The Divergent Anarcho-utopian Discourses of the Open Source Software Movement

Dale A. Bradley
Brock University

Abstract: The discourse informing open source programming is important for many reasons, not the least of which is the way in which its ideological positions are translated into practical actions. It is argued that the initial anarcho-utopian move initiated by Richard Stallman's GNU Project and Free Software Foundation is currently being transformed into an organizational utopia in the form of the largely Linux-based open source movement. The utopian impulse evident in open source software development is therefore addressed from the perspective that the promises of liberation that inform its anarchy-inspired politics may be undermined by efforts to integrate its communal programming practices into existing market hegemonies.

Résumé : Le discours sur la programmation libre est important à plusieurs égards, notamment dans la manière dont ses positions idéologiques se transforment en actions concrètes. Cet article soutient que le mouvement anarcho-utopique lancé par Richard Stallman avec son projet GNU et la Fondation pour le logiciel libre se transforme actuellement en une utopie organisationnelle prenant la forme d'un Mouvement du logiciel libre qui se fonde en grande partie sur le système Linux. Cet article adopte la perspective qu'on risque aujourd'hui de perdre de vue l'impulsion utopique qui était évidente dans le développement initial de logiciels libres. On risque en outre d'oublier les promesses de liberté qui sous-tendent les politiques du mouvement originel inspirées par l'anarchie.

Keywords: Electronic culture (Internet-based); Cultural studies; Technology theory; Computer science

The current discourse surrounding the rapid development and deployment of free/libre and open source software (FLOSS)[1] is framed by appeals to, and tensions among, various anarchic forms of organization and is also underwritten by an undeniably utopian impulse. The "openness" of open source software is informed by concerns both practical (freedom from oppressive software production and licensing/copyright schemes) and ideological (the valorization of anarchic organizational forms, communal production, and public property rights). While often characterized as a cohesive movement, the FLOSS community and its products

Dale A. Bradley is an Assistant Professor in Communications, Popular Culture, and Film at Brock University, St. Catharines, ON, L2S 3A1. E-mail: dbradley@brocku.ca.

***Canadian Journal of Communication*, Vol 30 (2005) 585-611**
©2005 Canadian Journal of Communication Corporation

are actually an amalgam of different operating systems and software applications produced by myriad programmers working with different goals in mind. The broadest and most obvious division within this community is between Richard Stallman's Free Software Foundation (FSF)—which seeks to make computing resources widely available to the public via the production and distribution of public domain software—and promoters of the much more business-friendly Open Source Initiative (OSI), which seeks to integrate open source practices into existing market practices and hegemonies. In spite of this basic division, a shared anarcho-utopian tendency underwrites discourses on both sides of the FLOSS divide.

What I want to trace here is a general trajectory from ideological position to practical action as it relates to differing forms of utopianism and anarchy. The central argument will be that the initial utopian anarcho-socialist challenge initiated by Stallman's GNU[2] Project and FSF exhibits a subtle but significantly different admixture of anarcho-utopianism than that of the OSI. This difference occurs more at the level of discourse than of practical action, inasmuch as both the FSF and OSI belong to, and provide an important degree of guidance for, the broader FLOSS movement. Producing freely licensed programming code and routing around the user restrictions imposed by proprietary software giants such as Microsoft remains the common goal of the FLOSS community, regardless of its internal divisions. I will therefore use "FLOSS" to characterize both the open source movement and its programming methodologies in their entirety. One should remain cognizant, however, of the fact that the FLOSS community is not homogeneous. It includes not only the FSF and OSI, but also a number of other organizations. That said, the purpose of the current project is to tease out and address the anarcho-utopian tendencies present in the FLOSS discourse while simultaneously foregrounding the similarities and differences between two of its vanguard organizations (the FSF and OSI). A certain degree of generalization is therefore necessary in order to distinguish the FLOSS movement from the traditional proprietary software industry. Due to the detail required to address the anarcho-utopian differences between the FSF and OSI, I will only specifically refer to these organizations where they diverge in terms of their discourse on, or deployment of, FLOSS principles and goals.

I have chosen the FSF and OSI as exemplars of FLOSS for two simple reasons. Firstly, they are two of the most significant organizations with regard to FLOSS. The OSI has steadily built a reputation since its 1997 inception as a primary advocacy group for FLOSS, and it has made significant inroads into corporate computing via its promotion of FLOSS as a viable alternative to proprietary software. While not the first instance of a community of open source programmers, the FSF has been frequently acknowledged to be the first such community to actively take FLOSS principles into the mainstream of computing (Raymond, 1999a; Williams, 2002). In addition, the software licence put forward by the FSF (the "General Public License," or "GPL") and its variants has become the very heart of the FLOSS movement, both technically (in terms of software distri-

bution) and ideologically (as the first clear statement of user rights as understood and promoted by the FLOSS community). Secondly, the founders of the FSF (Richard Stallman) and OSI (Eric Raymond) have both taken it upon themselves to play an evangelical role with regard to FLOSS. Stallman is resolutely against proprietary software and tends to advocate a more socialist version of FLOSS, whereas Raymond is more accommodating of business interests and promotes an approach that is more libertarian.

As I hope to demonstrate, the differences and similarities between the anarcho-utopian discourses and practices exhibited by Stallman's FSF and Raymond's OSI are a matter of degree, rather than kind. Stallman tends to be more concerned with achieving the utopian goal of distributing software unfettered by proprietary licensing regimes. He is somewhat less concerned with giving the control of software development to an anarchically organized community of programmers for fear that his utopian social goals will become subservient to the technical aspects of programming. Raymond, on the other hand, tends to view Stallman's utopianism as laudable but ultimately counterproductive when it comes to actually deploying FLOSS in existing corporately dominated software markets. Raymond thus tends to foreground the role of anarchic organization in the development and distribution of technically superior and less expensive FLOSS projects and applications, while playing down (but not ignoring) the utopian elements of FLOSS discourse and practice. But it is not simply that Raymond provides the anarchy to Stallman's utopia in terms of the anarcho-utopianism of FLOSS. Rather, the particular versions of anarchy and utopia that each implicitly promotes in his work—both written and computational—is what proves to be significant. Stallman's social anarchism leads him to advocate for freely distributed software as a common good for society. His utopianism is thus framed by an appeal to realize the ethical and moral dimensions of software development and distribution in order to produce a better world. Raymond's libertarian anarchism is exhibited in a tendency to pursue FLOSS as a means by which to free programmers and users from the artificial constraints of copyright regimes in order to enable a freer environment for the development and distribution of technically superior software.

The FLOSS movement as a whole encompasses the anarcho-utopian approaches and ideals of both the FSF and OSI. Rather than characterize these two organizations as two fundamentally divergent or oppositional approaches, it is perhaps more useful to think of them as occupying two ends of the FLOSS spectrum. The FSF is more committed to provoking changes in software development by way of an appeal to the social use of information technology. The OSI is more concerned with implementing the technical practices that emerge from this activity within everyday computer use. The key, then, to the utopian impulse in the FLOSS movement is thus the manner by which it proposes and seeks to reformulate the social relations of software production as an anarchic organizational form. It is for this reason that I characterize the sociopolitical discourse of the FLOSS movement as "anarcho-utopian."

In order to unpack this complex set of relations I will first provide some background information on open source and the hacker ethic that informs it. Following a brief overview of the roles played by Stallman and Raymond in the formation of the FSF and OSI (respectively), I will undertake a more detailed analysis not only of the divergent anarcho-utopian discourses revealed in the rivalry between the FSF and OSI, but also of their points of connection and co-operation in relation to the wider FLOSS movement. The similarities between Stallman and Raymond are profound and bind them to the same basic cause: they both promote FLOSS, view proprietary software licensing as the enemy, and believe that a better world (of some sort) will result from the widespread adoption of FLOSS development. Where they differ may be subtle, but it is equally profound: should it occur, the ultimate deployment of FLOSS on a grand scale will be rather different depending on which path proves more influential. With Stallman and the FSF, the criteria by which FLOSS projects are developed would likely be more overtly politicized and be held up to measures of social utility or notions of the common good. If Raymond and the OSI prove to be more influential (as they are currently), then the measures and criteria will likely shift from social utility to technical superiority and the cost of development.

While FLOSS is only in its infancy in terms of its practical diffusion, an attempt can nonetheless be made to understand the anarcho-utopian impulse that underwrites so much of the FLOSS discourse. This discourse is significant precisely to the degree that it is neither a convenient addendum to nor *ex post facto* justification for FLOSS. It is a constitutive feature of FLOSS as both an ideological movement *and* a practical methodology. The varieties of anarchy and utopia in FLOSS discourse are many and they are frequently combined into an anarcho-utopianism deemed to be essential to FLOSS development. In addition, it is the anarcho-utopian element of FLOSS that is most frequently cited as a model to be emulated and/or adapted to broader social formations and practices. FLOSS is therefore as much about anarcho-utopianism as it is about programming, because what marks FLOSS as different from traditional software development is not a new *technical* practice—coding languages remain largely unchanged—but a new *social* practice of software production, distribution, and use. The politically inspired discourse that weaves its way through the non-technical discussions surrounding FLOSS is replete with anarcho-utopian incantations. The question is: what kinds of utopia and what forms of anarchy are being offered up by those involved?

The ultimate successes and failures of the emerging FLOSS approach to software production and distribution remain to be seen, but it does seem likely that FLOSS already has, and will continue to have, a significant impact on the software industry. Whether the traditional software industry might contain, leverage, or otherwise adapt to FLOSS, or if the FLOSS movement will realize its hope that it can somehow radically transform the software industry, is an open question (please pardon the pun). The purpose here is neither to predict the future nor to take sides in the debate over the relative merits of FLOSS versus proprietary soft-

ware, but to address the intimations of hope and deprival that can be gleaned from the relationship between utopianism, anarchy, and the sociotechnological practices that inform contemporary software production, distribution, and use.

A selective history of FLOSS

Understanding the anarcho-utopian discourse embedded in the FLOSS movement first requires a brief definition of open source, as well as a discussion of the "hacker ethic." The hacker ethic is the seed from which the anarcho-utopian discourse springs, rather than from any particular political theory or practice of anarchism. While Stallman and Raymond are at times fairly explicit about their socialist and libertarian politics (respectively), neither claims to be a political theorist. The FLOSS movement's anarcho-utopian discourse is not a full-blown, or even necessarily well-thought-out, program so much as a peculiar re-production of two significant strains of anarchist thought (socialist and libertarian) in a seemingly unrelated field (programming). The reason why these notions have surfaced in relation to programming will have to remain the task of another project. It can nevertheless be noted here that the reasons for, the parallels with, and the loose appropriations of, anarchy and utopianism are likely traceable to the fact that there is a significant degree of resonance between late 19th nineteenth- and early 20th twentieth-century anarcho-utopianism and contemporary programming practices (Imhorst, 2005; Klang, 2005). These connections would seem to centre upon issues related to the tensions over communal versus private property, the individual right to use and produce goods, and the broader legal-bureaucratic modes of governance within which these practices are organized and regulated. Just as the social relations of production and communal life, rather than factories *per se*, were the primary focus of anarcho-utopianism at the turn of the (last) century so too are the technical features and merits of FLOSS secondary to the social relations of software production and communal programming.

Open source software

Simply stated, open source (or, in Stallman's parlance, free) software consists of a program for which the source code (a program's actual computer code) is distributed under the auspices of a licence that allows the user to freely use, modify, and re-distribute the code as they see fit, so long as they give credit for the work of others and, in most cases, include a similar licence in any modified re-distributions of the software. This form of licensing is therefore quite different from proprietary software licences, inasmuch as the company that produces and distributes the proprietary software typically retains all rights to the production, modification, and distribution of the source code and simply allows the purchaser to use the program. The purchaser of Microsoft Office, for example, does not own a copy of the software, so much as they lease the right to use the software on the condition that they do not modify or re-distribute it. The major difference between open source and proprietary software in terms of user and property rights thus resides in the conditions of use imposed on the user by the licence, rather than some sort of difference in functionality or programming code. Stated another way, users of

open source software are granted the right to both the *functionality* of the program (what it does) as well as its *methodology* (how it does what it does). Users of proprietary software only have rights to functionality.

Stallman's choice of the term "Free Software" has lead to a popular misunderstanding that "free" refers to cost. It must be noted here that Stallman's use of the term "free" is not intended to mean "free of charge," but "free to use, modify, and distribute" (his frequently cited phrase being "free as in freedom of speech, not free as in beer") (GNU Project Stallman, 1983). Often accused of rejecting copyright and intellectual property en masse, the FLOSS community (Stallman included) has in fact always maintained that a viable for-profit software industry is not only possible, but desirable under the auspices of FLOSS licensing. The issue is not the complete rejection of software property rights, but an approach that protects authorial rights for programmers' work while at the same time allowing others to adapt, improve, and re-distribute programs. The burgeoning GNU/Linux[3] software industry is free to charge for service, support, and additional programs, but it cannot charge for the original source code, which remains open (or free). The commercial market for FLOSS is thus defined primarily by support services, software bundling, and various ancillary products, rather than the core software.

The hacker ethic

It is the hacker ethic, rather than political theory, that mainly informs the practices of many FLOSS programmers. It therefore provides the discursive seed from which utopian and anarchist discourses emerge and fuse together. As such, it is the most direct source of the political dimensions of FLOSS. Based on playfully sophisticated experimentation and knowledge sharing, the hacker ethic reflects utopian notions of social improvement through efforts not only to produce public domain software, but to facilitate wider access to computing and computer-based services. An anarchic element is also evident inasmuch as programmer/hacker communities tend to be culturally constituted as informally networked groups of individuals whose primary allegiance is to programming itself, rather than to any particular organization they may belong to or for which they may work (Abbate, 1999; Himanen, 2001; Levy, 1985; Wark, 2004). Levy's (1985) seminal codification of the hacker ethic is as follows:

- Access to computers—and anything which might teach you something about the way the world works—should be unlimited and total. Always yield to the Hands-on Imperative!
- All information should be free.
- Mistrust authority—promote decentralization.
- Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
- You can create art and beauty on a computer.
- Computers can change your life for the better. (p. 39-49)

Of particular significance to the anarcho-utopian impulse found in the hacker ethic is the explicit call to develop non-proprietary software products ("all information wants to be free"). Rejecting software copyright regimes that severely limit users' ability to modify and re-distribute software entails the recovery and retention of the public domain style licensing practices of early computing. The eventual creation of a new model of software development (FLOSS) exemplifies many other elements of the hacker ethic's credo: decentralization, information-sharing, and experimentation. While its roots are in earlier modalities of software production and distribution prior to the desktop PC explosion of the late 1980s, the FLOSS model that emerged from the GNU Project introduced by Stallman in 1983 was catalyzed by the rapid diffusion of desktop PCs and networking throughout the 1990s. This in turn enabled new forms of communal grass-roots software production and distribution founded on code sharing and project development via the Internet, Bulletin Board Systems, and World Wide Web (WWW). The impetus behind the communal "openness" of Stallman's GNU Project is thus not only a significant break with the copyright regimes of industrialized software production, but a hopeful return to what Stallman views as a kind of golden age of programming.

Richard Stallman, GNU, and the Free Software Foundation

> Why I Must Write GNU
>
> I consider that the golden rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. (Stallman, 1983)

The popularity of FLOSS ideals and practices indicates a certain dissatisfaction with corporate technoculture on the part of some of those who work in these institutions. This dissatisfaction is perhaps most clearly evident in Richard Stallman's 1983 GNU Manifesto, wherein he critiques the institutional shift from public domain to copyrighted software development that has occurred in the last three decades. This shift occurred around 1965-70 as a result of the growth in the mainframe-computing market, which led to the practical realization that software development could be an economically rationalizable for-profit enterprise (Campbell-Kelly, 2003). Once instituted, the proprietary software industry expanded rapidly during the 1980s with the swift diffusion of PC hardware.

Prior to the broad commodification of software, programmers worked in what was then a rather rarefied realm of computing, consisting of mainframe devices ensconced in, and used by, a relatively limited number of organizations operating within large institutions (with universities and the military being the most influential arenas for non-manufacturer computing development). As such, programmers were a rather small and largely homogeneous sociocultural group that actively shared knowledge and code in what seems today to be a remarkably free manner. Without having to worry very much about software licences and pat-

ents, copyrights, institutionally commodified intellectual property, and non-disclosure agreements, programmers enjoyed a fair degree of freedom within their institutional workplaces, and this lead to the collectively produced set of informal practices and tenets that constitute the hacker ethic.

In a somewhat ironic twist, Stallman's greatest innovation (or "hack") may not be any particular computer code, but the licence under which the code is used. Stallman's inventive General Public License (GPL) was created not only in order to facilitate the production of free software, but also to keep it free. Stallman recognized that while public domain software could initially be made freely available, modified versions could subsequently be copyrighted and thereby transformed into a proprietary form. The problem was how to make the free-use provision carry forward through subsequent modifications. He thus devised the GPL for GNU in a manner that kept the software free—while still retaining authorial credit—regardless of the numerous modifications and iterations it might be subject to once released. Playfully termed "copyleft," the GPL is a licensing system that expressly prohibits copyrighting the open source components of any software developed using GNU software protocols and standards and, furthermore, requires that any modified versions of GNU software subsequently produced include the GPL as a condition of distribution. The OSI's recent mobilization around GNU/Linux continues and parallels Stallman's efforts via the OSI's adoption and promotion of the GPL's copyleft provision and its variants. While the hallmarks of Stallman's communal software production system remain, the overall nature of OSI inspired software is nonetheless framed by a rather different, and much more "business-friendly," notion of utopian openness than is evident in Stallman's manifesto musings.

Eric Raymond, GNU/Linux, and the open source initiative

> There is much more to Stallman's Manifesto . . . Suffice it to say that on the surface, it read like a socialist polemic, but I saw something different. I saw a business plan in disguise. (Tiemann, 1999, p. 72)

Stallman's GNU Project was supported by a cadre of hardcore coders for almost a decade by the time Linus Torvalds' 1991 release of GNU/Linux took FLOSS principles into the mainstream. This moment of popular actualization was not simply technical (available hardware, software, programmers, networks, et cetera); but was based on the recognition that communally produced copylefted programming could find a place within the everyday structures of IT industries, services, and markets. The moment that Tiemann saw a business plan in Stallman's "socialist polemic" is, Stallman would argue, the point at which the anarcho-utopian ideals of the FSF were co-opted and transformed into little more than an organizational methodology of software production spearheaded by the OSI.

Stallman's dislike and mistrust of the OSI have tempered over the years, but its roots are both personal and ideological (Scoville, 2001). Eric Raymond's address to the 1997 GNU/Linux Kongress, entitled "The Cathedral and the

Bazaar," compared two styles of FLOSS development and, upon its subsequent inclusion in his book of the same name, it became a bible of sorts for the FLOSS community, as well as a vehicle by which FLOSS ideas and ideals were brought to non-programmers and the media. Raymond (1999a) argues that the "cathedral" style of development is one in which source code is made available, but subsequent releases are vetted and controlled by an exclusive group of programmers. The "bazaar" style releases and develops source code in an anarchic and ongoing ad hoc fashion via the Internet (with incremental changes always visible and publicly available for use). Stallman was offended by Raymond's use of the FSF as an example of the cathedral style, and he was especially annoyed by the implication that the FSF was an elite and centralized group. More importantly, Stallman thought that the promotion of a potentially massive, and certainly for-profit, industry founded on the implementation and support of open source software could easily lead to the defeat of the more socially minded anarcho-utopian principles of FLOSS.

Anarcho-utopianism

The utopian impulse

So much of what matters in FLOSS is related to property and knowledge relations, and this has had the related effect of providing FLOSS with potentially transferable strategies of property, knowledge, labour, and power relations that may allow FLOSS principles to be applied beyond the realm of software production. Recent suggestions that FLOSS methods of software development may serve as a utopian model for more democratic policymaking resonates with political theory in general, and with anarchism in particular:

> The experience of open source development, or even just the acceptance of its value as a model for others, provides a real-life practice for the deeper change in perspective required if we are to move into a more networked and emergent understanding of our world. The local community must be experienced as a place to implement policies, incrementally, that will eventually have an effect on the whole. (Rushkoff, 2003, p. 61).

FLOSS' hacker-inspired anarcho-utopian model of software development here becomes the utopian model for society itself—a society fundamentally underwritten by an emergent form of organization built upon principles of anarchic communities.

FLOSS, however, is not a political platform. It is a modality of software production which finds its production founded in communal forms of decision-making, intellectual labour, and product distribution. This modality is necessarily framed within, and very much constrained by, the broader sociopolitical relations of post-industrial capitalism. While it cannot be said that FLOSS is of the same order as a totalizing social movement, fruitful comparisons may nevertheless be made with William Morris' large-scale utopian socialism:

> [The Open Source Movement] seems almost like a descendent of the Arts & Crafts movement. As its William Morris, Linus Torvalds supervises a global

> workshop in which programmers work together to develop a system that is accessible to all. John Ruskin's 'poet laureate' role is left to science fiction writer Neal Stephenson, whose novels celebrate the artisanal lifestyle of computing programming. In his legendary essay, 'In the beginning was the command line', Stephenson argues for the 'oral tradition' of Linux, which he depicts as an ecosystem of code sustained by [an] anarchic community of hackers (Murray, 2000).

The communities and communes begun or inspired by Morris' utopian ideals tended to be limited in scope and framed largely by a set of concerns related to the exigencies and inequities of the Industrial Revolution. The common thread between Morris' utopian socialism and the FLOSS movement is found in their shared valorization of anarchic communal forms as the foundation for social systems of production.

Morris' *News from Nowhere* (2003, original edition 1890) posits a pastoral twenty-second-century society in which decentralized craft production replaces centralized industrial production and the abolition of money leads to the free distribution of goods produced not only on the basis of necessity, but for the joy of artful production itself. The parallels and potential affinities with FLOSS' hacker ethic are clear. Morris' and FLOSS' utopianism both envision profound changes to property and labour relations, but do so by way of grounding their visions quite firmly within a fairly circumscribed set of practices related to one's personal attachment to joyful and artful activities of production. The tendency toward the personal marks Morris' thought as rather different than the more abstract notions of political economic re-organization emanating from the socialism of Marx and Engels or the anarcho-syndicalism of Goldman, Berkman, or Kropotkin. More significant than the actual form of Morris' utopia, however, is what Levitas (1990) calls its function:

> Broadly, one may divide approaches to utopian studies into two streams. The liberal-humanist tradition [which would include Morris] tends to focus on definitions of form. In contrast a largely, but not exclusively, Marxist tradition has defined utopia in terms of its function—either a negative function of preventing social change or a positive function of facilitating it, either directly or through the process of the 'education of desire.' (p. 6)

The distinction between the form and function of utopian thought and practice is essential to understanding the utopian element of the FLOSS discourse, because it allows one to separate utopian models from the impulses that inform them.

Virtually every utopian statement has been legitimately critiqued in some way or other, and from every political quarter, on the basis of the form of society it proposes. While not the first to do so, Karl Popper's (1963) critique of utopianism famously damns it on the basis of its singular tendency to propose highly managed social systems that are nothing short of totalitarian. Marx and Engels (1967, original edition 1848) find similar faults with utopian models, but reserved a more positive role for the utopian impulse inasmuch as it spurs individuals to critique, and possibly think beyond, the confines of currently existing social systems. For Louis Marin (1984), the value of utopian thought is the way in which the funda-

mental pun that constitutes the term "utopia" (as both outopia ["non-place"] and eutopia ["good place"]) indicates that utopianism is less about erecting a new social order than it is about the neutralization of contemporary social relations by positing an alternative set of relations. As Jameson (1988) notes:

> To understand Utopian discourse in terms of neutralization is . . . to propose to grasp it as a process, an energia, enunciation, productivity, and implicitly or explicitly repudiate that more traditional and conventional view of utopia as sheer representation, as the 'realized' vision of this society or that ideal. (p. 80-81)

To either attack or defend FLOSS as a fully fledged utopian system is to miss the point: what matters in FLOSS is the way in which it reveals contemporary socio-political tensions in both commodity and knowledge production by way of a neutralizing utopian impulse that seeks to posit an alternative to a component of post-industrial capitalism.

The utopian impulse evident in Stallman's 1983 GNU Manifesto addresses a narrowly delimited segment of social practice (software development), but it would not be long before it would migrate to a wider social arena. Computing and networking proliferated between 1983 and 1995, with one result being that issues related to computing came to be of broader concern. In 1983, the decommissioning of ARPANET and creation of the Internet led to the emergence and rapid growth of a what is now known as "cyberspace." Because the U.S. government chose not to regulate the content of the Internet and WWW, myriad discursive communities sprang up, and a remarkably free discursive environment prevailed in the new medium. About a decade later, the U.S. government became concerned about the amount of salacious and possibly illegal material perceived to be circulating through the Internet and introduced the Communications Decency Act (CDA) as a component of the Telecommunications Reform Act in 1995-96. The CDA was eventually struck down by a "netizenry" worried that the CDA's draconian regulatory codes would essentially criminalize much of the online discourse concerning sexuality and gender, as well as quash political discourse that deviated from the status quo.[4]

The battles over the CDA resulted in the restoration of free speech online, but perhaps more importantly, the debate firmly cemented the popular notion that cyberspace was more than simply networked data transfer: it was a public sphere, a venue for community, and a social space in which a new citizenry were struggling to establish social relations, rights, and codes of conduct. While not directly related to FLOSS, the struggle over the CDA was driven in large part by the same anarcho-utopianism evident in the hacker ethic and, by extension, the FLOSS movement. As early adopters and developers of the Internet and WWW, hackers were already using these networks as the communicative and communal infrastructure for their development projects. As such, the reaction to attempts made to regulate the Internet and WWW—especially one as draconian as the CDA—brought out the anarcho-utopian qualities of the hacker ethic in a more obviously political fashion.

The initial passing of the CDA did not achieve its intended goal of bringing cyberspace's "netizenry" into line. Rather, it had the opposite effect of constituting an online polis by providing a "them" (the nation state) that sought to control an anarchic "us" (a stateless cyberspace full of hackers). The day after the CDA was signed by President Clinton, Electronic Frontier Foundation (EFF) founding member John Perry Barlow issued—online, of course—"A Declaration of the Independence of Cyberspace," which included the following:[5]

> We have no elected government, nor are we likely to have one, so I address you with no greater authority than that which liberty itself always speaks. I declare the global social space we are building to be naturally independent of the tyrannies you seek to impose on us. You have no moral right to rule us . . . We believe that from ethics, enlightened self-interest, and the commonweal, our governance will emerge. (1996)

Barlow's declaration evokes one result of the public debate over freedom of speech on the Internet: the characterization of cyberspace as a utopian space. Capturing both the "eutopian" and "outopian," Barlow's declaration severs cyberspace from its embedded relationship with existing sociopolitical structures and declares it to be a virtual (outopic) and independent environment. Similarly, Barlow characterizes cyberspace as an eutopic ("good") place wherein the errors of the past might be avoided. Echoing Marin's notion of neutralization, cyberspace becomes a virtual utopia to the degree that it is used as a figurative device for the neutralization of the notion that the Internet might be made subject to the regulatory regimes of any single governing agency. Barlow also evokes anarchy inasmuch as he commits to an openended and emergent sociopolitical structure rather than a totalizing solution that would systematize and rigidify cyberspace's social structures. Taken together, the anarchic and utopian qualities of the hacker ethic expressed in Barlow's pronouncement fuse so as to constitute a call for a new form of anarcho-utopian organization to be deployed via cyberspace. As Parker (2002) argues, utopian thought and practice tend always to be organizational in nature: "most, if not all, fictional and actual utopias rely on a re-formulation of principles of social order. They are in that sense organized, though often on different principles to the market managerial hegemony" (p. 217-218). Barlow's very public fusion of the anarchic and utopian tendencies inherent in the hacker ethic unfolded in a number of online missives during 1997-99—at just the same time as the FLOSS community began to diverge over issues of how best to pursue their anarcho-utopian goals. This divergence was made most obvious in the growing rivalry between the FSF and OSI, a rivalry that forced both organizations to gradually elucidate their key differences (and similarities) with regard to their anarcho-utopian underpinnings.

The FSF's and OSI's anarcho-utopianism are both "organizational" in Parker's sense, because they both posit a new ordering of software development and distribution practices by which FLOSS projects may be facilitated and diffused. The key difference is that the FSF's utopianism is informed by a type of anarchy that has its roots in socialism and, for Stallman, that means placing the

social ethics of programming code over and above the legal codes of copyright, patents, and intellectual property. This position is evident in Williams's biography of Stallman:

> "[T]he freedom to copy and redistribute noncommercially should remain unabridged at all times," Stallman insists . . . . When I ask whether the courts would accept such a permissive outlook, Stallman cuts me off.
>
> "That's the wrong question," he says. "I mean now you've changed the subject entirely from one of ethics to one of interpreting laws. And those are two totally different questions . . . . How the courts would interpret existing laws is mainly in a harsh way, because that's the way these laws have been bought by publishers."
>
> The comment provides an insight into Stallman's political philosophy: just because the legal system currently backs up business' ability to treat copyright as the software equivalent of land title doesn't mean computer users have to play the game according to those rules. Freedom is an ethical issue, not a legal issue. "I'm looking beyond what the existing laws are to what they should be," Stallman says. "I'm not trying to draft legislation. I'm thinking about what should the law do? I consider the law prohibiting the sharing of copies with your friend the moral equivalent of Jim Crow. It does not deserve respect." (Williams, 2002, p. 71-72)

Reflecting the primary tenets of the hacker ethic, Stallman's moral/ethical approach thus exhibits the potential capacity of utopian thought to neutralize current social conditions in order to posit some other order.

This modality of utopianism is rather different from the OSI's, inasmuch as the latter tends to frame utopian goals in a much narrower fashion. Raymond's fundamental problem with Stallman's approach rests upon the latter's stubborn refusal to try to advance FLOSS development within—never mind in co-operation with—the commercial software market and/or the proprietary software industry. Raymond characterizes Stallman as an inspiring but overly zealous and irrationally anticommercial spokesperson for the cause, who may end up impeding, rather than advancing, the growth of FLOSS development. Raymond's utopianism is tempered by his avowed pragmatism, and this, he argues, allows him to remain dedicated to FLOSS without falling into the trap of the potentially counter-productive zealotry that he sees in Stallman:

> Historically, the most visible and best-organized part of the hacker culture has been both zealous and very anticommercial. The Free Software Foundation founded by Richard M. Stallman (RMS) supported a great deal of opensource development from the early 1980s forward . . . . But the FSF was never the only game in town. There was always a quieter, less confrontational and more market-friendly strain in the hacker culture. The pragmatists were loyal not so much to an ideology as to a group of engineering traditions founded on early opensource efforts that predated the FSF. These traditions included, most importantly, the intertwined technical cultures of Unix and the pre-commercial Internet . . . . To pragmatists the GPL is important as a tool, rather than as an end in itself. Its main value is not as a weapon against hoarding, but as a tool for encouraging software sharing and the growth of bazaar-mode development

> communities. The pragmatist values having good tools and toys more than he [*sic*] dislikes commercialism, and may use high-quality commercial software without ideological discomfort. At the same time, his opensource experience has taught him [*sic*] standards of technical quality that very little closed [i.e., proprietary] software can meet. (Raymond, 1991, p. 69-70)

These two statements from Stallman and Raymond encapsulate both their similarities and differences of purpose. They are both profoundly and personally committed to FLOSS advocacy and development. On this point they do not argue. Where they differ is not only over the means by which to achieve their goals (commercial co-operation or not) but in what constitutes the goal itself. Stallman's moral/ethical approach explicitly values the practice of software development more than its products. Raymond's push for technically superior software foregrounds the product as an expression of the value of the practice.

Williams addresses Stallman and Raymond's shared, yet divergent, commitment to FLOSS by way of addressing Stallman's response to an audience member's question that echoed Raymond's questioning of Stallman's approach *vis-à-vis* technical efficacy:

> [T]he term "open source" [has] political implications. For open source advocates, the term open source serves two purposes. First, it eliminates the confusion associated with the word "free" . . . . Second, it allows companies to examine the free software phenomenon on a technological, rather than ethical, basis . . . . Without a way to win over [business managers, investors and non-hackers], Raymond argues, programmers are doomed to pursue their ideology on the periphery of society:
>
> When RMS [Stallman] insists that we talk about "computer users' rights," he's issuing a dangerously attractive invitation to us [hackers] to repeat old failures. It's one we should reject—not because his principles are wrong, but because that kind of language, applied to software, simply doesn't persuade anybody but us. In fact, it confuses and repels most people outside of our culture. [citing Raymond, 1999b]
>
> . . . When an audience member asks if, in shunning proprietary software, free software proponents lose the ability to keep up with the latest technological advancements, Stallman answers the question in terms of his own personal beliefs. "I think that freedom is more important than mere technical advance," he says. "I would always choose a less advanced free program rather than a more advanced nonfree program, because I won't give up my freedom for something like that. My rule is, if I can't share it with you, I won't take it." (Williams, 2002, p. 115-116)

The question of technical efficacy reveals Stallman's and Raymond's divergent utopian goals and means for achieving them. Stallman remains stubbornly committed to the prioritization of utopian practices over technically superior products. Raymond's libertarian pragmatism leads him away from such entrenched positions and allows him to search for, and perhaps find, some sort of common ground with those who use the products of his ostensible enemies in the proprietary software world.

The reason for Stallman and Raymond's divergence over utopian goals and practices should not, however, be extended so far as to include their utopian impulses. Raymond and Stallman both acknowledge that their visions are actually quite compatible at the broadest levels. As Raymond writes: "the real disagreement between OSI and FSF, the real axis of discord between those who speak of "open source" and "free software", is not over principles. It's over tactics and rhetoric. The open source movement is largely composed not of people who reject RMS's ideals, but rather of people who reject his *rhetoric*" (Linux Today, Raymond, 1999b). The divergence over non-discursive tactics and discursive rhetoric arises out of the way in which their utopian visions are framed and inflected by the particular strains of anarchy that they promote. Stallman's socialist anarcho-syndicalism and Raymond's libertarian individualist anarchism are of far greater significance because their views in this area serve to constitute the organizational methodologies and sociopolitical rationalizations for their differing approaches to achieving their broadly similar utopian goals.

Anarchy

> Unless I am gravely mistaken—as I hope I am—the revolutionary goals of anarchism are suffering far-reaching erosion to a point where the word anarchy will become part of the chic bourgeois vocabulary of the coming century—naughty, rebellious, insouciant, but deliciously safe. (Bookchin, 1995, p. 3).[6]

The anarchic element of FLOSS' anarcho-utopianism is rather simply, but quite usefully, framed by what Bookchin (1995) has identified as the twin goals of anarchism: social democracy and individual liberty. For Bookchin, the history of anarchism reveals two basic but interrelated strains. Stemming from the work of Kropotkin and Bakunin, "social anarchism" seeks to create a grass-roots political system that tends to uphold principles of social democracy over the rights of individuals. It is to this tradition that Stallman belongs. Raymond, on the other hand, adheres more closely to "lifestyle anarchism," advocated by Proudhon and Godwin, which places individual rights above all else and offers up a more libertarian political system (Bookchin, 1995). Raymond is a proud libertarian, and his personal Web pages on the OSI's website include a link to the U.S. Libertarian Party as well as an extended discussion of his affinity for guns and his right to bear arms (Raymond, 2004). Bookchin's concern is that while lifestyle anarchism is positive insofar it advocates individual liberty, it ultimately undermines social anarchism by focusing on transient notions of individualism, rather than upon the more valuable goal of generating a sustainable form of social democracy. Given the debates between Raymond and Stallman over the future direction of FLOSS, it would appear that they provide an excellent case study of Bookchin's scenario. Relating this to the FLOSS movement in general, and the divergent goals and practices of the FSF and OSI more specifically, will require a brief discussion of the relationship between technology and anarchism.

Bookchin argues that the history of lifestyle anarchism exhibits a tendency to reject complex large-scale social structures and industrial/technological practices in favour of smaller, simpler social formations and more agrarian and/or craft-

based modes of production. William Morris' nineteenth-century anarcho-utopian Arts and Crafts movement might seem to embody this trend, but it actually resonates more clearly with social anarchism insofar as Morris included a fair degree of technological sophistication in his imagined future society (automated barges, for example). Even though his primary concern was to retain and/or reinstate craft-based production, he did so in terms of a rejection of the social relations of production that organized technological practices—industrialization, private property, and centralized governance. Rejecting and overcoming these social factors were, for Morris, of more importance than the rejection of any particular technology that might stem from them. For this reason alone, Morris' utopianism falls more squarely on the side of social anarchism. Indeed, Morris worked with and admired the work of Kropotkin, who saw in Morris' utopianism the basic elements of his own socialist anarchism.

Bookchin decries the anti-technological primitivist turn in so much anarchist thought and attributes it to the tendency of lifestyle anarchists' to reject technology and large-scale social systems as inherently inimical to human freedom. Using Mumford (1963) as a starting point, Bookchin (1995) rejects primitivist views as being counterproductive because:

> denouncing technology and civilization as inherently oppressive of humanity in fact serves to veil the specific social relations that privilege exploiters over the exploited . . . . Such concealment shields from public purview the causal role of capitalist competition in producing the crises of our time. To these mystifications, antitechnologists and anticivilizationists add the myth of technology and civilization as inherently oppressive, and they thus obscure the social relationships unique to capitalism—notably the use of things (commodities, exchange values, objects—employ what terms you choose) to mediate social relations and produce the technological landscape of our time. (p. 33)

From Bookchin's point of view, the primary struggle for power is not between anarchism and technology, but between anarchism and capitalism. To focus too intently on the capacity of various technologies to enact and maintain the social relations of production and, more broadly, power relations is thus to risk obscuring the fact that technologies are, themselves, the products of power-knowledge relations. Taken as a whole, the FLOSS movement tends to exhibit an inherent understanding of the importance of reigning power relations when it comes to software. This is perhaps nowhere more evident than in Stallman's GNU licence, which, as I have already noted, is vitally important because it addresses the social production, distribution, and use of software rather than its technical qualities. As a whole, the FLOSS movement's anarcho-utopianism thus resonates with Morris' because it, like Morris' Arts and Crafts movement, constitutes an attempt to reorganize a significant component of capitalism by reformulating the social relations of production that underwrite it along anarchic and communal lines.

At the heart of both the Arts and Crafts and FLOSS movements, there is a clear concern with community (whether artisan or hacker). In the FLOSS movement, however, there is a subtle yet important distinction to be made between the FSF and OSI. The utopian impulse of Stallman's FSF aligns more closely with

social anarchism, inasmuch as it foregrounds the role of instituting and sustaining programmer/user communities that have some sort of continuity. The OSI mitigates its anarcho-utopianism through its tendency to focus more on the continual re-constitution of communities around particular FLOSS projects. As Stallman (1999) argues, "the rhetoric of 'open source' focuses on the potential to make high quality, powerful software, but shuns the ideas of freedom, community and principle" (p. 70). Community thus tends to become somewhat of a means to an end in the OSI. In the FSF, the distinction between communal means and ends is less defined because generating a community of programmers and users appears to be as much of an end as is the software that it produces and uses.

Barlow's cyber-utopian declaration, Stallman's GNU Manifesto, and Raymond's free market libertarianism all exhibit anarchism's combinatory interest in individual autonomy, group action, and communal social systems. Regardless of the prioritization of these multiple goals, the concept of emergent organizational forms always and everywhere underwrites both anarchist and utopian thought. The popular tendency to understand anarchism in terms of chaos—as an utter lack of organization—rather than as a different modality of organization—is a problem that does not commonly appear in relation to utopianism (unless, of course, it is an anarchist utopia). In order to assess the viability of FLOSS' anarcho-utopianism, it is useful to think of anarchy as a spontaneous form of order (or organization) that, given certain conditions, is capable of being a stable system maintained over time. Hirshleifer (1995) argues that:

> Anarchy is not chaos. At least potentially, anarchic relationships can constitute a stable system. But not all environments are capable of sustaining an anarchic order. Anarchy can break down, to be replaced by another pattern of relationships . . . . So, as defined here, anarchy is a social arrangement in which contenders struggle to conquer and defend durable resources, without effective regulation by either higher authorities or social pressures. (p. 26-27)

For the purposes of the arguments presented here, the struggle could be defined as one between FLOSS and the traditional software industry, with the durable resource being software, and the environment being computing. There is thus a situation wherein FLOSS and the software industry are struggling for control over the means of software production (and its attendant communities and markets) within an environment of near-ubiquitous computing (at least in highly developed economies).

Hirshleifer proposes that anarchic economies may become stable where they achieve "optimization"—a viable balance of effort between activities of production and conflict—and "equilibrium"—an internally sustainable system of resource distribution that will allow an organization to maintain its optimization practices. Applying this to the FLOSS situation, it could be said that the current internal conflict between the FSF and OSI is the active working-out of the effort to achieve equilibrium, so that broader strategies of optimization may be carried out in the "war" against proprietary software regimes. By the same token, Hirshleifer's

argument also sheds light on the reason why the FSF and OSI can diverge in approach while still remaining bound to the wider cause of the FLOSS movement.

According to this model, there could be a stable anarchic form constituted by competition between two (or more) differing, yet wholly proprietary, software regimes. "Anarchy" here would not refer to a sociopolitical ideology, but simply to a modality of competition among organizational forms, which could be of any sociopolitical stripe. What I am discussing here, however, are anarcho-utopian organizations (FLOSS) in competition with classically capitalist organizations (the proprietary software industry). The stakes are thus somewhat higher, inasmuch as in the first scenario the "winning" competitor simply takes over the market, whereas in the latter case the winner gets to "make the rules" for the market itself. Currently, it would seem that there is a situation of stable anarchic competition between FLOSS and the proprietary software industry, because both coexist and compete for more or less the same resources (markets). Indeed, tactical concessions have been made on both sides (for example, IBM's turn to open source servers and the OSI's courting of venture capital for software start-ups).

It is quite possible that this process of conflict resolution may continue to the point where each side absorbs parts of, and/or insinuates itself into, the other such that the conflict resolves itself into a situation that produces a complex and interrelated FLOSS/proprietary software industry rather different from what we have today. On the other hand, Hirshleifer (1995) argues that anarchic systems can simply break down into either "amorphy"—a lack of form akin to chaos—or into "tyranny"—the dominating triumph of one competitor over any that remain (p. 27-29). If FLOSS practices were to be contained, co-opted, or otherwise redeployed by the proprietary software industry such that their software monopolies were maintained, then the current anarchic competitive system would break down into tyranny. At least, that is how the FLOSS community would describe the situation—Microsoft would no doubt cast it in a far different light. If, on the other hand, the FLOSS movement were to devolve into a disconnected series of non-cooperative programming enclaves, then the resulting amorphousness would likely eradicate FLOSS' ability to achieve internal equilibrium in order to optimize its competition with its proprietary software rivals. Just as the initial success of FLOSS depended upon a social rather than technical factor (the GPL), the relative success or failure of FLOSS' anarcho-utopianism likely rests upon its ability to maintain itself as a stable anarchic organizational form, rather than upon the relative technical merits of its software.

Struggles, strategies, and the current state of FLOSS

> I analogize opensource development to a free market in Adam Smith's sense and use the terminology of classical (capitalist) economics to describe it. . . . I advance an argument for the biological groundedness of property rights and cite Ayn Rand approvingly on the dangers of altruism. . . . [O]pensource development and the post-industrial capitalism of the Information Age are natural allies.

> In fact, I find the imputation of Marxism deeply and personally offensive as well as untrue. . . . [I]t is no secret in the opensource world that I am a libertarian, a friend of the free market, and implacably hostile to all forms of Marxism and socialism (which I regard as coequal in evil with Naziism [*sic*]). (Raymond, 1999c)

It would be difficult to find a more cogent sign of FLOSS' struggle for internal equilibrium than Raymond's response to charges launched by the proprietary software industry that FLOSS is redolent of socialism. This, in combination with the gradually increasing appeal of the OSI's version of FLOSS as a metaphor and model for businesses, lends credence to Stallman's fear that the high-minded socialist goals of the FSF may easily be lost if the OSI's tactics and rhetoric supplant his own. It also parallels Bookchin's concern that lifestyle anarchism may indeed be an easily, and safely, appropriated set of practices even if they are, at the moment, a bit "naughty." On the other hand, FLOSS' basic programming practices are founded upon practices initiated by the more socialist FSF, and they therefore confront and undermine the industrial control of copyright and intellectual property in a forceful manner. It may very well be that the FSF's anarcho-utopian ideals will have little influence if OSI-style FLOSS is embraced on a large scale. But this is to forget that the anarcho-utopian programming practices embedded in the FLOSS movement are not, as Raymond (1991) himself points out, not only not under the control of any centralized agency, but that they preexisted FLOSS.

These pre-FLOSS practices were amorphous, to be sure, but it was Stallman's GNU Project that brought them together into a stable anarchic organization. As Abbate (1999) has documented, the anarchic organizational form of the Internet and much of the early software industry is largely the result of its having been produced under the aegis of the institutional and academized academic computing practices of universities and government agencies—the very environment that Stallman worked within and subsequently left, after being asked to sign a non-disclosure agreement in 1982 (Stallman, 1999). The military-industrial complex that emerged during the Cold War had the peculiar effect of generating a robust and remarkably free research environment for the development of information and communication technologies. The promotion of high levels of co-operation among government agencies and business organizations came as a result of a desperate struggle to maintain a military-technological advantage over the Soviet Union. This situation made it difficult and disadvantageous to allow very much copyrighting of software, because it could hobble research efforts by making access to software more competitive and complicated.

Emerging from this highly cooperative environment, the Internet has subsequently become the "place" where the FLOSS movement can gather, share knowledge, work, and distribute its products. No single institution—military, government, or university—can provide the same interconnectivity, interaction, or peer group as the Internet. It is therefore an essential mechanism whereby a sufficient number of programmers can engage in FLOSS projects to produce robust software via what amounts to part-time voluntary labour. It is not surprising that

many features of the academic computing environment that spawned the Internet should form the basis for the FLOSS community. Such features include: available and accessible open source code, knowledge sharing via freely available (online) publications, inter-institutional forms and mobility, and a tendency to coalesce around research projects, interests, and problems, rather than organizations. In terms of its history and functioning, FLOSS' anarchic organizational form is thus more akin to the functioning of scientific and academic communities than it is to any particular anarchist movement. Nevertheless, as FLOSS programmers began to self-identify as a community after Stallman's introduction of GNU, they began to create and vigorously defend the loosely formulated anarcho-utopian discourse of the hacker ethic. The result of all of this is that the challenge to proprietary software regimes from FLOSS' mode of organization, its knowledge economy, and its property relations comes, in large part, from the social institution of academia rather than business.

The fact that the OSI not only co-operates with the market, but seeks to find a place within it, is a crucially different approach than the FSF's direct confrontation of market hegemony. If Stallman is correct in arguing that the openness introduced by the FSF is incorporated by the OSI only at the level of software production itself—thus containing and integrating its FLOSS' communal practices in the service of existing market needs and structures—then the OSI's version of FLOSS may not *challenge* the traditional proprietary software industry so much as it may end up being a means for that industry to *maintain* its market hegemony. In that scenario, the OSI would really only threaten software giants like Microsoft, and would do so simply because it proffers a new business model.

The potential success of the new business model(s) emerging from the OSI is evident in a number of areas where proprietary software has been supplanted or radically modified by FLOSS applications. Open source Web browsers are widely available, open source "APACHE" Web servers power approximately 70% of the WWW, Apple's OS X is based on an open source Unix core (FreeBSD), and it goes without saying that the various flavours of the GNU/Linux operating system are gaining ground in the software marketplace. While none of this constitutes an outright triumph for FLOSS, it does suggest that the software industry is undergoing a significant transformation. The changes, however, are more or less invisible to the average user, because they exist largely at the infrastructural level of computing and networking (operating systems and Internet protocols). Public perception of the change lags behind that of the software industry, and wider distribution in both the business and consumer software market is not likely to come until FLOSS-based applications become easier to install, use, and maintain. GNU/Linux, Windows, and Mac OS X users with above-average skills already have access to FLOSS versions of key applications. OpenOffice (a software suite comprising word processing, spreadsheet, database, drawing, and presentation programs software suite) and GIMP ("GNU Image Manipulation Program," for image editing) are robust FLOSS projects that offer FLOSS replacement alternatives to Microsoft Office and Adobe Photoshop. The FLOSS community's hope is

that as the installed base of FLOSS operating system users increases, so too will awareness, demand, and, perhaps most importantly, the pool of available and active programmers for FLOSS applications.

In addition to the OSI's ongoing promotion of open source software as a viable alternative for businesses to the Microsoft dominated PC operating system market, the advance of FLOSS into the existing software marketplace has been boosted by the adoption and support of GNU/Linux by PC computer manufacturers like Hewlett Packard and Compaq and GNU/Linux distributors such as Red Hat. The gradual increase in the adoption of FLOSS applications into existing markets has, however, had the unfortunate effect of producing numerous variations of open source licences in order to accommodate particular instances of mixed open and proprietary software. One task that the OSI has set for itself is to provide a means by which to certify these licences (58 at last count) by way of offering them in full on the OSI website and allowing approved licences to bear the trademarked "OSI Certified" symbol. Licence proliferation is clearly a problem in terms of the confusion it may cause for producers, distributors, and users. More importantly, licence proliferation has led to the introduction of non-certified licences that have the potential to undermine key carry-forward provisions either by error or intent.

The FSF is addressing this issue by releasing a new GPL ("GPL3") by the end of 2007. While little is known about the actual language of the upcoming version of the licence, GPL3 is expected to address the increasingly important issue of software patents. The FSF and OSI both fear that software patents will be used to limit or perhaps halt open source development, because patents bypass current open source licensing practices by allowing litigation over software functionality, rather than methodology. Software patents allow a company to patent a functional feature of a program, rather than simply the source code. One famous example is Amazon.com's attempt to patent the "one-click" function for online purchases. This would have prevented, or required payment from, any site that deployed such a function for over the next twenty years, regardless of the specific programming methodology used. Amazon.com failed in their bid, but this has not stopped multiple patent-holding software giants (such as Microsoft) from applying for, and receiving, yet more function-based patents.

The Foundation for a Free Information Infrastructure (FFII, a European-based FLOSS advocacy group) argues that software patents may be, and often are, used as a means to bypass both copyright and copyleft provisions by enabling litigation based on function rather methodology:

> Programming is similar to writing symphonies. When a programmer writes software, he [*sic*] weaves together thousands of ideas (algorithms or calculation rules) into a copyrighted work. Usually some of the ideas in the programmer's work will be new and non-obvious according to the (inherently low) standards of the patent system. When many such ideas are patented, it becomes impossible to write software without infringing on patents. Software authors are in effect deprived of their copyright assets; they live under permanent threat of being blackmailed by holders of large patent portfolios. As a result,

> less software is written and fewer new ideas appear. (Foundation for a Free Information Infrastructure, 2005, URL: http://www.ffii.org)

SCO (formerly Caldera Systems) and Microsoft have both attempted to use the software-patent issue to cast GNU/Linux as an illegitimate and potentially illegal form of software distribution. While Microsoft's position has been delivered largely via the bombastic and unsubstantiated pronouncements of CEO Steve Ballmer (Vaughn-Nichols, 2004), SCO has actually launched a lawsuit against IBM over the rights to Unix and its open source variants (GNU/Linux and FreeBSD in particular). Set to be heard in U.S. courts in early 2007, SCO's case is rooted more in issues of copyright law than software patents. Nonetheless, the effort to introduce a "chilling effect" into the world of open source development and distribution is the same.

Software patents and existing copyright regimes therefore appear, at this point in time, to be the FLOSS movement's Scylla and Charybdis. It will take all of their tactical and rhetorical acumen to navigate these legal waters. This means that the FSF's and OSI's struggles for anarchic organizational equilibrium within the FLOSS movement will likely have to be worked out sooner rather than later if they are to fully engage in any form of competition with the always litigious proprietary software industry. If FLOSS can successfully deploy and defend new forms of copyright, then it will secure the right to retain as its central programming methodology the free (re)distribution of Unix-based open source code. If it loses, then the very core of FLOSS—the GNU/Linux code—will be pulled out from underneath it.

While the major struggle between FLOSS and its proprietary software rivals is in its infancy, it does appear that FLOSS may have the upper hand in terms of software copyright and, by luck rather than right, in terms of software patents. SCO's copyright case has been widely denounced as being without merit by both the FLOSS community and the proprietary software industry (with the notable exception of Microsoft). The case rests upon SCO's ability to demonstrate that by buying AT&T's rights to Unix after AT&T divested itself of Bell Labs permits SCO to retroactively regain the rights to versions of Unix that AT&T (via Bell Labs) had already allowed to be freely licensed. At the moment, it seems unlikely that this tactic will succeed. Software patent holders face a different problem: enforceability. The patenting of such ubiquitous functions as windows-based user interfaces and the use of "clickable" credit card icons on websites would seem to make the enforcement of functionality based patent rights almost impossible. Amazon.com's inability to win its case with regard to its "one-click" purchasing function does not bode well for subsequent attempts to enforce such patents. It would appear then that FLOSS may actually have the potential, if only by default, to navigate around the proprietary software industry's imposition of legal restrictions on software functionality via the patent process. On the other hand, the outcome of these looming and ongoing legal struggles will rest upon the ability of the interested parties to marshal their legal forces (and funds) as well as to lobby various national governments and supra-national organizations for favourable

laws, regulations, and policies. The final result will therefore ultimately rest on the social and political conditions that surround and inform the software industry, rather than on the relative technical merits of the software itself.

While clearly dedicated to issues surrounding copyright, the OSI has been rather silent thus far with regard to software patents, and this has been taken as a sign by the FSF and FFII that the OSI is perhaps rather fearful of upsetting the very organizations they wish to court, because many of them are major software-patent holders. The FSF's concern is that the demise of proprietary software giants like Microsoft at the hands of the more pragmatic (which is to say, business-friendly) OSI would not usher in a truly FLOSS-based industry so much as remove a proprietary monopoly and replace it with an oligopoly of ostensibly FLOSS-based "mini-Microsofts." These resulting organizations, while employing FLOSS principles, might do so in a manner that would marginalize, contain, or otherwise dilute FLOSS through meta-level proprietary software and software-patent claims. Microsoft's attempt to have its "shared source" licence certified by the OSI exemplifies the kind of activity that is of concern to the FSF. In an effort to simultaneously accept and stave off FLOSS, Microsoft has introduced a modified version of open licensing, wherein code made publicly available by Microsoft as "shared source" can indeed be shared, but the licence stays with Microsoft, and any commercial product developed that includes the code is subject to a fee payable to Microsoft. Thus, while the code is ostensibly open, the licence is not. The result is a form of open source development without the GPL's carry-forward provision (thus defeating the very purpose of the GPL). It isn't difficult to see why Stallman fears this kind of scenario and suspects that the OSI is moving in a similar direction. From the FSF perspective, the fact that Microsoft believes that the OSI might certify their licence is reason enough to be highly suspicious of the future of FLOSS under the guidance of the OSI's licence certification practices.

Conclusions

With open source development poised on the verge of becoming technoculture's "next big thing," it is by no means certain whether the kernel of openness contained in the OSI's version of the FSF's "socialist doctrine" will be fully de- and re-territorialized in the service of status quo market conditions. Because FLOSS' practices are not only different from, but come from a different ordering of, knowledge and property relations than those of the existing commercial software industry, its organizational models may be less easily confronted, contained, or captured than that of traditional software industry rivals. The popular appeal of open source as both a metaphor and model for various businesses and organizations suggests that it is a set of practices that are can perhaps be easily appropriated and profitably exploited by the traditional software industry and, more broadly, by post-industrial capitalism. As it stands now, FLOSS practices are certainly being modified to some degree via the OSI's overtures to the business community, but FLOSS has already had a profound impact on the software industry and, by extension, on the institutions, organizations, groups, and individuals that use its software for myriad purposes on an everyday basis. As open source devel-

opment and distribution gains ground in broader software markets, it seems inevitable that transformations of practice and ideology will occur on both sides of the software industry. Given the relatively recent emergence of FLOSS in the mainstream of society, trying to predict what these changes might be is rather like predicting what the Internet would become (and is still becoming) when it made its way into the mainstream in the late 1980s. The ultimate organizational form and technical efficacy of FLOSS remains to be seen. Nonetheless, the story so far has been one of successfully building upon the strengths of proven projects like the APACHE server and GNU/Linux, while at the same time managing to avoid the legal constraints of copyright law and software patents.

In the end, however, it will not be the relative organizational and/or technical success of open source development that matters most. It is the sociopolitical dissatisfaction and utopian desire for anarchic openness expressed in FLOSS' doctrines and practices that is of most interest, because it speaks to a broader set of political concerns related to contemporary society. Seen in this light, the FLOSS world is no "nerdy" backwater consisting of arcane programming cultures and practices. Rather, the FLOSS movement's struggle over property rights and its advocacy of a kind of cultural commons for computing significantly overlaps with the anti-globalization movement's defence of public goods, practices, and culture from the hypercommodified practices of post-industrial late capitalism. Given the increasing digitalization of cultural products and practices, it seems logical that these two movements should share some affinities with respect to their goals and means for achieving them. The potential for these movements to come together in some way by teasing out their shared anarcho-utopian tendencies is certainly an intimation of hope. There is also an intimation of deprival; the prospect that the FLOSS movement's success will likely hinge upon its ability to find a relatively unified and stable, albeit anarchic, organizational form is worrisome. It would only be a pyrrhic victory if the wide-scale adoption of FLOSS-style practices succeeded for reasons of technical superiority, yet deprived the movement of its higher ideals.

Beyond the FLOSS movement's technical discussions of programming and code there lies a more fundamental concern with the re-negotiation of the social and industrial relations of property, knowledge, and, ultimately, power. The real significance of the FLOSS movement's anarcho-utopianism may therefore very well be that it is an intimation of something else entirely: a broad and creeping (viral?) dissatisfaction with the knowledge and property regimes of contemporary technoculture. This discourse is important and requires greater elaboration within the FLOSS movement, and further investigation from outside the movement, if only for the simple reason that it is an expression of revolt against the juridical constraints placed upon the production of culture and, concomitantly, cultures of production.

Notes

1. This more inclusive acronym has been proposed in order to replace the Anglo-centric FOSS (sometimes F/OSS), which simply refers to "free and open source software." Adopted by the Euro-

pean Commission in 2002, the acronym "FLOSS" combines the English-speaking acronyms for free software and open source as well as incorporating European acronyms for the same (with "F" referring to both the English "free" and German *frei*, and "L" being used to denote both the French and Spanish *libre*, Italian *libero*, and Portuguese *livre*).

2. "GNU" (GNU's Not Unix) is Stallman's playfully recursive acronym for the free/open source operating system (and related software) intended as a replacement for proprietary forms of Unix and other operating systems
3. While Stallman and the GNU community have yet to produce a fully functional independent operating system based solely on GNU components, they have produced a number of essential software elements that, when later combined with the Linux kernel, constitute an independent operating system. Hence the term "GNU/Linux."
4. Intended to combat the circulation of pornography and promotion of violence and hate, the provisions of the CDA may not have been quite as nefarious as cyberspace's early adherents and advocates suggested, but the CDA nevertheless overstepped much of the existing law concerning pornography and freedom of speech. Founded in 1990 in order to protect hackers from malicious prosecution, promote online privacy rights, and advocate for freedom of speech on the Internet, the Electronic Frontier Foundation (EFF) took up the battle against the CDA. With the help of the American Civil Liberties Union and many other organizations, CDA was defeated after two Congress-led attempts to revive it. The "blue ribbon" campaign was key to the EFF's success because it publicized the issue in a very visible manner: a significant number of websites protested the CDA by displaying an image of a blue ribbon and the statement "Stop Internet Censorship/Free Speech Online" on their pages. As a show of protest, many websites were voluntarily taken offline for 48 hours beginning on February 8, 1996 (the day then President Clinton signed the act into law). These forms of protest were quite effective because they were difficult to miss in a time when the WWW was relatively limited in scope when compared with its present day version. Any casual user of the Internet and WWW could not fail to run across a site with a blue ribbon or, for a short period, no site at all, other than a stark black and white page giving detailed reasons for the protest and/or a link to the Electronic Frontier Foundation.
5. Barlow notes in the introduction that precedes the declaration that it was written with "characteristic grandiosity." In keeping with the hacker ethic, he also states that any recipient of the e-mail should feel free to pass it along—with or without crediting Barlow as the author—such that it may grow, change, and replicate.
6. It is an interesting coincidence that the timing of Bookchin's statement was published in the year between the public deployment of the WWW and the introduction of GNU/Linux.

References

Abbate, Janet. (1999). *Inventing the Internet.* Cambridge, MA: MIT Press.

Barlow, John Perry. (1996). *A cyberspace independence declaration.* URL: http://www.eff.org/Misc/Publications/John_Perry_Barlow/barlow_0296.declaration.txt.

Bookchin, Murray. (1995). *Lifestyle anarchism or social anarchism: An unbridgeable chasm.* San Francisco, CA: AK Press.

Campbell-Kelly, Martin. (2003). *From airline reservations to Sonic the Hedgehog: A history of the software industry.* Cambridge, MA: MIT Press.

Himanen, Pekka. (2001). *The hacker ethic and the spirit of the information age.* New York, NY: Random House.

Hirshleifer, Jack. (1995). Anarchy and its breakdown. *The Journal of Political Economy, 103*(1), 26-52.

Imhorst, Christian. (2005). Anarchy and source code—What does the free software movement have to do with anarchism? *Free/Open Source Research Community.* URL: http://opensource.mit.edu/papers/imhorst.pdf.

Jameson, Fredric. (1988). Of islands and trenches: Neutralization and the production of utopian discourse. *The ideologies of theory: Essays 1971-86, Vol. 1: Situations of theory* (pp. 75-101). Minneapolis, MN: University of Minnesota Press.

Klang, Mathaias. (2005, March). Free software and open source: The freedom debate and its consequences. *First Monday, 10*(3). URL: http://firstmonday.org/issues/issue10_3/klang/index.html.

Levitas, Ruth. (1990). *The concept of utopia.* Syracuse, NY: Syracuse University Press.

Levy, Steven. (1985). *Hackers: Heroes of the computer revolution.* New York, NY: Penguin.

Marin, Louis. (1984). *Utopics: Spatial play.* (Robert A. Vollrath, Trans.). Atlantic Highlands, NJ: Humanities Press.

Marx, Karl, & Engels, Friedrich. (1967). *The communist manifesto.* London, U.K.: Penguin. (Originally published 1848)

Morris, William. (2003). *News from nowhere, or, an epoch of rest: Being some chapters from a utopian romance.* Oxford, U.K.: Oxford University Press. (Originally published in 1890)

Mumford, Lewis. (1963). *Technics and civilization.* New York, NY: Harcourt, Brace & World.

Murray, Kevin. (2000). *Back to nowhere.* URL: http://www.kitezh.com/texts/nowhere.htm.

Parker, Martin. (2002). Utopia and the organizational imagination: Eutopia. In Martin Parker (Ed.), *Utopia and organization* (pp. 217-224). Oxford, U.K.: Blackwell.

Popper, Karl. (1963). *The open society and its enemies.* Princeton, NJ: Princeton University Press.

Raymond, Eric. (1999a). *The cathedral and the bazaar.* Sebastopol, CA: O'Reilly & Associates.

Raymond, Eric. (1999b, June 28). Shut up and show them the code. *Linux Today.* URL: http://linuxtoday.com/news_story.php3?ltsn=1999-06-28-023-10-NW-SM.

Raymond, Eric. (1999c, November 1). A response to Nikolai Bezroukov. *First Monday, 4,* (11). URL: http://www.firstmonday.org/issues/issue4_11/raymond/.

Raymond, Eric. (2004). *Personal website.* URL: http://www.catb.org/~esr/personal.html.

Rushkoff, Douglas. (2003). *Open source democracy: How online communication is changing offline politics.* London, U.K.: Demos.

Scoville, Thomas. (2001). *Whence the source?: Untangling the open source/Free software debate.* O'Reilly.com. URL: http://opensource.oreilly.com/news/scoville_0399.html.

Stallman, Richard. (1983). *The GNU manifesto.* URL: http://www.gnu.org/gnu/manifesto.html.

Stallman, Richard. (1999). The GNU operating system and the free software movement. In Chris DiBona, Sam Ockman, & Mark Stone (Eds.), *Open sources: Voices from the open source revolution* (pp. 53-70). Sebastopol, CA: O'Reilly & Associates.

Tiemann, Michael. (1999). Future of cygnus solutions: An entrepreneur's account. In Chris DiBona, Sam Ockman & Mark Stone (Eds.), *Open sources: Voices from the open source revolution*. pp. 71-90. Sebastopol, CA: O'Reilly & Associates.

Vaughn-Nichols, Steven J. (2004, November 19). Author of GNU/Linux patent study says Ballmer got it wrong. *eWeek.* URL: http://www.eweek.com/article2/0,1759,1729908,00.asp.

Wark, McKenzie. (2004). *A hacker manifesto.* Cambridge, MA: Harvard University Press

Williams, Sam. (2002). *Free as in freedom: Richard Stallman's crusade for free software.* Sebastopol, CA: O'Reilly & Associates.

Websites
Electronic Frontier Foundation. URL: http://www.eff.org
Foundation for a Free Information Infrastructure. URL: http://www.swpat.ffii.de
Free Software Foundation. URL: http://www.fsf.org
GNU Project. URL: http://www.gnu.org
Open Source Initiative. URL: http://www.opensource.org/index.php