restrictions.

We take as a bedrock principle that freedom is a moral good: that it is worth having and, moreover, that society and state must work to ensure the greatest freedom for its members and citizens. Or, as Lansing Pollock's (1981) freedom principle states, "each person must grant to the other persons an equal right to be free." The only justifiable violation of this freedom principle is the restraint of a person whose actions interfere with the liberty of another. This is echoed in Michael Ignatieff's (1998) remark, in his treatment of Berlin, that the primary purpose of a liberal political community is to create the public circumstances in which men and women are left alone "to do what they want, provided that their actions do not interfere with the liberty of others."

THE LICENSING SCHEMES

We argue that though the licensing schemes under consideration present us with a putative ethical dilemma, it can be resolved in favor of copyleft licenses such as the GPL on three grounds: their greater facilitation of freedoms (negative and positive alike), their conscious appeal to a broader community (that is, their responsiveness to the social context of code) and their commitment to ethical foresight (we defer the explication of this concept for later). We argue that the GPL ensures for itself, and facilitates on the part of the licensee, an ethical vision, by using an explicit language of rights, responsibilities, moral injunctions and freedoms. The restrictions it imposes act to ensure the liberties of each citizen; the only alternative it fails to provide is the possibility of restricting others[9].

The GPL is the archetypal copyleft license. Like its progenitor the FSD, it employs an explicit language of responsibilities and freedoms, as in this excerpt from its Preamble:

> The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. When we speak of free software, we are referring to freedom, not Price. Our General Public Licenses are designed to make

[9] Our reading is distinct from the kind of legal analysis that examines the licenses for legal loopholes, possible entrapments, and enforceability. As examples of the kind of analysis we have in mind see (Moglen 2001; Nadan 2002).

> sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

The GPL states its larger objective immediately (to preserve the freedom of all software licensees) and in the next clause of the license makes prescriptions to implement this objective:

> To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it…if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

The moral imperative for the GPL is that every member of society should be free to inspect, use and modify software. To ensure this, the license enjoins, *thou shalt not make code proprietary.* As a corollary, it is clear that, under copyleft, one user's right to free software may not infringe another's right to the same.

Both the FSD and the OSD enable four freedoms: to run a program for any purpose, to inspect and adapt the code for private use, to redistribute (share) code, and to make public improved versions of the code. The language with which these freedoms are expressed, however, differs significantly. For example, the FSD states

> Free software [includes] [t]he freedom to improve the program, and release your improvements to the public, so that the whole community benefits,

whereas OSD-compliant licenses must

> allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

A non-copyleft license could comply with either definition by allowing modifications and derived works while not requiring that the modified source be kept open. A non-copyleft license, then, implicitly grants its users rights that have the potential to impinge on the rights of others. It can demand very little from its licensees while complying with both the FSD and the OSD. The latest version of the BSD license, for example, only requires that derived works carry the original copyright notice:

> Redistributions of source code must retain the […] copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the […] copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Similarly, the MIT license only requires that the original author's name be kept on the credit list of all derived works.

Prima facie, these licenses read like models of permissiveness with respect to the requirements of the GPL. But of course, the derived works *need not be distributed* with the modified source code. A non-copyleft licensor then, is committed to releasing the source code for a program, but is not committed to future versions of that program being free any more.

A DILEMMA AND ITS RESOLUTION

Non-copyleft OSD-compliant licenses do not significantly constrain their licensees, avoiding any mention of 'freedom' or restrictions on future usage in their text; the GPL, claiming to be about freedom, places more restrictions on code and its uses. Consider an analogy: A blacksmith sells some nails to a carpenter. Typically, the blacksmith implicitly grants the unrestricted right to use the nails in any way the carpenter sees fit, just as non-copyleft licenses may allow licensees nearly unrestricted use of the licensed code. But the blacksmith might require the carpenter to sign an agreement that the nails will not be used for the building of prisons for political prisoners. The carpenter's freedom has been restricted: he cannot use those nails for a particular purpose (where this restriction on his freedom has been imposed for the sake of others' freedom). Similarly, copyleft requires that licensees cannot distribute modified works without providing source code.

The ethical question at the heart of this issue now stands starkly before us: what is the moral course of action – to copyleft or not? Is it morally justifiable to restrict the freedom of another in order to ensure that the overall freedom of the community not be compromised?

Restrictions and Alternatives

To answer this question, we first must clarify: the nature of the 'restriction,' the contours of the 'community,' the concept of 'overall freedom' and the extent of its 'compromise'. Towards this clarification, we begin by asking which freedoms (positive and negative) are facilitated by the two licensing schemes. Which alternatives are made available, and which restrictions are made by each scheme?

First, what is the nature of the copyleft restriction? It constrains the actions of its licensees via precise conditions on redistribution and future use of the software. The copyleft clause of the GPL requires of the licensee that all future versions of the program should be released under the terms of the GPL:

> You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

Importantly, these restrictions are contingent. *If* the licensee decides at some point in the future to make modifications *and* to distribute the modified program *then* certain terms must be obeyed. This circumscribes a very narrow set of future actions of the licensee: any public release of modified GPL'd software must be licensed only under the GPL. The GPL's restriction is not paternalistic; it does not restrain an individual for his own good (he already has the source!); it simply restricts his ability to restrain others. The legal strictures of the GPL apply only if the license is accepted. Note that if the programmer wants to modify the program for personal use (or for internal use in a corporation), then he is not obliged to make the source code public.

Copyleft does not restrict the ability of its licensee to earn a living. A programmer using code licensed under the GPL can, for example, contract to provide services or further modifications to the code. Proprietary code is neither sufficient nor necessary to earn a living, but access to the code is necessary (not sufficient) to enable the positive freedoms identified in both the OSD and the FSD. Furthermore, because the programmer is aware of the restrictions placed by the GPL as he first makes use of the code, he is not blindsided by any of its provisions.

Lastly, there is no coercion in copyleft. The programmer is free to study the code carefully, note its workings, to replicate its functionality by writing new code and to distribute it under any conditions he desires (including making his code proprietary). This freedom is never taken away from him. So, the programmer is free to write new code and to distribute it the way he wants. He is not free, however, to distribute the copylefted code in a manner that the original licensor did not want. The licensor's decision to release copylefted code indicates a desire to maintain the freedom of source: she has either written the code from scratch and copylefted it, or she has modified already copylefted code. In both cases, she has accepted copyleft and its accompanying arguments and restrictions.

By way of contrast, consider the non-copyleft MIT license:

> Permission is hereby granted,… to any person… to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies… and to permit persons to whom the Software is furnished to do so, subject to the

> following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Therefore, *if* the licensee decides to distribute source code (a very different contingency – secondary to the decision to distribute modifications), then the only requirement is that a copyright notice of the original be retained and the original permission be included along with the source distribution. This, significantly, means that future modifications need not include source code distribution. After one round of modifications the freedom of the software is no longer protected[10].

If GPL'd software is linked with another piece of software, the entire product must be GPL'd ("This General Public License does not permit incorporating your program into proprietary programs"). Thus the linked code is freed as well. It is as if the blacksmith in our example above were to stipulate that any artifact his nails are used to build cannot be part of any other device (not just jails) that restricts the freedom of citizens. Critics of the GPL view this clause of the license as a burdensome restriction[11]; certainly, copyright holders of non-GPL-licensed code may not be enthusiastic about licensing parts of their code under the GPL as a result of linkages with code released under the GPL. Both the intention and the effect of this restriction, however, are to provide more alternatives to the users of the hybrid product – copyleft is designed not only to protect freedoms but also to spread them. Non-copyleft licenses say nothing about this strategy for increasing the amount of free software.

What compromises do the restrictions of these licensing schemes necessitate? A strong copyleft license like the GPL compromises technical efficiency by requiring every modification, including those that use proprietary software components, to be released under the GPL. This can result in a slower rate of development as programmers may need to reimplement code that previously existed as only free or only proprietary[12]. But this does not constitute a disregard for societal wellbeing in favor of abstract rights: there is strong evidence for the claim that making source code available leads to higher-quality software[13].

Copyleft does not restrict the freedom of users in any way except to ensure that they cannot place any further restrictions on anyone. Non-copyleft licenses only provide alternatives for the immediate licensee (the licensee is free to do what she pleases with the code); their concern does not extend to future licensees, whose alternatives may therefore be reduced.

Free software remains in a public-domain-like setting because it is protected by copyright; copyleft extends these protections. The act of distributing code under copyleft ensures that no one can take the code out of this pseudo-public domain: copyleft works to remove restrictions on all citizens.

The Community

What are the communities with which copyleft and non-copyleft licenses concern themselves? To answer this, we examine the language deployed by the FSD and the OSD. The FSD keeps its broader perspectives at the fore; it explicitly intends to uphold the right of *all* citizens to use, inspect and modify software. The OSD concerns itself with programmers – specifically with the rights of software businesses – and shows little concern for potential future licensees. On closer inspection, it seems that the OSD is not even concerned with the programmers' community at large, for that community includes programmers who would like to keep source code freely available. It is concerned merely with making itself attractive to those who *might* want to keep code closed in the future, i.e., precisely

[10] See http://www.gnu.org/philosophy/x.html for a case-study detailing precisely such an occurrence with software released under the X-Windows license, which is non-copylefted.

[11] Stallman terms this the 'spider-plant' clause

[12] That is, proprietary developers may feel a need to provide their own implementations of code licensed under the GPL so as to avoid their code coming under the GPL; free software developers may need to provide free implementations of previously proprietary code.

[13] This claim needs amplification: for present purposes it is sufficient to point to the long list of free software projects that power most of the Internet, and to those programs which are universally recognized as technical trend-setters, such as GCC and Emacs. See http://www.catb.org/~esr/writings/cathedral-bazaar/ for an extended discussion of the thesis that free software results in better quality code from an engineering perspective. Note, too, that high-quality software contributes more vigorously both to innovation and creative flux within the software industry and also to social and personal wellbeing.

those that feel little resonance with the original technical and moral imperatives of free software. But a moral stance that is concerned with its effects on only a small community distances itself from the weight of its injunctions. By shirking universality, the OSD loses much persuasiveness.

There is a close analogy here with the law. Pronouncements on legal matters that disclaim legal universality risk being rendered meaningless. Consider the notorious Terri Schiavo case. The US Congress drafted a law that named the Schiavo family as its only beneficiaries, surely an unprecedented step in the history of legislation in the US. Similarly, after the controversial elections of 2000, the US Supreme Court stated, in their resolution of the stalemate, that their ruling should not be taken as setting a precedent for future disputes, thus undermining the legal weight of their judgment. Similarly, a moral stance that disdains its implicit universal moral content renders itself incoherent. If the OSD's claims are to protect the rights of a particular group (it takes a moral stance on *their* rights), then it finds itself pronouncing a meaningless morality, one that only applies to a certain group and not others. In contrast, as the FSD states, it is concerned with ensuring a particular set of freedoms for all users.

There is a clear inverse relationship between the size of the community the license intends to protect and the level of societal freedom the license engenders – the smaller the community the license applies to, the less freedom it carves out. Copyleft licenses, privileging the freedom of society, restrict one `freedom' of programmers: to keep code modifications private while simultaneously releasing a modified binary. Non-copyleft licenses, however, are willing to compromise the freedom of the users of the software (and hence society at large) to inspect source code, in order to protect the freedom of one group of programmers to keep their modifications proprietary.

Ethical Foresight

Moral agents are capable of deliberative agency. We ascribe the status of moral agent to individuals capable of thinking through the effects of their actions, directly and indirectly, in the context of some moral principle. Even if not the calculating agent of utilitarian analysis, a moral agent is one capable of understanding the effects of its actions, of realizing the import of an obligation or moral imperative. Refusing to think through the effects of actions, denying one's autonomy, or disregarding moral principles are hallmarks of an amoral agent[14]. To possess *ethical foresight*, then, is to be able to deliberate upon the moral stance implicit in ones actions and pronouncements and to act with full knowledge of the consequences of its actions. To what extent do the licensing schemes we consider demonstrate ethical foresight?

The FSD keeps its ethical principles in clear view at all times[15]. The OSD, however, carefully avoids any explicit ethical statement; it allows morality but does not require it. While both definitions agree that moral behavior includes free distribution of source code, the OSD refuses to look any further than the immediate freedoms granted by compliant licenses.

The strong copyleft provisions and particular language of the GPL point to its grounding in the FSD. As this portion of clause 7 makes clear, the demands of the license trump all other obligations that the user faces:

> If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then… you may not distribute the Program at all…. if a patent license would not permit royalty-free redistribution of the Program…then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

The language of the OSD is much more permissive of non-copyleft licenses:

> The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

[14] Such an assessment is at the heart of the judgment of juveniles and the insane, for example, as innocent of crimes they have committed.

[15] It seems that this is what is referred to as 'ideological tub-thumping' on the OSD website.

Crucially, this does not *require* them to be distributed under the same terms – it “allows” them. In contrast with the FSD, the OSD does not mention the issue of copyleft and offers no argument for any particular compliant licensing scheme. With this weaker language, the OSD looks away from the moral import of its permissiveness. The goal of the OSD is widespread acceptance; in the name of meeting this goal, it disclaims moral responsibility. The disavowal of the language of the FSD by the framers of the OSD indicates a discomfort with explicit ethical statements, as if the realm of technical enterprise would be compromised by such traffic. But a moral agent must exercise such foresight to truly be moral.

The FSD encourages the explicit language of rights, responsibilities and freedoms found in the GPL:

> To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it…. We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

The decision to copyright the software (and not to place it in the public domain) is made precisely to maintain the freedom of future rights of the users.

The GPL's careful legalese, which anticipates – and closes off – various paths through which it might be defeated, is testament to the care it takes to preserve its moral spirit:

> We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The license is uncompromising on the demands it makes on users. Any use of the licensed program automatically indicates acceptance of the license; if the use is not in conformance with the license, it is illegal. The license includes a copyleft clause that automatically licenses future recipients of the redistributed code with the same license.

The GPL works to make its licensees moral agents: it seeks to inculcate ethical foresight in them as well. It reminds the licensees of its original goals:

> All of the FSF's decisions […] are guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

and urges them to propagate the freedom of software using a copyleft license:

> If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software…. if you distribute copies of such a program, …you must give the recipients all the rights that you have.

The connection with positive freedom is clear: the GPL seeks to increase the alternatives available to its present and future licensees not only by provisioning free software but also by educating them about the ethical issues at stake.

CONCLUSION

Free software makes available the knowledge and innovation inscribed in its source code. This dissemination breeds independence in, and facilitates the agency of, the community of software users. The free availability of source code also enables the community-wide critical dialog crucial to the objectivity of computing as a scientific practice. Both the OSD and FSD demonstrate a commitment to the idea of software being free. The remarkable difference in the language of the two definitions, however, and the attendant encouragement on the part of the OSD of non-copyleft licensing, lead us to the questions of freedom we have considered above.

The contrast between the FSD and the OSD, and between copyleft and non-copyleft licensing schemes, reveals that our putative dilemma is resolved through a consideration of the potential loss of liberty that is a consequence of non-copyleft licensing. Non-copyleft licenses would do well to display the ethical sensitivity manifest in the GPL and to confront the undesirable consequences of the freedoms they do permit.

The Open Source Initiative represents a schism in the free software movement: it seeks greater acceptance among corporate developers for the free software development model, but not for its attendant political and ethical message. In disdaining the explicit ideology of the FSD, the OSD is forced implicitly to make facile claims that writing software is just engineering, that free software is not a moral or social imperative. But these ring hollow in our world, where software is deeply implicated in the creation and maintenance of contemporary social and political structures, ranging from electronic voting to public education to an ever-increasing set of economic transactions. That this deliberate narrowing of the free software ideal is tolerable is perhaps all we need to know in our comparative ethical assessment of the free and ‘open source' software.

ACKNOWLEDGEMENTS

We would like to thank David Coady, Richard Stallman and members of the Faculty Fellowship Publications Program at the CUNY Graduate Center (Marcia Esparza, Jordi Getman-Eraso, Janet Johnson, Anru Lee, Costas Panayotakis, Fredrick Wasser, Richard Wilkins and Sharon Zukin) for their comments.

REFERENCES

Benn, S. I. and Peters, R. S. (1959) *The Principles of Political Thought*. Free Press, New York.

Berlin, Isaiah. (1969) *Four Essays on Liberty*. Oxford University Press, Oxford.

Berry, D. M. (2004). The contestation of code: a preliminary investigation into the discourse of the free software and open software movement. *Critical Discourse Studies*, **1**.

Chopra, Samir and Dexter, Scott. Proprietary software and the death of objectivity in computer science. In preparation.

Cranston, Maurice, (1954) *Freedom: A New Analysis*. Longmans, Green, and Co., New York.

Crocker, Lawrence. (1980) *Positive Liberty: An Essay in Normative Political Philosophy*. Martinus Nijhoff Publishers, Boston.

DiBona, Chris et al, Eds. (1999) *Open Sources: Voices from the Open Source Revolution*. O'Reilly & Associates, Sebastopol, California.

Ignatieff, Michael. (1998) *Isaiah Berlin: A Life*. Chatto & Windus, London.

Lessig, Lawrence. (2000) *Code and Other Laws of Cyberspace.* Basic Books, New York.

MacCallum, Gerald. (1967) Negative and positive freedom. *Philosophical Review*, **76**, 312–334.

Moglen, Eben. (2001) Free software matters: enforcing the GPL, I. *LinuxWorld*.

Nadan, Christian H. (2002) Open source licensing: virus or virtue? *Texas Intellectual Property Law Journal*, **10**, 349–377

Raymond, Eric S. (2001) *The Cathedral and the Bazaar*. O'Reilly & Associates, Sebastopol, California.

Stallman, Richard. (2002) *Free Software, Free Society Selected Essays of Richard M. Stallman*. Free Software Foundation, Cambridge.

Taylor, Charles (1979) What's wrong with negative liberty? In *The Idea of Freedom: Essays in Honour of Isaiah Berlin*. Alan Ryan, Ed. Oxford University Press, Oxford.

Vaidhyanathan, Siva (2003) *Copyrights and Copywrongs*. New York University Press, New York.

Weber, Steve. (2004) *The Success of Open Source.* Harvard University Press, Cambridge.

Samir Chopra is an Assistant Professor of Computer and Information Science at Brooklyn College of the City University of New York. He earned a BA in Mathematical Statistics from Delhi University (1984), an MS in Computer Science from the New Jersey Institute of Technology (1990) and a PhD in Philosophy from the City University of New York (2000). He has worked on logics for belief revision and merging; his current research interests include the philosophical foundations of artificial intelligence, the legal theory of artificial agents, and the politics and ethics of technology.

Scott Dexter is an Associate Professor of Computer and Information Science at Brooklyn College of the City University of New York. He earned a BS in Mathematics and Computer Science from Denison University and his MS and PhD in Computer Science and Engineering from the University of Michigan. His research interests include network and multimedia security, distributed computing, computer science pedagogy, and the history, politics, and economics of technology. He has been an invited speaker at meetings in Philadelphia, Toronto, Prague, and the People's Republic of China, and is co-author of an introductory programming textbook.