The Political Economy of Communication
2(2), 3–19

http://www.polecom.org

Conflict in the Commons: Towards a Political Economy of Corporate Involvement in Free and Open Source Software

Benjamin J. Birkinbine, *University of Nevada*

Keywords: free software; Open Source; digital labor; copyleft; commons; forking; interoperability

Abstract

Free (libre) and open source software (FLOSS) gives users the right to study, modify, adapt, or improve upon software by granting access to the source code. Proprietary software, on the other hand, relies on restricting certain uses of software to protect the marketability of intellectual property. From its beginnings in the 1980s, FLOSS has provided a radical alternative to proprietary software and is generally heralded as one of the triumphs of commons-based peer production. However, corporations are becoming increasingly involved in FLOSS projects, which would seem to be a clash of interests. In this paper, I argue that the increasing corporate involvement in FLOSS projects potentially threatens the radical potential of FLOSS by directing commons-based development toward corporate goals. To date, the FLOSS community has been able to leverage its collective labor power to counteract such corporate maneuvering. This type of collective action relies on a class consciousness among software laborers that needs to be maintained if these strategies are to remain effective. To illustrate how these dynamics operate within the software industry, I focus on the acquisition of Sun Microsystems by Oracle and its consequences for FLOSS development.

Free (libre) and open source software (FLOSS) emerged as a radical alternative to proprietary software during the 1980s. Since then, FLOSS has been heralded as one of the success stories of commons-based peer production (Benkler, 2006). However, for-profit corporations are now becoming increasingly involved in FLOSS projects. For example, the Linux Foundation listed Red Hat, Intel, Texas Instruments, IBM, Samsung, Google, and Oracle among the top developers of the Linux kernel in its 2013 annual report (The Linux Foundation, 2013: 9). The kernel is a key part of a computer operating system that facilitates communication between software and hardware [1]. On its face, corporate involvement in FLOSS projects would seem to be a contradiction of interests, particularly because most FLOSS projects are freely available to consumers. Furthermore, the interests of the corporation may clash with the underlying ethos of the broader FLOSS community. That ethos is generally characterized by a belief in productive freedom, defined as the community's right to "autonomously improve upon their peer's work, refine their technical skills, and extend craftlike engineering traditions" (Coleman, 2013: 3). As such, corporations may run the risk of violating these cultural norms when they increase their involvement in FLOSS projects, especially if they attempt to influence the direction of the project in a way that would only benefit the corporation. In such a situation, a conflict may occur within the commons, whereby a community of developers working together to create a FLOSS project is met with unwanted corporate influence in their project.

To illustrate how this type of conflict can occur, this paper focuses on the histories of three FLOSS projects and how they were affected by a case of corporate acquisition that was beyond the control of the community. More specifically, I focus on the different ways that the Oracle Corporation attempted to influence, or altogether eliminate, development of three software projects: the OpenSolaris operating system, the MySQL relational database management system, and the OpenOffice office productivity software (after Oracle's acquisition of Sun Microsystems). In the face of this unwanted influence, members of FLOSS communities resisted Oracle's actions by abandoning these projects and focusing their development efforts elsewhere. The central argument presented here is that the FLOSS community maintains a unique ability to leverage its collective labor power against corporate encroachment into its projects by using technical, legal, and governance strategies that allow them to abandon a project without losing the products of their labor.

To demonstrate this, I begin by explaining how and why a critical political economy approach is useful for understanding corporate involvement in FLOSS projects. Next, I briefly describe some of the shared ideals of the FLOSS community, as well as the strategies used by corporations when sponsoring FLOSS projects. Then, I discuss Oracle's acquisition of Sun Microsystems and how three FLOSS projects were affected by corporate maneuvering. Finally, I reflect on how FLOSS communities leverage their collective labor power and why it is increasingly important for the FLOSS community to maintain these capacities.

Critical political economy and digital labor

The present study is informed by a critical political economy approach to the study of communication. Researchers working within this tradition are interested in the "social relations, particularly the power relations that mutually constitute the production, distribution, and consumption of media resources" (Mosco, 2009: 24). According to Mosco (2009), a political economic approach is characterized by a concern for social change and history, social totality, moral philosophy, and praxis. Furthermore, *critical* political economists look for moments of contradiction in social relations that offer possibilities for resistance. Most often, the critiques provided by political economists of communication are directed at media corporations that operate according to an industrial or commercial logic that often fails to meet the true needs of citizens (see Meehan, 2005; Murdock and Golding, 1973; Wasko, 2001). In part, the industrial logic guiding media production is due to the commercial nature of media corporations, which requires them to seek out large audiences for media content that can be sold to advertisers. As Dallas Smythe (1960) argued, audiences constitute a commodity that can be sold to advertisers on behalf of media corporations. By doing so, the working and leisure time of the audience become collapsed so that even the cultivation of interests outside of working time can be commodified into demographic and psychographic data about one's interests. This in turn, provides valuable data that can be sold to marketers and advertisers who want access to data about consumer preferences. Because capital relations invade an increasing part of everyday life, Smythe and others working within critical political economy place class and capital relations at the center of their analysis.

A similar industrial logic drives development in the software industries. As such, the critiques offered by political economists can be directed at the software industry in general and at the commercial exploitation of free and open source software projects more specifically. Critical political economy is particularly well suited to such an analysis because it focuses our attention on the commodification of communication resources as well as the labor relations that underpin those processes. Moreover, critical political economists take seriously the project of finding sustainable alternatives to the commercial exploitation of communicative resources.

The present study is informed by such a perspective, although the focus on FLOSS workers' ability to define their own group dynamics apart from purely capitalist firms as a way to protect their right to productive freedom (Coleman, 2013) places the analysis closer to an autonomous Marxist approach (Hardt and Negri, 2000; Lazzarato, 1996; Terranova, 2004). The autonomous approach focuses on workers' ability to define themselves independently from capital, while focusing on the different strategies for resistance that are possible in all aspects of social life. As will be demonstrated throughout this paper, the FLOSS community is positioned both in the service of capital, while at the same time resisting it in certain ways. Moreover, because the work of these communities is primarily associated with the creation and maintenance of digital systems and tools, we might classify their work as immaterial or digital labor.

The terms 'immaterial labor' and 'digital labor' have found increased currency in contemporary debates about online life. FLOSS labor can be viewed as a form of "immaterial labor" insofar as the final products of such work are "immaterial products such as knowledge, information, communication, [or] a relationship" (Hardt and Negri, 2005: 108). The term 'immaterial labor' was first introduced by Lazzarato (1996) and has since been debated by critical scholars [2]. Similar debates have occurred within critical scholarship circles about the nature of 'digital labor' (see Scholz, 2013). The primary concerns of these debates relate to the nature of work and labor within the information, knowledge, and communication industries. Within this work, particular attention has been paid to forms of unpaid labor occurring online (see Andrejevic, 2012; Fuchs, 2012). As an increasing amount of time is spent online during both work and non-work time, our digital labor—socially necessary time spent online, offers the possibility for exploitation to occur as browsing data is extracted from users and transformed into value by service providers and other third-party elements (Fuchs, 2011).

FLOSS projects also offer a form of digital labor and, in many cases, a form of unpaid labor. In FLOSS production, volunteers contribute directly to the development of a technology rather than having data extracted from their online behaviors. Therefore, the value of FLOSS projects does not necessarily lie in data about development but in the utility of the end products of the development process [3]. In this sense, the technologies produced by FLOSS communities may have use value for both the community as well as the corporation that sponsors the community. While not all FLOSS projects have a corporate sponsor, the corporations sponsoring such projects benefit from the exchange value of FLOSS projects when certain components from those projects are incorporated into their proprietary software. The full range of complexity involved in this type of appropriation cannot be completely detailed here because of the particularity of various project arrangements, but two strategies are generally used to facilitate this type of behavior.

The first strategy is the development of unique intellectual property licenses to protect the code produced by FLOSS communities. In certain cases, these licenses are designed by the corporation to allow portions of the source code to be incorporated into proprietary software or vice versa [4]. The specific terms of these licenses vary greatly and have a dramatic effect on the overall attractiveness of the FLOSS project to contributors (Santos et al., 2012). The second strategy is the use of contributor agreements, whereby those wanting to contribute to a particular FLOSS project must sign an agreement that stipulates what types of intellectual property licenses are allowed to be assigned to their contributions. In some cases, contributors may be asked to forfeit their claims to intellectual property altogether, thus granting ultimate ownership of their work to the sponsoring organization [5]. Since signing these agreements is a prerequisite for participation in the sponsored project, contributors to these projects are aware of the terms prior to their involvement. However, this is not to say that the terms are not subject to change, especially when a sponsoring organization is acquired by another firm.

The exact point at which corporate involvement in a project becomes an unwanted encroachment will vary between projects and their respective communities. Each community supporting a particular project determines what is and is not permissible through open dialog with other members of the community. Through this process, the community develops a shared consciousness that functions as a sort of moral economy (Thompson, 1971). The moral economy of the FLOSS community as a whole is based on the shared ideals of peer-to-peer relationships, collaborative development, transparency, community, and productive autonomy (Coleman, 2013).

When a corporation violates this moral economy, the community can rebel by effectively halting production on a project. In this sense, the actions of the community have an effect similar to that of a strike or factory walk out. While this analogy may be useful for considering how labor resists corporate exploitation, there are qualitative differences in the consequences such an action has for the corporation as well as the community. Whereas a strike or factory walk out halts production on the factory floor, the technical and legal characteristics of FLOSS products enable the community to move their production elsewhere by ‘forking’” their project. When a project is forked, the underlying source code of a software is copied, and independent development of the software project can continue. The corporation, while still maintaining access to the production performed by the community in the past, loses the prospect for future production of that same project. This is particularly harmful in software development because if development ceases, the software may not be able to interface with new technological systems. Moreover, the corporation risks having its reputation tarnished within the community, thereby decreasing the community’s willingness to contribute to its projects.

While more precise measurements of the reputational and, ultimately, economic consequences for corporate violations of a community’s moral economy are undoubtedly deserved, the present study focuses on how the FLOSS community used forking as a strategy for resisting unwanted corporate encroachment into its projects. More specifically, I focus on how one—Sun Microsystems, maintained a relatively good relationship with the FLOSS community by remaining transparent about its intentions for FLOSS projects that it supported. When Sun Microsystems was acquired by Oracle Corporation, the relationship with the FLOSS communities became strained, as Oracle attempted to alter the nature of those relationships. Oracle used different strategies to effectively discontinue FLOSS development within its newly acquired properties, which can be illustrated by three case studies: the OpenSolaris operating system, the MySQL relational database management system, and OpenOffice, which was developed as an open-source alternative to Microsoft Office. Before proceeding to a discussion of these three case studies, some background information about the companies involved is provided, as well as a brief account of how and why FLOSS emerged as a prominent alternative to proprietary software development.

Oracle acquires Sun Microsystems

Oracle Corporation (hereafter simply Oracle) is one of the largest software companies in the world. The company has three main operating segments: software business, hardware business, and services (Oracle Corporation, 2013) [6]. In turn, these three segments are divided into seven smaller operating divisions: 1) new software licenses and cloud software subscription services; 2) software license updates and product support; 3) hardware systems products; 4) hardware systems support; 5) consulting services; 6) managed cloud services; and 7) education services. Out of its three main operating segments, Oracle earns nearly 75% of its total income from the software business segment. In 2013 alone, the company earned more than $37 billion in total revenues and employed approximately 120,000 people (Oracle Corporation, 2013). If calculated by total revenues, Oracle is the third largest company in the global software market behind only IBM and Microsoft. The reason Oracle has remained competitive within the global software market is, in part, because of its strategic acquisitions. One of the company's largest acquisitions took place when it acquired Sun Microsystems in 2010.

Prior to its acquisition by Oracle in 2010, Sun Microsystems provided network computing infrastructure solutions which included software, systems, storage, and microelectronics. In 2009, the final year of its independent operation, Sun reported approximately US$11.45 billion in revenues and employed approximately 29,000 employees in over 100 different countries (Sun Microsystems, 2009) [7]. The majority of the company's revenues (42%) came from its Systems operating segment, which included the sale of servers that provide computing power and storage space to customers as a key part of internet infrastructure. Other core brands owned by Sun Microsystems were the Java technology platform, the Solaris Operating System, MySQL database management software, Sun StorageTek storage solutions, and the UltraSPARC processor. Because its core business relied on the provision of infrastructure-based services and products, Sun was a large supporter of interoperability. Interoperability is the capacity of different programs to exchange data with one another by using common formats. To facilitate innovation and interoperability, Sun made its key intellectual properties freely available to support open standards, open interfaces, and open source software. By making a commitment to open source, Sun was viewed favorably by the open source community and maintained a relatively good relationship with the community because it was transparent about its corporate goals (J Riddle, 2013, personal communication). To better understand the reasons for Sun open-sourcing some of its key intellectual properties, the following section provides a brief history of how corporations became involved in FLOSS projects.

A brief history of operating systems and interoperability

Throughout the 1980s, the market for operating systems was dominated by proprietary versions of Unix-based operating systems. For example, Hewlett Packard offered HPUX, IBM offered AIX, and Sun Microsystems offered SunOS. These operating systems dominated high computing, or infrastructural level computing, while the consumer market was dominated by Microsoft's DOS, which was not based on Unix. Importantly, the proprietary Unix-based systems were *source-incompatible*, meaning that each distinct proprietary version of Unix could not function correctly with other derivatives. In effect, although these systems were all based on Unix, the development of separate proprietary versions had caused the code to diverge in such a way that programmers could no longer assume interoperability between systems. As a result, programmers had to maintain separate codebases for each system, and companies could sell entire stacks of software to their customers who had to accept the entire stack. This resulted in an inefficient system that was dominated by proprietary software vendors, while simultaneously increasing the workload for programmers (J Riddle, 2013, personal communication). During the mid-1980s, however, the Free Software Foundation (FSF) emerged as a response to the overly protective intellectual property restrictions placed on software (FSF, 2014). This in turn, led to the development of free and open source software, which was collaboratively developed as a commons-based resource for others to study, use, adapt, or modify in any way.

Because this model of development was so successful, by the mid-1990s, Linux, an open-source operating system, had become the dominant Unix-like operating system (Moody, 2001). Linux undercut the competition by offering a comparable product at a significantly lower cost. Because Linux was licensed under the GNU General Public License (GPL), an alternative form of intellectual property (copyleft), improvements to Linux could be shared by everyone, which improved the quality and stability of Linux. Proprietary companies could not compete with Linux because the commons-based peer production driving Linux constituted a larger labor force than any of the individual companies could employ. Rather than competing directly with Linux, certain proprietary companies began to open source their products as a way of attracting interest from the free and open source software community. By doing so, the companies could indirectly increase demand for proprietary versions of their software. Sun Microsystems was one of those companies.

Sun open-sourced their Solaris operating system, which became OpenSolaris. They also open-sourced the MySQL database management software, as well as StarOffice, which became OpenOffice. As mentioned earlier, Sun maintained a good relationship with the broader FLOSS community because of their commitment to, and support for, FLOSS projects. After the company was acquired by Oracle, this relationship was strained in certain ways. The following sections discuss how the developers working on each of the three projects mentioned above strategically resisted Oracle's influence in their projects.

OpenSolaris

In 1987, Sun Microsystems and AT&T announced that they were going to merge some of the most popular Unix-based operating systems into a single project. This project eventually became Solaris, which was a proprietary operating system held by Sun that contained both open-source and closed-source components. To attract interest in the project and build a community of users and developers around the project, Sun Microsystems created OpenSolaris. OpenSolaris was an open-source version of the Solaris operating system, although OpenSolaris did contain some elements in the code that were not open source. After attracting a larger community of interest in the project, a Community Advisory Board (CAB) was created to direct the project. The CAB was comprised of two Sun employees, two members who were elected by the broader community, and one member who was appointed by Sun from the broader free software community (Sun Microsystems, 2005). In effect, most of the CAB members were connected with or appointed by Sun, and Sun made clear what its intentions were for the OpenSolaris project.

Sun's strategy for OpenSolaris, as well as the other open source projects that it supported, was clearly defined in its annual reports prior to its acquisition by Oracle:

> Our open source initiatives are intended to increase participation in software and hardware design by making our innovative hardware and software intellectual property freely available. A core premise to the success of our software business is our ability to attract innovative application developers to our Java platform and Solaris Operating System. We build relationships with these communities of developers to stimulate demand for our commercial products and services (Sun Microsystems, 2009: 3).

In effect, Sun supported open source projects for two reasons. First, the company sought to attract interest in its products from developers who could tinker with the open source code. Allowing this type of experimentation was a way to foster innovation that could be incorporated into its proprietary versions of the same software. Second, Sun could increase overall demand for its software by making its source code freely available to others. The claim was that increased availability and use of its open source code would translate into greater demand for its software and services, particularly among other enterprises who would be willing to pay for access to those services.

To facilitate the incorporation of open source innovation into its proprietary offerings, Sun protected OpenSolaris under a free software license created by a company called the Common Development and Distribution License (CDDL) [8]. This license enabled Sun to include proprietary, free software, or software protected under any other license in their Solaris and OpenSolaris operating systems. Consequently, Sun could use the OpenSolaris community as a way to drive development, quality control, or innovation that could be included in their proprietary Solaris offering. Importantly, however, Sun made this strategy very clear to the OpenSolaris community, and Sun was supportive of the broader FLOSS community. Once Oracle acquired Sun, they took a very different approach to this strategy.

After its acquisition of Sun, Oracle announced plans to discontinue the regular distribution and development model of OpenSolaris (Laishram, 2010). Instead, Oracle would focus its development strategy on a new proprietary version of Solaris called Solaris Express. In effect, the new strategy from Oracle would not allow the community of developers that supported OpenSolaris to continue their work. In response, the CAB of OpenSolaris project decided to fork the project (Shankland, 2010). The resulting fork of the OpenSolaris project was called OpenIndiana, which was created to continue the development and distribution of the OpenSolaris project. Currently, Oracle still continues development on the proprietary Solaris Express operating system, while the community of developers supporting OpenSolaris have left Oracle to work on OpenIndiana.

In the case of the OpenSolaris operating system, Oracle's strategy was simply to discontinue the open source project and focus development on a proprietary version of Solaris under the new name Solaris Express. This represents the most direct strategy for ending open development. Oracle announced that the open source project would be discontinued and, in response, the community had to fork the project to continue development under a new name. This is a similar fate to that of MySQL and OpenOffice, but Oracle's strategy for ending development in each of those projects took different forms.

MySQL

In 2008, Sun Microsystems acquired MySQL AB for approximately US$1 billion (MySQL, 2008). At the time, MySQL was growing in the market for relational database management systems

(RDBMS), and Sun's acquisition of MySQL would allow the company to compete directly with Oracle in that particular market. That year, the RDBMS market was dominated by Oracle, which had a 44.3% share (Olofson, 2008). One year later, when Oracle acquired Sun, the MySQL property was one of the key properties that drew Oracle's interest. In fact, the Sun-Oracle merger was originally approved by regulators in the United States, but the European Union (EU) did not immediately approve the deal, specifically because of concerns that Oracle's acquisition of the MySQL property would lead to an anticompetitive market for RDBMS in Europe (Chapman and Newman, 2009). Consequently, the EU pressured Oracle to divest the MySQL property as a condition for approval of the merger. As leaked documents provided to the whistle-blowing site WikiLeaks have since shown, the United States Department of Justice communicated directly with the European Commission's Directorate General for Competition in support of the merger in October 2009 (United States Mission, 2009). Less than three months later, in December 2009, the merger was approved without the divestiture conditions sought by the EU.

MySQL relied on a dual-licensing approach that was similar to the licensing of OpenSolaris. The dual-license model for MySQL allowed the codebase for MySQL to be protected by the GNU GPL copyleft license, but proprietary versions could be created for enterprises that wanted customized installations. When the Sun-Oracle merger was approved, employees working for MySQL had reservations about Oracle's intentions for the GPL-protected codebase of MySQL (Babcock, 2009). Most notably among them was Michael 'Monty' Widenius who authored the original version of MySQL and co-founded MySQL AB, which was the original owner of MySQL. Widenius sold MySQL AB to Sun before Sun was acquired by Oracle. Widenius, along with other MySQL developers, were concerned that Oracle would try to discontinue MySQL or make it a closed-source program by using the same strategy it had used with OpenSolaris. In response, Widenius urged MySQL users to 'Help MySQL' by starting an online petition under the same name (Widenius, 2009). Leading up to its acquisition of Sun, however, Oracle pledged to keep the same licensing strategies in place that had been negotiated with current customers for an additional five years (Whitney, 2009). That commitment was set to officially expire in December 2014.

Fuelled by concerns about Oracle's intentions for MySQL, the developers forked the project to create MariaDB [9]. The codebase for MariaDB is protected by the GNU GPL, and is designed to be a drop-in replacement for MySQL. As a fork of MySQL, MariaDB allows its community of developers and users to ensure that the code continues to be protected by the GNU GPL regardless of what Oracle decides to do with MySQL. Furthermore, although MySQL remains dominant in the RDBMS market with an approximately 58% market share, MariaDB has now grown to claim approximately 18% of the market (Fydorenchyk, 2014). MariaDB has experienced increased growth in the database market in part because of some notable companies switching from MySQL to MariaDB, including Google and the Wikimedia Foundation.

MariaDB gained additional attention from some of Oracle's competitors who directly invested into the project. Most notably, SkySQL invested nearly US$20 million to support the growth of MariaDB (MariaDB Corporation Ab, 2014). Backed by capital from Intel and other venture capital firms, SkySQL is directed by some of the founding members of MySQL as well as former executives who left the company after Oracle acquired the project. SkySQL also announced a merger with The Monty Program AB, which is led by Monty Widenius. The merger reunites the original members of MySQL and transfers ownership of the MariaDB trademark to SkySQL. The resulting partnership will focus on developing MariaDB to compete with MySQL.

Furthermore, both the Monty Program AB and SkySQL belong to the MariaDB Foundation, which is a non-stock, non-profit corporation established to provide legal and technical support for the MariaDB project and provide a platform for supporters to contribute money to the project. For example, the MariaDB Foundation sells corporate memberships beginning at $50,000 and corporate sponsorships beginning at $5,000. According to the Foundation's website, corporate memberships allow "engagement with the governance of the Foundation", although no further details are provided about exactly what that entails (MariaDB Foundation, 2014).

In sum, MariaDB represents another example of how communities of FLOSS projects maintain the ability to protect their commons-based resource against unwanted corporate influence. In this case, however, Oracle's strategy was not to discontinue the open source project. Rather, Oracle's acquisition of Sun would allow the company to gain a greater market share of the RDBMS market, and Sun's ownership of MySQL was one of the primary properties that attracted Oracle to Sun. Although development of MySQL still continues under Oracle, many of the community members resigned from Sun, and Oracle's commitment to maintain the same licensing agreements for MySQL are set to expire at the end of 2014. To resist what could ultimately be a similar fate to that of OpenSolaris, the MySQL community forked the project to develop MariaDB. MariaDB enjoyed the additional benefit of receiving investment capital from some of Oracle's competitors, which ensures the survival of the project for at least the foreseeable future. By establishing the MariaDB Foundation, the community has a legally recognizable organization to provide technical and legal support for the project, while also collecting additional donations to the project.

StarOffice, OpenOffice, and LibreOffice

During the dot-com bubble in the mid- to late-1990s, Sun Microsystems experienced dramatic growth that allowed the company to make some key acquisitions. One of those acquisitions came in 1999 when Sun acquired a German company called StarDivision, which developed StarOffice (Shankland, 1999). StarOffice was designed as a proprietary office software, which featured word processing, spreadsheet, presentation, drawing, database, and formula programs. When Sun acquired StarDivision, the company continued to develop StarOffice as proprietary software. However, Sun forked the project and relicensed the software so that the source code was open source (Sun Microsystems, 2000). Once again, Sun's strategy was to use the newly open-sourced software, known as OpenOffice, to develop new features and fix bugs in the software. Then, the changes made to OpenOffice could be integrated into StarOffice, which contained certain proprietary elements. OpenOffice could continue to remain free to consumers, while Sun tried to monetize StarOffice by selling the software and services to customers who wanted the additional features. The upshot for Sun was the maintenance and support for essentially two different versions of the same software: OpenOffice 1.0 was a forked version of StarOffice 6.0. Sun maintained the legal rights to both properties, although they were protected by different licenses (Hillesley, 2012).

The early versions of OpenOffice were protected by the Sun Industry Standards Source License (SISSL) and the GNU Lesser General Public License (GNU LGPL). Later versions were protected by an updated version of the LGPL after Sun discontinued the SISSL. The LGPL was chosen because it had less restrictive requirements for integrating free and open source software components into proprietary versions of the software. Although a full discussion of the distinctions between free and open source software licenses is beyond the scope of this paper, the basic differences between the GPL and the GNU LGPL can be summarized briefly [10]. The GPL requires that any modified or derivative software must be redistributed under the same licensing requirements. In effect, any code that incorporates GPL-protected code becomes a derivative work, thereby requiring the code to be protected by the GPL. This ensures that free software remains free software rather than being exploited by commercial companies. The LGPL is a more flexible license that allows free software elements to be linked into code that is protected by other licenses. The only restriction on using LGPL-protected software is that the end-user must have the ability to modify the source code. By protecting OpenOffice in this way, Sun could ensure that developments in OpenOffice could be used in their proprietary StarOffice.

Thus, the symbiotic relationship between StarOffice and OpenOffice continued under Sun because Sun was transparent about its intentions for the two properties. OpenOffice was also governed by a Community Council comprised primarily of members from the broader OpenOffice community, but the board also included a Sun employee. The Sun employee on the Community Council was responsible for communicating Sun's intentions to the community (OpenOffice.org, 2009). Once again, however, this relationship was strained when Oracle acquired Sun in 2010.

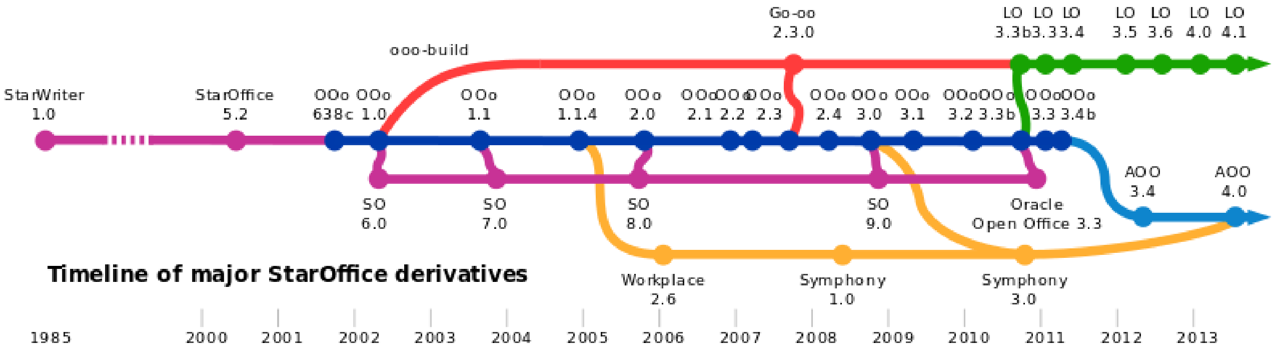
When Oracle discontinued the OpenSolaris operating system, members of the OpenOffice Community Council decided to create The Document Foundation and fork the OpenOffice project under the name LibreOffice until Oracle made its intentions clear for the OpenOffice project (Paul, 2010). In the event that Oracle ultimately decided to discontinue OpenOffice, the Community Council would be able to move development to the newly created LibreOffice. The Document Foundation was established as a non-profit organization to manage the LibreOffice project and promote the use of open-source document software more broadly. The initial governance of The Document Foundation was directed by a temporary steering council featuring some of the same members of the OpenOffice Community Council (DrewJensen, 2010). However, Oracle viewed the Community Council members' involvement on two governing boards as a conflict of interest and asked those members to step down from the Community Council (Apache OpenOffice, 2010). This effectively ended community support for OpenOffice and the project was renamed Oracle OpenOffice, which became the proprietary software offering from Oracle that was meant to replace Sun's StarOffice.

While Oracle cited a conflict of interest, members of the broader open source community viewed Oracle's strategy as simply wanting to discontinue open source projects that existed under Sun because they did not provide any real value to the company (Moody, 2010). In response, The Document Foundation continued its development of LibreOffice. Since LibreOffice had strong community support, the project immediately outpaced the growth of OpenOffice in its initial release by developing features that were not available in OpenOffice (McAllister, 2011). In effect, all of the collective labor behind the development of OpenOffice had abandoned the project but continued its work under the LibreOffice project. Because production on OpenOffice had been abandoned, Oracle announced that it would end development on the project entirely and fire the majority of OpenOffice developers (Byfield, 2011). Ultimately, Oracle donated the codebase for OpenOffice to The Apache Software Foundation, which has since resumed development on the project under the name Apache OpenOffice.

To summarize this somewhat confusing history of software that has been forked numerous times, Figure 1 illustrates the development history of StarOffice, its transition to OpenOffice (OOo) under Sun, the dual development of StarOffice (SO) alongside OpenOffice, the forks into LibreOffice (LO) and Oracle OpenOffice after Oracle acquired Sun in 2010, and the donation of OpenOffice back to The Apache Software Foundation to be developed as Apache OpenOffice

(AOO). Figure 1 also contains additional forked projects that have not been discussed in this article, including IBM Lotus Symphony (Symphony) and Go Open Office (Go-oo). As illustrated in Figure 1, the developments from Go Open Office were ultimately included in LibreOffice and IBM's Symphony was ultimately included in Apache OpenOffice.

Figure 1. Development of StarOffice derivatives



Source: David Gerard, 2013, available at: http://en.wikipedia.org/wiki/File:StarOffice_major_derivatives.svg
Licensed under CC0 1.0 Universal

In sum, the forking of OpenOffice into LibreOffice once again illustrates how members of the broader free and open source software community can protect the products of their collaborative labor against unwelcome influence from corporations. In the case of OpenOffice, Sun Microsystems maintained a good relationship with the community by supporting open-source development and being transparent about their intentions for the project. In addition, Sun was able to facilitate a symbiotic relationship between the open source OpenOffice and its proprietary StarOffice by using a licensing scheme that allowed them to integrate open source components into their proprietary software. Moreover, the OpenOffice project was governed by a Community Council, which included members from Sun who served as liaisons between Sun and the community. When Oracle acquired Sun, the Community Council was effectively disbanded after Oracle cited council members' involvement on LibreOffice steering committee as a conflict of interest.

Defending the commons

Oracle's acquisition of Sun Microsystems and its consequences for the FLOSS community illuminate some of the primary concerns and conflicts lying at the heart of corporate involvement in FLOSS projects. On the one hand, commons-based peer production is made possible by the unique technical, legal, and governance strategies used by FLOSS communities. The technical capability to coordinate, reproduce, and distribute its technologies without any significant cost allow for greater participation in the productive process. As such, FLOSS communities have tremendous potential in their capacity for collaborative and cooperative labor. Further, these communities protect the products of their labor with alternative forms of intellectual property licenses to ensure that their commons-based resources remain freely available to members of the community as well as others interested in accessing or contributing to the projects.

On the other hand, the assignment of intellectual property licenses represents a primary area of struggle. As corporations become involved in FLOSS projects, the codebases of those projects may contain portions of code that are protected by different intellectual property licenses. Indeed, this is one of the common strategies used by corporations that sponsor FLOSS projects. Corporations either design unique intellectual property licenses to assign to the code, which enables the corporation to appropriate the commons-based peer production into their proprietary offerings, or they control the types of licenses that can be assigned to the contributor's code through the use of contributor licensing agreements (CLAs). The CLAs function similarly to End User Licensing Agreements (EULAs) in the sense that they are non-negotiable contracts to which the contributor must agree in order to participate in a corporately sponsored FLOSS project.

To the extent that potential contributors to such projects agree to the terms established by either the sponsoring corporation or the project's governing board, the choice to contribute is based on publicly available and well-documented information. In other words, the relationship between the contributor and the organization governing the project is founded on transparency, which is one of the core tenets of the FLOSS community's moral economy. Once a contributor becomes a member of the project community, however, unforeseen changes to the governance structure of the project can occur in the event of a corporate acquisition like Oracle's acquisition of Sun Microsystems.

To cope with these unforeseen changes, especially when those changes generate unwanted influence in the project, the FLOSS community may choose to fork their project to avoid the unwanted influence. In this sense, forking functions as a tactic used by FLOSS laborers to defend against unwanted corporate appropriation of their collective labor. While the corporation may retain possession of the code that has already been produced, the community also maintains possession of the codebase but with the prospect of continuing development independently of the former corporate sponsor.

Forking a project, however, can be a risky endeavor for at least two reasons. First, once a project is forked, the community risks the possibility of splintering into smaller subgroups, which would dilute its productive capacity and the potential for developing a robust software project. To the extent that the community can collectively agree on a practical solution for redirecting future development, this risk can be diminished. In the case studies discussed here, the community was able to successfully coordinate independent development while still retaining most of its productive capacity. The second major risk of forking, is associated more specifically with the organizational security of having a sponsor. In fact, one definition of success in open source projects is to receive backing from a company. The perception is that a sponsored project is more likely to survive at least into the immediate future if not on a more long-term basis. In turn, sponsored projects are likely to be more attractive to potential contributors who want to see FLOSS projects succeed. When this type of organizational security is lost because of a forked project, FLOSS communities are placed in a somewhat precarious position as digital laborers.

One of the primary strategies used to mitigate the loss or perceived loss of organizational security is the establishment of non-profit organizations. Such organizations provide legally recognizable entities that can more effectively defend the intellectual property rights and licensing requirements of the project. These organizations can also facilitate the transfer of funding from alternative donors or sponsors who wish to contribute to the project directly. Furthermore, from a governance standpoint, the community can reclaim direct governance of the project by democratically electing governing councils made up of community members.

While this article focuses on one example of how FLOSS communities responded to unwanted corporate influence in their projects, further exploration of the dynamics existing between specific FLOSS projects and their sponsoring organizations seems warranted. To that end, at least two primary areas for further research emerge. First, since intellectual property licenses represent a primary area of struggle, additional research might focus on the exact terms of those licenses, especially those authored by corporations that sponsor FLOSS projects. Second, a more thorough understanding of the exact reputational or economic consequences of project forking could assist the development of theoretical frameworks for evaluating the material consequences of forking (for both FLOSS communities the corporations that sponsor FLOSS projects). These additional studies would undoubtedly offer important contributions to our understanding of the complex relationships that exist between corporations and FLOSS communities.

The broader FLOSS community needs to maintain its ability to protect commons-based resources, especially in the face of growing corporate involvement in FLOSS projects. The protection of these resources depends, at least in part, on a shared understanding of how the community can use its collective labor power against unwanted corporate involvement. The lessons learned from Oracle's acquisition of Sun Microsystems offer exemplary cases of effective strategies. Most important, however, is the recognition that the struggles taking place within the FLOSS community are just one part of a broader social struggle. While FLOSS communities represent perhaps the most notable form of commons-based production, a more generalized system of commons-based production will not be possible until we have a commons-based society (Fuchs, 2008). Until that time, commons-based movements like FLOSS may be subjected to increasing corporate encroachment that threatens to abate, assimilate, or altogether annihilate progress toward alternative economic configurations.

Author Bio

Ben Birkinbine is a Visiting Assistant Professor in the Reynolds School of Journalism & Center for Advanced Media Studies at the University of Nevada, Reno. He holds a PhD in Media Studies from the School of Journalism & Communication at the University of Oregon. His research interests are related to the political economy of communication, digital media studies, and open source technology.

Endnotes

[1] More specifically, the kernel manages input/output requests from software and translates those requests into data processing instructions for the hardware components of a computer.

[2] I cannot offer a fully developed account of the debates about immaterial labor and digital labor here, especially because these types of labor take many different forms. For a critique of immaterial labor as an analytical concept, see Sayers (2007).

[3] I use the term 'end product' here to differentiate open source production as a *process* from the *products* that result from such production. However, this is a bit of a misnomer because many (if not most) open source projects hold significant value because they are constantly developing, undergoing revision, and implementing new features or fixes. In this sense, there is no 'end product' unless a project has been discontinued. Rather, open source projects are continuously changing or adapting to the unique needs of users within the community.

[4] Although not entirely detailed, Wikipedia contains a page that compares the general terms of various free and open source software licenses: http://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses

[5] For example, Red Hat, one of the largest and only publicly traded firms dealing almost solely in free software, once had contributors sign the Individual Contributor Licensing Agreement (ICLA), whereby Red Hat was granted the intellectual property rights of the contributors. The ICLA was replaced by the Fedora Project Contributor Agreement (FPCA), which stipulated which types of intellectual property licenses could be assigned to a programmer's contributions. Information about these two licenses can be found on the Fedora Project wiki: http://fedoraproject.org/wiki/Legal:Licenses/CLA and http://fedoraproject.org/wiki/Legal:Fedora_Project_Contributor_Agreement

[6] The information about Oracle comes from its 2013 Form 10-K Annual Report, filed with the Securities and Exchange Commission (SEC) in the United States unless otherwise noted.

[7] The information about Sun Microsystems comes from its 2009 Form 10-K Annual Report, filed with the Securities and Exchange Commission (SEC) in the United States unless otherwise noted.

[8] The full details of the Common Development and Distribution License (CDDL) cannot be provided here. A full-text version of the license is available from the Open Source Initiative at: http://opensource.org/licenses/CDDL-1.0

[9] MariaDB is just one fork of the MySQL project. Others include Drizzle and Percona Server.

[10] The full text of the GNU GPL is available at: http://www.gnu.org/copyleft/gpl.html and the LGPL is available at: http://www.gnu.org/licenses/lgpl.html

References

Apache OpenOffice Wiki (2010, October 14) Community council log 20101014. Available at: http://wiki.openoffice.org/wiki/Community_Council_Log_20101014 (accessed 4 September 2014)

Andrejevic M (2012) Exploitation in the data mine. In: Fuchs C, Boersma K, Albrechtslund A and Sandoval M (Eds) *Internet and surveillance: The challenges of Web 2.0 and social media*. New York: Routledge, pp. 71–88.

Babcock C (2009) MySQL users wary on Oracle acquisition. *InformationWeek*, 21 April. Available at: http://www.informationweek.com/mysql-users-wary-on-oracle-acquisition/d/d-id/1078810? (accessed 4 November 2014)

Benkler Y (2006) *The wealth of networks: How social production transforms markets and freedom*. New Haven, CT: Yale University Press.

Byfield B (2011) Arguments over the future of OpenOffice.org. *Datamation*, 2 June. Available at: http://www.datamation.com/osrc/article.php/3935001/Arguments-Over-the-Future-of-OpenOfficeorg.htm (accessed 4 November 2014)

Chapman P and Newman M (2009) Oracle faces in-depth EU probe over Sun purchase (update 2). *Bloomberg*, 3 September. Available at: http://www.bloomberg.com/apps/news?pid=newsarchive&sid=aCWYuHl5bHC8 (accessed 5 September 2014)

Coleman G (2013) *Coding freedom: The ethics and aesthetics of hacking*. Princeton, NJ: Princeton University Press.

DrewJensen (2010) Document Foundation formed. *Apache OpenOffice forum*, 28 September. Available at: https://forum.openoffice.org/en/forum/viewtopic.php?t=34510 (accessed 4 November 2014)

Free Software Foundation (FSF) (2014) Free software is a matter of liberty, not price. Available at: http://www.fsf.org/about (accessed 4 November 2014)

Fuchs C (2008) *Internet and society: Social theory in the information age*. New York: Routledge.

Fuchs C (2011) New media, web 2.0 and surveillance. *Sociology Compass* 5(2): 134–147.

Fuchs C (2012) *Critique of the political economy of web 2.0 surveillance*. In: Fuchs C, Boersma K, Albrechtslund A and Sandoval M (Eds), *Internet and surveillance: The challenges of Web 2.0 and social media*. New York: Routledge, pp. 31–70.

Fydorenchyk T (2014) Software stacks market share: January 2014. *Jelastic*, February 6. Available at: http://blog.jelastic.com/2014/02/06/software-stacks-market-share-january-2014/ (accessed 5 September 2014)

Hardt M and Negri A (2000) *Empire*. Cambridge, MA: Harvard University Press.

Hardt M and Negri A (2005) *Multitude: War and democracy in the age of empire*. New York, NY: Penguin Books.

Hillesley R (2012) Open-source development: The history of OpenOffice shows why licensing matters. *TechRepublic*, 1 October. Available at: http://www.techrepublic.com/blog/european-technology/open-source-development-the-history-of-openoffice-shows-why-licensing-matters/ (accessed 4 November 2014)

Laishram R (2010) Oracle has killed OpenSolaris. *Techie Buzz*, 14 August. Available at: http://techie-buzz.com/foss/oracle-has-killed-opensolaris.html (accessed 5 September 2014)

Lazzarato M (1996) Immaterial labor. In: Virno P and Hardt M (Eds), *Radical thought in Italy: A potential politics*. Minneapolis, MN: University of Minnesota Press, pp. 133–150.

MariaDB Corporation Ab (2014) Intel Capital leads $20 million investment in SkySQL to grow MariaDB. Press release. Available at: https://mariadb.com/news-events/press-releases/intel-capital-leads-20-million-investment-skysql-grow-mariadb (accessed 4 November 2014)

MariaDB Foundation (2014) About the MariaDB Foundation. Available at: http://mariadb.org/en/foundation/ (accessed 5 September 2014)

McAllister N (2011) Open office dilemma: OpenOffice.org vs. LibreOffice. *Infoworld*, 16 February. Available at: http://www.infoworld.com/article/2623185/office-software/open-office-dilemma--openoffice-org-vs--libreoffice.html?page=1 (accessed 4 November 2014)

Meehan E (2005) *Why TV isn't our fault*. Lanham, MD: Rowman & Littlefield Publishers.

Moody G (2010) Has Oracle been a disaster for Sun's open source? *The H Online*, 28 June. Available at: http://www.h-online.com/open/features/Has-Oracle-been-a-disaster-for-Sun-s-open-source-1030066.html (accessed 4 November 2014)

Moody G (2001) *Rebel code: Linux and the open source revolution*. Cambridge, MA: Perseus Publishers.

Mosco V (2009) *The political economy of communication*, 2nd ed. London: SAGE.

Murdock G and Golding P (1973) For a political economy of mass communications. *The Socialist Register* 10: 205–234.

MySQL (2008) Sun to acquire MySQL. Press release. Available at: http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html (accessed 5 September 2014)

Olofson CW (2008) Worldwide relational database management systems 2007 vendor shares. *IDC.com*. Available at: http://download.microsoft.com/download/6/C/D/6CDC123B-A159-438C-BBAD-7750EE5D0D67/IDC-Server Workloads Forecast.pdf (accessed 4 November 2014)

OpenOffice.org (2009, March 25) Community council charter. Available at: http://web.archive.org/web/20111013035757/http://council.openoffice.org/councilcharter12.html (accessed 4 November 2014)

Oracle Corporation (2013) Annual report Form 10-K. Available at: http://oracle.q4cdn.com/e8a91886-38c1-427e-a6c5-a32536c1fa4e.pdf (accessed 5 September 2014)

Paul R (2010) Document Foundation forks OpenOffice.org, liberates it from Oracle. *ArsTechnica*, 28 September. Available at: http://arstechnica.com/information-technology/2010/09/document-foundation-forks-openofficeorg-to-liberate-it-from-oracle/ (accessed 4 November 2014)

Santos C, Kuk G, Kon F, and Pearson J (2012) The attraction of contributors in free and open source software projects. *Journal of Strategic Information Systems* 22(1): 26–45. Available at http://dx.doi.org/10.1016/j.jsis.2012.07.004 (accessed 11 November 2014)

Sayers S (2007) The concept of labor: Marx and his critics. *Science & Society* 71(4): 431–454.

Scholz T (Ed) 2013 *Digital labor: The internet as playground and factory*. New York: Routledge.

Shankland S (2010) Lacking Oracle help, OpenSolaris group disbands. *CNET.com*, 23 August. Available at: http://www.cnet.com/news/lacking-oracle-help-opensolaris-group-disbands/ (accessed 4 November 2014)

Shankland S (1999) Sun shelled out $73.5 million for Star Division. *CNET.com*, 9 November. Available at: http://news.cnet.com/Sun-shelled-out-73.5-million-for-Star-Division/2100-1001_3-232561.html (accessed 4 November 2014)

Smythe DW (1960) On the political economy of communications. *Journalism & Mass Communication Quarterly* 37(4): 563–572.

Sun Microsystems (2009) Annual report Form 10-K. Available at: http://www.sec.gov/Archives/edgar/data/709519/000119312509183962/d10k.htm (accessed 5 September 2014)

Sun Microsystems (2005) OpenSolaris community advisory board formed. Available at: http://www.prnewswire.com/news-releases/opensolaris-community-advisory-board-formed-54181347.html (accessed 4 November 2014)

Sun Microsystems (2000, July 19) Sun Microsystems open sources StarOffice technology. Press release. Available at: http://www.openoffice.org/press/sun_release.html (accessed 4 November 2014)

Terranova T (2004) *Network culture: Politics for the information age*. Ann Arbor, MI: Pluto Press.

The Linux Foundation (2013) *Linux kernel development: How fast it is going, who is doing it, what they are doing, and who is sponsoring it*. Available at: http://www.linuxfoundation.org/publications/linux-foundation/who-writes-linux-2013 (accessed 5 September 2014)

Thompson EP (1971) The moral economy of the English crowd in the eighteenth century. *Past & Present* 50(1): 76–136.

United States Mission to European Union. (2009, October 27). Oracle concerned over EU investigation of Sun merger. *WikiLeaks*. WikiLeaks cable: 09BRUSSELS1455. Available at: http://wikileaks.org/cable/2009/10/09BRUSSELS1455.html (accessed 5 September 2014)

Wasko J (2001) *Understanding Disney: The manufacture of fantasy*. Cambridge: Polity.

Whitney L (2009) Oracle pledges to play well with MySQL. *CNET.com*, 14 December. Available at: http://news.cnet.com/8301-1001_3-10414686-92.html (accessed 5 September 2014)

Widenius M (2009, December 12) Help saving MySQL. Available at: http://monty-says.blogspot.com/2009/12/help-saving-mysql.html (accessed 4 November 2014)