Ethics Inf Technol (2009) 11:279–286
DOI 10.1007/s10676-009-9207-9

ORIGINAL PAPER

On the meaning of free software

M. J. Wolf · K. W. Miller · F. S. Grodzinsky

Published online: 3 October 2009

Abstract To many who develop and use free software, the GNU General Public License represents an embodiment of the meaning of free software. In this paper we examine the definition and meaning of free software in the context of three events surrounding the GNU General Public License. We use a case involving the GPU software project to establish the importance of Freedom 0 in the meaning of free software. We analyze version 3 of the GNU General Public License and conclude that although a credible case can be made that the added restrictions are consistent with the definition of free software, the case requires subtle arguments. Strong arguments against the added restrictions are less subtle, and may therefore be more convincing to many users and developers. We also analyze the Affero General Public License and conclude that it is inconsistent with the definition of free software.

Keywords Freedom 0 · Free software · General Public License

Abbreviations

AGPL	Affero GPL
DRM	Digital Rights Management
GNU	GNU is Not Unix
FS	Free Software
FSF	Free Software Foundation
GPL	General Public License
GPLv2	GPL version 2
GPLv3	GPL version 3
OSS	Open Source Software

M. J. Wolf (✉)
Bemidji State University, 1500 Birchmont Drive NE, #23,
Bemidji, MN, USA
e-mail: mjwolf@bemidjistate.edu

K. W. Miller
University of Illinois at Springfield, One University Plaza,
Springfield, IL, USA
e-mail: miller.keith@uis.edu

F. S. Grodzinsky
Sacred Heart University, 5151 Park Avenue, Fairfield, CT, USA
e-mail: grodzinskyf@sacredheart.edu

Introduction

Free Software (FS) is a notion that was first introduced by Richard Stallman in 1984 at the beginning of the GNU project. The Free Software Foundation (FSF), founded in 1985, supports the ethic of Free Software, which is embodied in the GNU General Public License (GPL). The FSF defines free software as software licensed so that it supports four freedoms.[1] The first and fourth freedoms are pertinent to our arguments. The first, Freedom 0, reads: "The freedom to run the program, for any purpose (Freedom 0)." The fourth, Freedom 3, reads: "The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits (Freedom 3)." The remaining two freedoms (the freedom to modify the program and the freedom to redistribute copies) complete the definition of FS. This definition was established in 1985, and since then, its interpretation by the FSF has been clarified through the

[1] See the FSF website, http://www.fsf.org and the GNU website http://www.gnu.org/ for information regarding the four freedoms and the GPL. Both accessed January 27, 2009.

GPL, but the fundamental four freedoms have not changed.[2] In this paper we explore how the meaning of FS has taken hold among the community of free software developers.

The Free Software Foundation's website includes the following, often attributed to Stallman: "Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer."[3] Although Stallman links the concept of free software to the philosophical tradition of free speech, a tradition often traced to Mill (1963), we will not analyze the moral basis of the four freedoms. Rather we will analyze their effect on the software development community and the alignment between the four freedoms on the one hand and various versions of the GPL on the other.

Our attention has been drawn to the interaction between the four freedoms and the GPL by recent events. We focus on a particular software project that tried to prohibit military use of the software via the GPL, on version three of the GPL, and on another FS license, the Affero GPL. We argue that the understanding of the meaning of FS is closely tied to the four freedoms, especially Freedom 0, and that software licenses that are inconsistent with that understanding are met with confusion and resistance by the free software developer community.

FS, like other Open Source Software (OSS), does not exist without the community defined by those who develop the software and those who use the software. However, since individuals use many different combinations of FS and OSS in their programs, we include in the FS community anyone who develops or uses any piece of FS. In order to qualify as "official" free software, the software must include a license that is approved by the FSF. For most programmers, this means using one of the GNU GPLs provided by the FSF. Each GPL is a legal document that has the goal of promoting and protecting both FS and the FS community.

Stallman led the development and refinement of the GPL, with version 1, published in 1989, version 2 published in 1991, and version 3 published in 2007. Each of the three versions of the GPL has the explicit purpose of implementing the four freedoms. Stallman has also developed numerous essays that lay out the argument for FS.[4] Through the first 10 years or so, there was little attempt among those in and close to the software development world to challenge Stallman's principles. That began to change in the late 1990s as Larry Wall, John Osterhout, Guido von Rossum and others began releasing their software under licenses that were not strictly compatible with the GPL; their licenses did, however, retain many of the same features, including the open availability of source code that is essential to FS. About that same time, Eric Raymond analyzed OSS as a software development methodology and posited reasons for its success in developing reliable widely-used software.[5] He and others saw FS not so much as an ethic, but as a software development technique that led to high quality software. This perspective diverted attention away from using software to promote the ideals of the FSF, and refocused it on a community-based software development methodology.

Since the emergence of OSS, there has been a tension between those who advocate for FS because of its ideals and those who advocate for OSS because of the quality of software that is developed and the business opportunities it affords. Chopra and Dexter are proponents of the former perspective and develop a rich argument that for societies that value all types of freedom essential in a democracy, FS has essential features that OSS lacks.[6] In this paper, we will focus on this community tension from a different perspective. We will examine how GPLv3 differs from GPLv2, how it restricts developers with respect to Digital Rights Management (DRM), and how these restrictions are perceived to dilute Freedom 0. GPLv2 was the first widely known legal embodiment of FS and came to be a de facto standard. Fifteen years after GPLv2's introduction, GPLv3 was drafted and after a long, community-based debate, the FSF finalized it. Much of this debate centered on restrictions in GPLv3 that were not part of GPLv2. We will explain how the asymmetry between restrictions against DRM on the one hand and other possible restrictions (such as military use, predatory websites, and computer viruses) on the other hand, is a particularly difficult issue for proponents of GPLv3 to explain to the FS community.

Although arguments that the added restrictions are inconsistent with the four freedoms may be convincing to many users and developers, we conclude that a credible but subtle case can be made that the added restrictions *are* consistent with the definition of FS as expressed by the four freedoms. These restrictions, as we will demonstrate, are based on software *distribution*, rather than software *use*.

In the next section we chronicle a software project that tried to license its software under a GPL-like license that prohibited running the software for military purposes. In this case, the FSF, made the case that Freedom 0 really means "for any purpose," and therefore the GPL could not be used to restrict the software. In the subsequent section, we analyze GPLv3. Here we note the tension set up by Freedom 0 and Freedom 3 and how GPLv3 supports both

[2] GNU Project. The free software definition. http://www.gnu.org/philosophy/free-sw.html. Accessed April 26, 2009.

[3] Ibid.

[4] Stallman (2002).

[5] Raymond (2001).

[6] Chopra and Dexter (2008).

freedoms, even though it may be inconsistent with the understanding held by some in the FS community. Then we consider the Affero GPL, version 3 (AGPL).[7] In this license we find exceptions to Freedom 0, and thus, the meaning of FS.

GPLv2 and Freedom 0: the GPU case

Freedom 0 appears straightforward and wide-ranging. The phrase "for any purpose" seems to leave little room for misinterpretation, and the FSF website includes several explanations that reinforce its broad scope. For example, in a list of questions and answers about GPL,[8] the following appears:

> I'd like to license my code under the GPL, but I'd also like to make it clear that it can't be used for military and/or commercial uses. Can I do this?
>
> No, because those two goals contradict each other. The GNU GPL is designed specifically to prevent the addition of further restrictions. GPLv3 allows a very limited set of them, in section 7, but any other added restriction can be removed by the user.

We invite the reader to note two things about this question and answer. First, the idea of changing a GPL license to block military uses was not merely hypothetical. There was a fairly well-publicized case involving a proposed military use restriction, the GPU case. We will briefly describe this case in this section. Second, note that the FSF indicates that GPLv3 includes a set of restrictions. These restrictions were not in the earlier GPL versions and therefore do not apply to the GPU case. We will discuss the new restrictions contained in GPLv3 in the next section of the paper.

GPU is a software project whose authors originally licensed their software with GPLv2. The GPU project was given web-hosting resources by SourceForge.net, an organization dedicated to hosting free and open source software development projects.[9,10]

On August 31, 2005, the GPU developers published a "status report," on their project that included these statements[11]:

> GPL for no military use
> Following inquires of Nick Johnson (npj), we decided to create our own version of the GPL. The text can be read here http://gpu.sourceforge.net/GPL_license_modified.txt. Goal of the modification is to prohibit military use, without giving away the advantages provided by GPL.

Almost a year later on August 14, 2006, *NewsForge* published an article about GPU's no military use modification of the GPL.[12] Following the article, there were many comments in blogs and mailing lists about the pros and cons of such a restriction being added to a GPL license. That same day, GPU posted the following (spelling and grammar retained from the original).[13]

> Meanwhile, we have been written be members of the Free Software Foundation, asking us to reconsider the change or at least not violate their copyright by removing the preamble and altering the name. We are aware modifying the GPL is not allowed by the GPL license itself, but did it without bad intentions. We go consider what is appropriate. After all, we're not after a legal conflict with the FSF. Give us some time for internal debate, we'll keep you informed.

On August 19, 2006, the GPU developers posted a retraction of their non-military use clause. They reverted to the unmodified GPLv2 as a license for what appear to be largely practical reasons. They needed to use SourceForge as a host, and unless they were compliant with an appropriate license, they would likely lose that privilege. The GPU developers saw reverting to GPLv2 as the easiest way out of the potential legal conflict with the FSF.[14]

The military restriction introduced by GPU was, as demonstrated in the question/answer above, contrary to Freedom 0 and was not allowed by GPLv2. (It was also not a proper open source license according to the OSI.[15]) Some other free software advocates expressed sympathy with the intent of the GPU developers, but insisted that in order to be faithful to Freedom 0, military uses (and other downstream uses that some might object to) could not be singled out in a GPL license. This situation offers evidence that Freedom 0 is a central tenant that lends meaning to FS.

[7] http://www.gnu.org/licenses/agpl.html, accessed December 7, 2007.

[8] http://www.gnu.org/licenses/gpl-faq.html, accessed December 6, 2007.

[9] See http://gpu.sourceforge.net for a description of the project. Accessed December 6, 2007.

[10] For a more detailed discussion of the case see Miller (2007).

[11] http://sourceforge.net/forum/forum.php?forum_id=492617, accessed December 7, 2007.

[12] Tina Gasperson. Open source project adds "no military use" clause to GPL. August 14, 2006. http://www.linux.com/articles/56426, accessed December 7, 2007.

[13] http://sourceforge.net/forum/forum.php?forum_id=601861, accessed December 7, 2007.

[14] http://sourceforge.net/forum/forum.php?forum_id=603503, accessed December 10, 2007.

[15] See http://opensource.org for more information on open source licenses, accessed December 6, 2007.

Representatives of the FSF and the OSI agreed that the no military use clause was unacceptable in any software license that was labeled "free," "open source" or GPL-compatible. We emphasize this because the latest version of the GPL, GPLv3, adds restrictions that for many may seem to violate, or at the very least constrain, Freedom 0.

GPLv3 introduces restrictions not in GPLv2

During its drafting and after its release, GPLv3 was controversial within the FS community.[16] Some of this controversy was about the complexity of GPLv3 when compared to GPLv2. GPLv2 had fewer than 3,000 words; GPLv3 has 5,680 words. Some of the dense language of GPLv3 was devised with the specific purpose of using legal restrictions to discourage the use of GPLv3-licensed software in systems that implement DRM.[17] This language was controversial due to its apparent restriction on the use of FS in DRM systems. As illustrated in the question and answer from the FS website that we quoted earlier, there is no controversy about whether GPLv3 has restrictions. The controversy is about the ethics of those restrictions.

The FSF's website makes it clear that the FSF intends to fight software patents and DRM. Software patents are a major issue for the FSF[18] and have been discussed extensively elsewhere.[19] We focus on DRM, a combination of hardware, software, and laws that collectively attempts to automatically enforce content owners' patent and copyright privileges by making it impossible for buyers to do certain things, even legal things, with products they buy.[20] DRM has numerous critics including the FSF and Richard Stallman, which are both on record opposing DRM due to the fact that it is a threat to FS. GPLv3 was specifically designed to make it difficult to use GPLv3-licensed software in software distributed as part of a DRM system.

The dense language of GPLv3 may confuse some, but the FSF website includes passages that attempt to clarify the purposes and goals of GPLv3[21]:

> In order to honor Freedom 0, your freedom to run the program as you wish, a free software license may not contain "use restrictions" that would restrict what you can do with it.
>
> Contrary to what some have said, the GPLv3 draft has no use restrictions, and the final version won't either.
>
> GPLv3 will prohibit certain distribution practices which restrict users' freedom to modify the code. We hope this policy will thwart the ways some companies wish to "use" free software – namely, distributing it to you while controlling what you can do with it. This policy is not a "use restriction": it doesn't restrict how they, or you, can run the program; it doesn't restrict what they, or you, can make the program do. Rather it ensures you, as a user, are as free as they are.

We find a distinction made here useful in our analysis: GPLv3 is explicitly attempting to restrict how software is *distributed*, not how it is *used*. Indeed, Freedom 3 grants the freedom to *improve* the program and release the improvements so that the whole community *benefits*. By arguing that DRM software does not improve the program or benefit the community, the distribution restrictions are consistent with the four freedoms. Thus, if someone writing DRM software wants to use GPLv3-license software to develop in-house software for DRM, they are free to do so; they can "use" GPLv3 software in this way. However, the resulting DRM software, which includes software licensed with GPLv3 may not be *distributed*, because that DRM software would itself not allow Freedom 0 for people who received the DRM software.

This apparently subtle distinction between unrestricted *use* and restricted *distribution* is, on close inspection, consistent with positions taken by Stallman and FS advocates in other debates. For example, Stallman has stated that users can incorporate GPL-licensed software in closely held internal software without consequence. However, that same software is restricted by any of the GPL licenses from distribution unless it is licensed by the GPL. Thus, should Google's internal search software incorporate GPL-licensed software, unless Google distributes that software to others (unlikely, since it contains trade secrets), then, as Stallman has stated, there is no violation of the GPL.

If one accepts this view of the distinction between unrestricted use and restricted distribution, GPLv3 is consistent with the two earlier versions of GPL. All the GPL versions (and indeed all software licenses that we are aware of) include restrictions on distribution. In the case of earlier versions of the GPL, it seemed clear that the restrictions were focused on disallowing certain methods of distribution and encouraging others. However, GPLv2 did not

[16] For example, see http://blogs.cnet.com/8301-13512_1-9738466-23.html?tag=more and http://lucky13.blogsavy.com/category/fsf-sucks/, both accessed December 10, 2007.

[17] As an example of DRM, the iTunes Store originally encrypted music files to prevent customers from sharing files. They have since changed their policy. See http://www.apple.com/pr/library/2009/01/06itunes.html, accessed September 1, 2009.

[18] http://www.fsf.org/news/oasis.html, accessed December 3, 2007.

[19] See for example, Kevin Panko, An analysis of software patents, http://www.cs.rpi.edu/courses/fall00/ethics/papers/pankok.html, accessed December 6, 2007.

[20] http://www.cippic.ca/digital-rights-management/, accessed December 6, 2007.

[21] http://www.fsf.org/news/gplv3-clarification/, accessed December 6, 2007.

seem to cause confusion about the distinction between use and distribution among developers and users. GPLv2 said, in effect, that you must distribute the software in a manner consistent with the way you received it. Particular technologies (like DRM) were not explicit issues when GPLv2 was developed. As new technologies threaten the FS community, Freedom 3 takes on a more prominent role in the meaning of FS.

Many readers of GPLv3 have found this distinction between distribution restrictions and use restrictions for DRM and the like too fine of a distinction.[22] They seem to be questioning whether software licensed with GPLv3 is indeed free software because of the apparent shift in the meaning of Freedom 0. GPLv3 is written to discourage a developer from using GPLv3-licensed code to implement DRM. For almost any other use, the intended use is irrelevant. The fact that the purpose of DRM software has to do with the *distribution* of software does, according to critics of version 3, remove that purpose from the set indicated by Freedom 0's "for any purpose." The elegant simplicity of Freedom 0 seems, to some, at odds with the singling out of DRM for restrictions. Critics may sense GPLv3 is shifting the definition of FS away from the four freedoms, when it is really shifting the emphasis among the freedoms in response to changing technology.

GPLv3 advocates rightly point out that Freedom 0 would be rendered meaningless if restrictions against military uses were allowed. But critics of GPLv3 claim that restrictions that single out DRM are just as damaging to the spirit of Freedom 0. They believe that protecting FS from threats to its widespread adoption should trump this apparent shift in the meaning of Freedom 0 and thus, FS.

The GNU Affero General Public License

The issue of license restrictions that attempt to promote FS by restricting distribution of software is replayed in the AGPL, a license designed to add a distribution *requirement* (rather than restriction) beyond those required by GPLv3 to developers who modify and run AGPL-licensed software. This requirement applies even when the modifier of the software does not distribute the software. Since this modification is a requirement, it oversteps the bounds of Freedom 3 as is demonstrated in the scenarios developed below.

The AGPL is largely the same license as GPLv3. The most significant departure comes in section 13 of the AGPL. This section addresses the situation where a user of software is interacting with it through a network (software as a service). GPLv3-licensed software permits someone to modify it, and then give others access to that software via a web-based interface. When the software runs on a server owned by the modifier, the modified software is never distributed to anyone. Thus, the developer is not obligated by the license to release the modified source code. However, when the original software is licensed with the AGPL, the developer of the modifications is obligated to release the modified source code.

This sets up an interesting distinction. Imagine software developed for converting digital photographs to printed photographs that is based on modified AGPL-licensed software. If the developer creates a web-based interface through which the user interacts with the software and has the prints mailed to the user, the developer is obligated by the AGPL to release the modified source code. Now consider the developer who creates a standalone machine that produces the prints on site using exactly the same software. This machine can use the AGPL-licensed software without restriction or requirement because the software is never distributed externally, even through the Internet. A third possibility: the developer has the machine installed in drug stores. If the developer sells the machine to the drug stores, the developer must release the source code to satisfy the terms of AGPL (and GPLv3 for that matter). However, if the machines are leased, the situation becomes murky. The question becomes one of "Did the developer distribute the software to the lessee?" As a final scenario, consider the situation where the developer installs its own machines in space it rents from a drug store. It seems in this case, since control of the machine did not transfer, the developer is under no obligation to release the source code.

On what ethical grounds can one justify the distinct requirements regarding the release of the source code for these different scenarios? The scenarios are similar in that the same stakeholders are involved and the same duties to the FS ethic apply. Yet small details of the distribution system lead to dramatic differences in whether the GPLv3 and AGPL licenses allow or prohibit the use of a particular software base. The AGPL changes Freedom 3 into a burdensome requirement.

GPLv3, FSF and the FS community

If it is the case that the GPL is the legal embodiment of the spirit of FS and the four freedoms, we have three legal documents to consider. Which of GPLv2, GPLv3 and AGPL are consistent with the notion of FS that is held by those who develop and use FS?

Without a community of developers, there would be no free software. As seen in the above cases, the GPL software

[22] Jem Matzen, GPLv3 license marks GNU's decline. http://www.thejemreport.com/mambo/content/view/317, accessed December 9, 2007.

licenses codify the community's ideals. However, not all FS developers are particularly attuned to the philosophies of the FSF. Surveys of FS developers suggest that many of them are ambivalent about struggles of the FSF.[23,24] On the other hand, due at least in part to arguments within the community, the development of GPLv3 included four drafts and took over 18 months. While the initial drafts raised many eyebrows and even some anger with regard to some of its restrictive language, that language was eventually softened.

Compatibility of licenses is important for the FS community to ensure interoperability of software with different licenses. Even though the licenses may support some of the same underlying ideals, sometimes software licensed under different licenses may not be legally brought together in the same software product due to restrictions contained in the licenses.

This compatibility issue defines an important distinction between GPLv2 and GPLv3. GPLv2 and GPLv3 are incompatible. A developer who combines GPLv2-licensed software and GPLv3-licensed software cannot legally distribute the resulting software, unless the version 2 software contains a clause in its license allowing it to be distributed under a later version of the GPL. This incompatibility can be viewed as subtle pressure to move code from version 2 licenses to version 3, ultimately shifting the community's notion of the meaning of FS. Note that much of the Linux kernel does not contain the "or later" clause, and Linus Torvalds, holder of the license to a substantial portion of the kernel code has not been convinced that there is a benefit to moving the kernel to GPLv3. He thinks "GPLv2 is simply the better license."[25] He views the issues addressed by GPLv3 as worries, rather than threats, suggesting that at least one important community member believes GPLv2 better captures the meaning of FS.

A survey by Evans Data Corporation conducted 3 months after the release of GPLv3, indicated that two-thirds of FS developers had no plans on adopting GPLv3 in the next year.[26] John Andrews calls GPLv3 "controversial because it imposes restrictions on what you can do with programs implemented under this license."[27] Others take exception with the statistics presented by Evans. Palamida, an organization that tracks open source licensing on a project basis, showed that in September 2007, of the 6,286 projects licensed "GPLv2 or later," 734 had been converted to GPLv3 (11.6%). By November 2007, that number had grown to 1,234 (19.6%).[28] Thus, developers are slowly accepting the restrictions on distribution as part of the meaning of free software.

An ethical analysis of free software, GPLv3 and AGPL

Freedom 0 is a "meta-restriction" written to constrain the developers of free software from placing conditions on those who use their software. The wording appears unambiguous, declaring that a user can use free software "for any purpose." This broad language has had its critics. For example, Don Gotterbarn objected to the idea that programmers have no responsibility for the uses of their software.[29] Gotterbarn points out that Freedom 0 requires that the developers of free software not discriminate against the use of their software for evil purposes.

In opposition to Gotterbarn, Miller argues that for both practical and theoretical reasons Freedom 0 is appropriate.[30] Practically, it is highly unlikely that a developer can anticipate or even accurately guess how a specific piece of software will be used. Theoretically, so-called downstream uses of any technology are not traditionally thought to be the ethical responsibility of the developers except when the eventual uses are obvious and imminent; these exceptions (such as nuclear weapons development) prove the general rule—unless the downstream uses are obvious, dire, and inevitable, the developer should not be required to anticipate and forbid them. (Could the makers of bricks have anticipated their use in the Nazi ovens?)

Both Gotterbarn and Miller can find support for their positions in the language of GPLv3. On the one hand, GPLv3 anticipates and discourages the use of GPL-licensed software in DRM. In so doing, GPLv3 answers Gotterbarn's call for developers to exercise their power (in authoring and copyrighting software) to oppose projects they judge to be unworthy. On the other hand, except for DRM and software patents, GPLv3 prohibits developers from judging any other uses as unworthy. GPLv3, consistent with earlier GPL versions, does not allow a distributor to discriminate against uses of free software except for DRM and software patents.

Therein lies a central issue of GPLv3's discouragement of DRM: if a developer wants to oppose racism, sexism, military abuses, child molestation, or any other evil by restrictions in a software license, then that developer cannot use any GPL license to enforce that opposition. However, if a developer wishes to use GPLv3, then the

[23] Bonaccorsi and Rossi (2004).

[24] Hertel et al. (2003).

[25] http://lkml.org/lkml/2007/6/10/147, accessed December 5, 2007.

[26] http://biz.yahoo.com/bw/070925/20070925006182.html?.v=1, accessed December 3, 2007.

[27] Ibid.

[28] http://gpl3.palamida.com:8080/index.jsp, accessed December 3, 2007.

[29] Wolf et al. (2001).

[30] Miller (2007).

developer must be willing to include restrictions against DRM and software patents. It should be noted that a developer can still use GPLv2 to avoid applying the added restrictions. And a developer could write a license that *isn't* GPL to restrict other uses. However, to adopt a GPL license, and to get the power of that "seal of approval" from the FSF, a developer must choose one of the GPL licenses, and all of them prohibit restrictions other than those listed explicitly in the GPL.

There are at least two responses to this clear asymmetry. In support of the GPLv3 restrictions, one could argue that restricting DRM is directly relevant to the purpose of promoting and supporting FS, and that DRM detracts from the four freedoms, limiting (or in some cases blocking entirely) their practical implementation. As such, the schemes fundamentally attack the core ideas of free software, and using Freedom 3 to resist these schemes is not only appropriate, but also vital. The other "evils" that might be, but are not, restricted (like military use) are not directly relevant to the four freedoms, and (the argument goes) are therefore not appropriate to restrict in the GPLv3. The FSF website essentially makes this case.[31]

The case against the AGPL as a FS license is even more apparent. If the AGPL embodies the notion of FS as defined by the four freedoms, then there is a clear contradiction. The only purpose of having AGPL in addition to GPLv3 is to require the distribution of source code of any AGPL-licensed software that is run as a service. This is an apparent contradiction to the notion of running software for *any purpose*. It is hard to argue that software licensed this way supports Freedom 0. A justification of special pleading is weak in this case. On the one hand, advocates of AGPL could claim that they are supporting community development by providing a license that requires distribution of the code to people who use the code. On the other hand, as our photo printing service example illustrates, the practical effect of the AGPL is that it restricts some business practices while not impacting others that are seemingly ethically equivalent. The AGPL, rather than being free, places burdens on those who modify AGPL-licensed code and those burdens extend beyond the reach of the four freedoms.

Conclusion

The FSF distinction between "use" and "distribution" has validity in the context of the four freedoms, despite the apparent problems. The added complexity of drawing a line of demarcation between acceptable and unacceptable distribution of software based on how that software is used has made it more difficult to claim that GPLv3 embodies the four freedoms, especially Freedom 0, as accurately as GPLv2 does. DRM is a clear challenge to the FS philosophy and the development and propagation of FS, and the FSF makes it clear why it opposes DRM. But by using GPLv3 to thwart DRM, the FSF has made it more complicated for developers and users to understand the license and adopt it with a clear sense of its meaning. Furthermore, the FSF is in the difficult ethical position of defending its prohibition against others from restricting the distribution of free software based on other criteria (like military use) while at the same time defending restrictions against DRM. The FSF's ethical position is further clouded with its support of the AGPL as an FS license.

Because DRM is a direct threat to the philosophy, distribution, and implementation of free software, the FSF can make a legitimate case for its opposition to DRM through an appeal to Freedom 3. However, by using the GPL (perhaps free software's strongest legal weapon) against DRM, the FSF may have made it more difficult to present a clear and convincing case that the FSF itself consistently supports Freedom 0. The ethical case can be made that the restrictions added in GPLv3 do indeed, in the long run, support Freedom 0 (and the other freedoms), but defending that position requires the FS community to change its understanding of free software to something broader than Freedom 0.

While the GPLv3 appears to articulate a legal position consistent with the four freedoms, the long term consequence of that position may be weaker support for FS and the FSF in the community of developers and users. Since that community is crucial to the future success of FS, this weaker support is an unfortunate outcome, especially for those of us who regard FS as an ethically important segment of the software development landscape.

References

Bonaccorsi, A., & Rossi, C. (2004). Altruistic individuals, selfish firms? The structure of motivation in Open Source Software. *First Monday, 9*.

Chopra, S., & Dexter, S. D. (2008). *Decoding liberation: The promise of free and open source software*. New York: Routledge.

Hertel, G., Neider, S., & Herrmann, S. (2003). Motivation of software developers in Open Source projects: An Internet-based survey of contributors to the Linux kernel. *Research Policy, 32*, 1159–1177.

Mill, J. S. (1869) . *On liberty*. London: Longman, Roberts & Green; Bartleby.com, 1999. www.bartleby.com/130/. Accessed 27 Sept 2009.

[31] http://www.fsf.org/blogs/licensing/gplv3-drm, accessed December 6, 2007.

Miller, K. (2007). Open source software and consequential responsibility: GPU, GPL, and the no military use clause. In P. Boltuc (Ed.), *APA Newsletter of philosophy and computers, 6*(2), 17–22.

Raymond, E. S. (2001). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: O'Reilly & Associated.

Stallman, R. M. (2002). *Free software, free society: Selected essays of Richard M. Stallman*. Boston: GNU Press.

Wolf, M., Bowyer, K., Gotterbarn, D., & Miller, K. (2002). Open source software: Intellectual challenges to the status quo. In *ACM SIGCSE Bulletin* (Vol. 34). New York: ACM.