

## **The Open-source Revolution as a Badiouian Event**

**Daryl Campbell**

**Abstract:**

This paper will show how the Free/Libre/Open Source Software movement (FLOSS), or the Open Source revolution, constitutes an event in the Badiouian sense. Surprisingly, this phenomenon has not yet been connected to Alain Badiou in the literature. I will analyze two salient aspects of FLOSS, thereby establishing the latter's correspondence with Badiou's conception of the event. The General Public License (GPL), as one of these aspects, uncovers in copyright law a usage that, while consistent with existing copyright law, nonetheless contains implications that break with the latter in significant ways. The second aspect is the FLOSS governance model, which introduces a break within the existing norms of managerial driven governance in the dominant capitalist world order. These two aspects can be seen as exceptions within the existing situation from which they arise. Badiou's doctrine of the event admits events as being materially tied to their existing situations even while constituting a break from those very situations. FLOSS will be interpreted through both Badiou's mathematical ontology as well as his more recent logical phenomenology. This argument will be used to challenge the critics who deny fundamental change by maintaining that FLOSS is an outgrowth of, or a phenomenon that is merely dependent upon, the capitalist system. If ownership of property and relations between capitalists and workers structure the capitalist scene and by extension contemporary politics, then weighing the consequences of this change can have implications for politics in general. The paper will conclude with reflections upon these implications.

“It seems to me that we are obliged, at least for the moment (I also don't wish to anticipate the course of things), to consider the consequences of that which is given as a local difference, that is, to think and to act on one point or, at most, a few.” (Bruno Bosteels 2011, 326)

“Revolutionaries aren't born. Revolutions can't be planned. Revolutions can't be managed.

Revolutions happen. . . .” (Linus Torvalds and David Diamond 2001, x)

## **Introduction**

Alain Badiou has proposed a concept of the event which supports the thought of radical change. Other than Badiou's own examples we have an absence of concrete cases upon which to reflect (Badiou 2011: 81). Given the teeming expanse of the void, a source for generating events, it is unthinkable that so few events should be discovered (Badiou, 2005b: 74). This paper will test a point of disturbance in the world of commercial software development by the free/libre/open-source software (FLOSS) development model, otherwise labelled the Open-source Revolution (DiBona et al., 1999), as a case of a Badiouian event. This concrete investigation examines the intellectual property rights and organizational governance model that supports a predominately voluntary, distributed community-managed mode of software development. It is here argued that open-source should be added to the registry of those rare Badiouian events that occur in localized settings but which offer infinite possibilities for that world.

To equip ourselves for this investigation, we will first establish Badiou's conception of an event factoring in both his ontological constructs and his more recently developed logic that accounts for being's appearing. An event is conceived of as an exception to a situation, or world, phenomenologically speaking, and our investigation examines the exception open-source introduces into the world of commercial software development.<sup>1</sup> To establish an event's existence requires an examination of the moment in time when the event disrupts a situation. The exposure of an event into a world happens at an event site, and so we retrospectively dig through the accounts of the appearing of open-source on the corporate scene. This study then turns to some demonstrations for how open-source can be thought as an event and finishes with some implications.

## **The Badiouian Event**

As a strict definition, an event in Alain Badiou's conceptual development “is a one-multiple made up of, on the one hand, all the multiples which belong to its site, and on the other hand, the event itself” (2005a: 179). Parsing this Badiouian concept requires becoming comfortable with some basics for thinking about multiples. A multiple when stripped of its qualifying determinations is at base a set composed of elements. To be considered an element means that what the element is made of is counted as a one. The presentation of the element in a multiple provides the sense of what it means to belong. Presentation reveals what belongs. Although an element can be counted as a one, an element in-itself is a multiple, the one is merely an effect of counting. Therefore, any dissemination of a situation at its ontological level shows itself to be a multiple composed of elements which are themselves multiples leaving any situation composed of multiplicities. In a normal situation, where everything is presented, the composition at sub-levels of this multiple-of-multiples is clearly visible, but events are not normal.

An event disrupts a normal situation by not revealing the elements that make up its composition (Badiou, 2005a: 181). In accordance with the second half of the definition of an event, all that is presented is the presentation of the event itself. This one-multiple of an event shows up in a situation at a localized place which is called the event site. The event site is composed of a multiple, but the elements beneath the surface of this site are not presented. These elements of the event are not detected as belonging to the pre-event situation. What the event site does present is a signifier that is itself “the belonging of the signifier of the event to its signification” (Badiou, 2005a: 189). It names itself and this name appears to belong to the event site multiple. There is the appearance of the event site being plucked from the void. This parallels a standard problem posed when infiltrating an Italian Mafia, you have to have someone that is known to vouch for you. An event site only has itself to vouch for it, a successful infiltration. The normal case for entry into the Mafia is that everything about you is known. Events fit the profile of other multiples but they do not expose their elements at their event site, and they have only their self-declaration to show for itself.

With Badiou's development of a logic for the phenomenal appearing of being, events take on another characterization. The same situation can be thought from its compositional base as a multiple, or it can be thought as to how the elements are related to each other, their world order. World and situation are interchangeable depending on the level of analysis. The established world order, when thought relationally with Badiou's logic of the phenomenon, is the ordered system that maps all the relational degrees of

## Daryl Campbell

difference and identity (2011: 48-49). To exist in this world is to be brought into relation with other objects. With an event there is an object, that previously did not have the requisite relation to other objects in the established world, which transitions from not existing, its status was inexistent, to becoming maximally exposed in the new world order.

Badiou provides three principles to guide an event investigator: 1) an event must be situated in an existing situation thus requiring an event site; 2) an event is self-reflexive, it illegitimately declares itself, violating rules of logic, and is thus forced to disappear into the background; 3) an event is a maximal exposure of its inexistent elements, first for the duration of the event site showing its power, and then through the implication of consequences showing its force (Bruno Bosteels, 2011: 306; Badiou, 2011: 80). Each unique event will bring its own set of expressions, or directives, as Badiou says: “The directives express the way in which the principles, which are largely invariant, might become active in a situation” (Bosteels, 2011: 330). The challenge for this intervention is to detect the specific directives of the open-source revolution and how they reflect the principles of an event.

Expanding on these principles, we first note that what will be required for an event to appear in a particular scene, so it is not considered other-wordly, is an event site materially grounded to a specific situation. The event site is the localization where the event self-declares that what is new to the situation is a multiple with itself being that multiple. Encountering a multiple of this type which only presents itself, I am a multiple which only presents myself, violates the laws of belonging. A set cannot belong to itself. The bowl of fruit on the kitchen table cannot itself be counted as a member of the bowl of fruit. “Self-belonging annuls itself as soon as it is forced, as soon as it happens. A site is a vanishing term: it appears only in order to disappear” (Badiou, 2009: 391). When thought from the established situation, event sites, that appear, are forced to disappear shortly thereafter collapsing back into their unstable foundation. The event site cannot initially present its elements to the world it is exposed to because these elements cannot be brought into any relation with the existing world. At best there is the blinding flash of a lightning strike, the event site, which provides a momentary glimpse of something new.

When a world experiences radical change in Badiou's model it goes through a process where something that did not exist, the inexistent, is made to appear maximally. Anything less than maximal cannot be defended as evental change. Badiou attributes the power of the event to this maximal exposure of an event site (2009: 374). The event site is short lived because of its self declaration but this leaves the elements that make up the site to be worked into valid recognized positions in a revised world order.<sup>2</sup> Any

lasting investigation into the transforming effect of an eventual change on the existing situation will speak to the force of the event (Badiou, 2009: 374). There is the immediate and the subsequent sustained maximal intensity of that which was undetectable prior to the event site.

### **GNU/Linux Phenomenon**

Our account begins by examining the world of hacker culture with FLOSS<sup>3</sup> designating the core values. We look at the two elements that politically charge this study, in their own right introducing an eventual impact on hacker culture: the General Public License (GPL) and community-managed organizational governance. The phenomenon identified below as GNU/Linux<sup>4</sup> is the reference case demonstrating these two elements in an unadulterated state. The account of the GPL and Linux, with its community-managed governance, is the pre-history to the actual event which is the primary focus of this paper.

The beginnings of the GNU/Linux phenomenon are traced back to the experience, in the early 1980s, of Richard Stallman who was working in the MIT Artificial Intelligence Lab (Sam Williams, 2002: 1-12). The early stages of the development of computer software, with a culture of sharing software, was giving way to the commercialization of software by companies with universities following suit claiming ownership of software developed by their staff (Peter Wayner, 2000: 78-79). Stallman felt growing resistance to the status quo that companies projected: he objected to their “natural right to own software,” the notion “that we computer users should not care what kind of society we are allowed to have,” and “that we would have no usable software . . . if we did not offer a company power over the users of the program” (Richard Stallman in Chris DiBona et al., 1999: 54). Companies had caught on quickly that the source code, the human readable version of computer programs, could be locked away with users paying for the licensed executable program, the machine readable form. This restricted who could modify the code.

Thus, in 1985, Stallman published his, GNU Manifesto (1985), which initiated the revolution. Stallman attempted to capture the guiding principles for an alternative software distribution model, and his manifesto “laid out Stallman's plan for creating a virtual commune where people would be free to use the software” (Wayner, 2000: 80). The manifesto clarified the freedoms intended to govern this alternative software development model:

The freedom to run the program, for any purpose (freedom 0).

The freedom to study how the program works, and adapt it to your needs (freedom 1).

## Daryl Campbell

The freedom to redistribute copies so you can help your neighbour (freedom 2).

The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). (Wayner, 2000: 84)

Elsewhere, Stallman clarifies that the freedom he is promoting “is not against business” nor the selling of software (Stallman in DiBona et al., 1999: 61). He is advocating freedom from the encumbrances restricting the sharing of source code that the standard use of copyright protection promotes. To break with the proprietary licensing schemes meant developing a new license, the GNU General Public License (GPL).

Stallman's insight, with the GPL, is to reverse the effects of license provisions built on copyright. “[I]nstead of a means of privatizing software, it becomes a means of keeping software free” (Stallman in DiBona *et al.*, 1999: 59). Relying on copyright law, the GPL leverages copyright to recognize the property protections for the software creator, including the previous unrecognized right of a license to protect the freedom for software to remain accessible:

Part of the GNU GPL's effect is therefore to ensure that the source is always available for a program. Moreover, the GNU GPL also ensures that such improvements must be freely available, along with their source code, to enable the community as a whole to benefit from the collective advances of all users. (Glyn Moody, 2001: 27)

The name copyleft communicates this reversing of the normal application of rights:

The central idea of copyleft is that we give everyone permission to run the program, copy the program, modify the program, and distribute modified versions - but not permission to add restrictions of their own. Thus, the crucial freedoms that define ‘free software’ are guaranteed to everyone who has a copy; they become inalienable rights. (Stallman in DiBona et al., 1999: 59)

Partially implied in Stallman's explanation is that property rights under traditional copyright interpretation are in fact the right to exclude others (Eben Moglen, 1999: unpaginated). With the GPL, a software developer has a mechanism to enforce a right that will ensure that others are not excluded. The objective is not to restrict use but to ensure that any future use is not restricted. The GPL leverages the protection found in copyright law to propagate to derivative works the same terms as the original. It is the viral effect that the GPL has when it comes to derivative works that Stallman, fighting for the preservation of the developers’ freedom to protect their software from future re-appropriation, was seeking to address:

Anyone could freely modify and redistribute such software, or sell it, subject only to the restriction that he not try to reduce the rights of others to whom he passed it along. In this way free software could become a self-organizing project, in which no innovation would be lost through proprietary exercises of rights. . . . Section 2(b) of the GPL is sometimes called ‘restrictive’, but its intention is liberating. It creates a commons, to which anyone may add but from which no one may subtract. (Moglen 1999: unpaginated)

This further identifies the intent behind the GPL and the covert means that Stallman deploys to lodge within the copyright system a means to protect from the normalized use of copyright. "The GPL employs copyright to suspend the usual operation of copyright within the domain of F/OSS development" (McGowan 2005: 363). Through the inherent logic of copyright, the normative effect of copyright is cancelled at the same time as it provides the support for copyleft.

Completion of Stallman's vision of a full-featured operating system stalled despite his efforts with his non-profit Free Software Foundation at creating a legal framework receptive to community participation (Stallman in DiBona *et al.*, 1999: 65). The requisite bundle of supporting utilities and development tools were in place, but the central program, the kernel which directly manages systems resources, was missing. The unexpected development that would complete the vision happened in the early 1990s. University of Helsinki student Linus Torvalds had been working with Andrew Tannenbaum's Minix kernel for use with Intel x86 processors found in commodity PCs (Torvalds and Diamond, 2001). Linus decided to build from scratch, as a hobby project, his own kernel targeted for commodity hardware to work around shortcomings in Minix.

The precipitous move is a post that Linus makes to the Minix Usenet discussion group. His first post in August 1991 announced his intentions:

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) . . . I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :- ) (Clay Shirky, 2008: 238)

The second post in October 1991 announced the first release of the source code of Linus's Minix-like kernel:

I'm working on a free version of a Minix look-alike for AT-386 computers. It has finally reached the stage where it's even usable (though it may not be, depending on what you want), and I am willing to put out the sources for wider distribution. . . . This is a program for hackers by a hacker. I've enjoyed doing it, and somebody might enjoy looking at it and even modifying it for their own needs. It is still small enough to understand, use and modify, and I'm looking forward to any comments you might have.

I'm also interested in hearing from anybody who has written any utilities/library functions for minix. If your efforts are freely distributable (under copyright or even public domain) I'd like to hear from you so I can add them to the system. (Steven Weber, 2004: 54-55)

A new software development approach is inaugurated. The appeal is for the community to define what they want and to contribute to the process. Linus' invitation is as hacker to hacker, worker to worker. The

## Daryl Campbell

users of the system are the contributors and owners of the system. The source code contributed is posted on the Internet for open access and with a licensing agreement that will permit further unrestricted access. The intent is that others will modify to suit their needs. Lastly, Linus continues to reserve some role as final arbiter for ongoing contributions. This move is facilitated by seamless communication in an Internet-based world and by the spirit with which Linus, as a benevolent dictator, manages to negotiate the surge and growth of thousands of programmers around the world who eventually contribute to his project. Combining Stallman and Linus's work culminates in an operating system, GNU/Linux, the product of community development with a licensing structure that obstructs proprietary appropriation for the purpose of exclusive access.

Linus himself connects the success of Linux to more than just code contributions. The power of Linux is as much about the community of cooperation behind it as the code itself. "If Linux were hijacked --if someone attempted to make and distribute a proprietary version -- the appeal of Linux, which is essentially the open-source development model, would be lost for that proprietary version" (Torvalds and Diamond, 1999: 109). The license ensures the continuity of project even if the leader disappears. Such continuity has been confirmed with projects like Debian, a Linux distribution, which has had multiple turnovers in leadership (Siobhán O'Mahony, 2007).

The governance structure of this community is of equal significance in the study of this phenomenon. The unique governance model operates by inclusion of developers distributed world-wide, not under the umbrella of any single organization. Entrance into the community is marked by openness: gate-keeping restrictions are removed or reduced, and there is neither long-term contracts nor an obligation to contribute (Benoît Demil and Xavier Lecocq, 2006: 1454-1456). The removal of barriers to community entrance is referred to as a practice of nonexclusion. "Open-source software is aggressively nonexcludable at several different levels: in fact, that is its major differentiating characteristic" (Weber 2004: 133). The removal of barriers of this type has been identified as part of the motivational appeal to joining FLOSS communities (Shah 2006: 1009-1010). This is not to say there is a complete absence of roles and hierarchy within this community model, only that the importance given to these factors approaches the minimal levels bearable while still producing quality software.

Closely associated with this removal of barriers is a practice of self-selection for the direction of member's contributions. From the beginning the assignment of work was self directed: "he [Linus] never orders anyone to do anything and even his suggestions are mild-mannered. Typical suggestions are of the form:

'hint, hint, Linus wants to get out of doing it himself ;^)'” (Jae Yun Moon and Lee Sproull, 2000: 6). Steven Weber speaking about the voluntary nature of both participation and selection of work tasks summarizes thus:

“What makes it different from the theoretical option of exit from a corporate organization is this: Each person is free to choose what he wishes to work on or to contribute. There is no consciously organized or enforced division of labor. In fact the underlying notion of a division of labor doesn't fit the open-source process at all.” (2004: 62)

At a micro-political level of governance where we are dealing with relations and interests of individuals, the Linux community introduced patterns of governance that were mostly unforeseen in large-scale operations. The norms in this community of practice have a pattern that encourages the contributions of individuals to occur in volume. Open access to the code along with Internet-based forms of communication, such as mailing lists, creates an environment of disclosure. Lee and Cole quote Linus: “The point of open development is that people see what's going on” (2003: 639).

Lee and Cole's analysis is from a study of five years of reviewing the Linux mailing lists in an investigation of the community's practice. Their analysis concludes that the knowledge creation development of this community in its adherence to open principles functions generatively to produce norms and structures (2003). The openness, fostered by the nature of Internet communication, translates into an environment where flaming, critiquing technical considerations, is an accepted and common occurrence. One may exercise some flaming, but there is no threat of dismissal. As a cultivated norm this open sharing approach is strongly reliant on a peer-review process where error correction spawned from criticism is highly valued.

A firm-based approach typically opts for nondisclosure, and authority in the hierarchy tends to halt criticism and critical evaluation. The favouring of criticism and critical evaluation that is intimately tied to the transparency of the open approach is identified as an error correction mechanism: “Logical arguments made on technical grounds are the primary currency of debate” (Weber 2004: 164). Eric Raymond coined the popular phrase “with more eyeballs all bugs become shallow” (2000: 8) legislating it as Linus's Law. The phrase refers to debugging programs, but it represents an inherent aspect of the way that the Linux community harvests the contributions from the whole community in creating a mass corrective force.

## Event Site

Switching context from the world of hacker culture to the commercial world of software development is where I claim an event has occurred. Badiou acknowledges that elements from one world can come to be exposed in another world (2005a: 178; 2011: 54). An account of what can be said about the laws governing the pre-existing state-of-the-situation frames a before and after snapshot:

In 1990 it would have been easy to predict the demise of the Unix world, the full marginalization of the Free Software Foundation, and the decisive triumph of the proprietary software development model exemplified by Microsoft. The prediction would have rested on an argument about basic forces of industrial organization. It would claim the source code, the key knowledge asset for a software company, is created most effectively in a market setting when innovators capture directly the rents associated with their innovation. The proprietary software development model maximizes the incentives to innovate because it can generate those rents by keeping source code secret. Proprietary software, then, is the equilibrium result of fundamental economic forces, not historical accident, policy choices, or something else. (Weber 2004: 94-95)

What this background emphasizes is the perceived necessity of keeping source code secret for the purposes of rent acquisitions. Implied is that this development has to occur within an organization under the guidance of a managerial led hierarchical structure. Against this background the event will be traced out.

To happen within this commercial world the event requires an event site location. “[T]he shot heard 'round the world' [...]” (Raymond in DiBona *et al.*, 1999: 210) is the announcement from Netscape on January 22nd, 1998, that they plan to give away their browser and release the source code. “For the first time in the history of the hacker culture, a Fortune 500 darling of Wall Street had bet its future on the belief that our way [community-managed governance] was right” (Raymond in DiBona *et al.*, 1999: 210). We view this as the opening of the event site because there is a reverberation in the corporate world as initiated by a Fortune 500 company. Netscape's CEO, Jim Barksdale, had attributed this decision to the influence of Eric Raymond's “The Cathedral and the Bazaar,” as Raymond recounts it (in DiBona *et al.*, 1999: 210). This short paper attempted to describe the hacker culture that produced Linux and how this approach, community-managed governance, could prove superior to previous strategies.

Coinciding with a visit to Netscape to consult on a licensing strategy, Raymond and a group of developers from the hacker community met, on February 3, 1998, to develop a strategy for a marketing campaign to the corporate world, coining the term open-source (Raymond in DiBona *et al.*, 1999: 211). Linux would provide the reference case to convince senior IT managers that this model would work, and Netscape

would be the first Fortune 500 to go all in on the model. Their strategy was to mount a crusade bombarding Wall Street and the Fortune 500 with the message that open-source was ready for corporate adoption. This would correct what was thought to have stood in the way of success:

It seemed clear to us in retrospect that the term "free software" had done our movement tremendous damage over the years. [. . .] Most of it came from something worse -- the strong association of the term "free software" with hostility to intellectual property rights, communism, and other ideas hardly likely to endear themselves to an MIS manager. (Raymond in DiBona *et al.*, 1999: 212)

Raymond and the group, at that time, formed a non-profit organization, Open-source Initiative (OSI), with the purpose of seizing the opportunity to consolidate the movement. The organization developed the Open-source Definition, a specification for what certifies a license as open-source ("The Open-source Definition": unpaginated). The definition preserved the guarantee of rights to access the source code and subsequent rights to alter and redistribute the program. The GPL's copyleft provisions were accommodated by a statement on derived works: "The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software" (Bruce Perens in DiBona *et al.*, 1999: 177). However, it is now only a requirement to permit the perpetuation of the same license for future modifications with each particular license setting out the conditions. No discrimination is made against open-sources licenses that let the licensee take future modifications private. With this specification, the obstacle to acceptance from commercial interests was largely removed as the name open-source did not specify whether future modifications could be privatized.

On March 31, 1998, Netscape threw a party and released the code to their Mozilla browser. The code was released under a newly minted license, the Netscape Public License (NPL), which attempted to straddle the difference between the GPL license, with its viral nature, and other licenses that risked modifications not being returned to the community (Jim Hamerly et al. in DiBona *et al.*, 1999: 200-201). This marks the closing of the event site because the full measure of what FLOSS could introduce had been masked with the name open-source and a license softened from the terms of the GPL. Although the event site has disappeared the event itself continues on.

Open-source circulates in the commercial world as the signifier of revolutionary change. It introduces the major conceptual aspects of the FLOSS movement, while not being exhaustive, and at the same time avoiding the full experience of the event and thus radical change. Open-source remains the surface referent to a revolutionary movement with GNU/Linux functioning as an existence proof. Open-source

Revolution names the event providing the reference when thought from the world of commercial development, but FLOSS designates the full exposure to the elements that make up the event. The GPL remains the real support for radical change which signals for contributors that a community-managed governance model is desired and now confirmed possible. If only understood as open-source, the changes brought into this world are below the threshold of evental status which would leave the existing situation fundamentally unchanged.

### **Badiouian Reading of the Open-source Revolution**

To complete the investigation requires bringing into relation the concrete case of open-source with Badiou's conception of an event. Verification involves stepping through Badiou's three principles to see how this event fits with the profile. Is the open-source event situated in an established world? Does it demonstrate the symptoms of an event site in this world? Do the FLOSS qualities reach maximal intensity?

The open-source event is situated in this case by being a deliberate strategy by a Fortune 500 company making it the cornerstone of their business practice, materially linking it to this world. Commercial viability had been demonstrated with Red Hat offering a service model built around GNU/Linux operating system (DiBona *et al.*, 1999: 113-125), but Netscape's move was a first attempt by a major player to release a major commercially developed software project as open-source. The OSI, a component of the event site, was a vehicle established for the purpose of presenting a “new rhetoric of pragmatism and market-friendliness” to the corporate world (“History of the OSI”; unpaginated). The GPL, the license marked with the greatest demarcation, was itself not the annulment of intellectual property rights. Making this event doubly situated, the GPL is rooted in copyright law: “Open-source is built upon a foundation of intellectual property law, particularly copyright law” (Lawrence Rosen 2005: xx).

The event site location, having been identified in Netscape's 1998 announcement and subsequent release of Mozilla source code wedged around the formation of the OSI, further confirms that an event site status has attained through the self-declaration that the new member belongs. If Netscape had explained a new engineering approach that brought about the development of the Mozilla browser, there would be no need for any self-declaration since a demonstration proof could have been provided. Instead what followed was a marketing campaign putting into circulation the language of open-source which provided the veneer

cover to the FLOSS movement. Ironically, the naming of the movement comes from those inside the movement aligned with the sympathies in the FLOSS movement who are trying to intentionally conceal some elements of the movement. Although Badiou has reconsidered the role of naming (2009: 361), the name open-source plays on the open nature of the movement without exposing the more disruptive aspects. Open expresses the unbinding property of any exposure to the void making this event also reflexive to Badiou's conceptual development (2005b: 72-73).

Self-belonging produces a merely superficial exposure of the event for the existing situation. The event site flashes its brilliance exposing a supposed impossible<sup>5</sup> alteration to the situation but the conditioning from within the existing situation blinds the observer. “[T]he laws of being close up on that which will have violated them for a flash of time” (Badiou, 2009: 368). Open access to the source code becomes the invariant across the array of licenses that are developed. However, the elements beneath the label and their effect remain obscured. No greater illustration exists than to peruse the sixty-nine current licenses that have obtained OSI compliance and then attempt to sort out their differences (“Licenses by Name”: unpaginated). The differentiation between the GPL license and less restrictive licenses disappears into the background.

This event site closed when Netscape released their Mozilla source code with their own NPL open-source license, one which did signal to developers that future modifications would not be taken private but which allows the mingling of private code with open code. The full effect of the type of freedom the GPL offers to a community of developers is masked over in the founding event site. The difficulty with Raymond's advocacy is that by only stressing the community approach to governance he lost the symbiotic relationship the license provided. The open qualifier, applied to a new production process, is even more obscurely masked for corporate managers. “It took a long time before the project [Mozilla] truly lived up to its open-source promise. There was a camp of Netscape insiders who would not accept small patches from outsiders” (Torvalds and Diamond, 2001: 231). Netscape signalled they would only selectively tap into aspects of what defined the community-managed phenomenon.

Badiou's recent shift to include the phenomenological account of an event stresses the change from the in-existent's pre-event status (Badiou, 2009). With the open-source event the in-existent element was the unforeseen space in copyright law that created room for a new thinking about intellectual property. By extension this in-existent element in copyright law fostered altered states of governance exposing in turn in-existent practices in organizational formations. For the brief window of the event site the rearranged

## Daryl Campbell

situation with an alternate software license scheme and a different model of organizational governance appears infinitely unlimited, a testament to the inexistent's power. With Netscape nailing down the NPL license, the process of reining in possibilities begins. In the afterglow of the event site the situation moves into a state of methodically testing out and experimenting with arrangements that will maintain a loyalty to a full expression of FLOSS.

The OSI initiative led to grouping all licenses under one category. However, if the licenses at the less-restrictive side of the spectrum, non-copyleft, are attesting to the event then they would be the licenses that appear maximally in the situation as something inexistent. By definition, though, these licenses are the ones that leave the possibility open for the software to be taken back into a proprietary state, of exclusive access, or at best mixed with software with a closed license. This may lead to increased profit returns, but it likewise means there is no event here, only practices that can be subsumed under the established proprietary regime. If the specific license is not equal to the GPL in its restriction of future protection for the community, then it falls below the threshold that can support eventual change, and it leaves itself positioned to being absorbed back into the commercial system without any significant alteration. The OSI efforts left this an option, and this compromised position leaves open a possible return to business as usual. The conclusion is that the license makes a difference.

To identify the inexistent elements at the community-managed governance level is to track the configuration as a system. What we can say in general is that where the conventional wisdom would see the optimization of production protected by a hierarchical system with a control structure governed through the careful supervision of a manager, with the open-source event a different configuration rises to the surface. We know that prior to Linux this community-managed governance model was thought impossible even by those within the hacker culture: “anyone who says you can have a lot of widely dispersed people hack away on a complicated piece of code and avoid total anarchy has never managed a software project” (DiBona *et al.*, 1999: 247). A measure of the full force, or maximal intensity, of FLOSS will only be determined over time as experiments are undertaken to put into practice the various aspects of this community organization and observe the effect on commercial development. Naming a few examples: the gate-keeping access to the community (control), the nature of selection of work activities which is placed fully in the community members' hands (command), and the transparency level of decisions and knowledge of the software (secrecy). We know the commercial software world still responds to the impactful force of FLOSS when marketing teams cannot avoid mentioning the companies positioning with respect to open-source. The logic is fairly straightforward at this stage in the development. Anything less

than the adoption of GPL license or anything omitted from the major elements composing a community-managed governance model, and it will be declared that something other than an event has taken place.

In general it can be said that the FLOSS model in its composition tends toward being open to a maximal degree tolerable. The inexistent that appears is the disappearance of the importance of previous organizational supports that provided: control, command, and secrecy. This disappearance is not an absolute disappearance and avoids any identification with radical anarchy. Instead there is the reduction to the minimal supportable state while still maintaining cohesion. This organizational adjustment approximates, without reaching completely, the voiding out of organizational boundaries.

### **A FLOSS altered world**

Having elicited the case of the world of commercial software development encountering an event, what then are the implications? How will subsequent reactions testify to radical change altering this world, first at the level of companies producing in the realm of information-based products, as opposed to any mere recovery of the status quo after a minor disruption? If the truth of an event can span worlds and given the political tenor of this FLOSS study, can an invariant dimension of FLOSS contribute to discovering new ideas for progress in other politically charged settings? Lastly, if we entitle ourselves to invoke a reading of Badiou on FLOSS, what is the attendant reading of FLOSS on Badiou?

The rupture of the open-source event continues to the present in the world of commercial software development (Kevin Crowston *et al.*, 2012) and companies producing immaterial goods. If the responses by commercial software firms fall short of adopting the complete suite of FLOSS elements, then we can anticipate, following Badiou, that the situation will remain unchanged. Change that remains below the threshold of evental change is susceptible to collapsing back through subsumption into the norms of the status quo. Cases demonstrating a return to status quo, including alignment with open-source, do not singularly disprove the logic of the event. The criticism that the free labour harnessed in this movement simply returns back to the capitalist hand would be confirmed (Tiziana Terranova, 2004). Only continued experimentation testing the consequences of this event and its logic will produce one possible alteration to this world, driven forward by the incentive for an improved mode of production. Proving the event wrong would involve testing, under full compliance, the GPL and what we can make out of community-managed governance, and repeatedly failing.

The ontological implications of the FLOSS movement is that workers have positively responded to an environment created through license and community-managed governance when they are ensured that they and their work will belong and be presented. Companies that risk licensing with the GPL and continue to adopt the progressive traits of FLOSS test out the force of this event. When Facebook development teams let developers direct where they will apply their effort, the march towards an event altered world continues (Yee Lee, 2011: unpaginated), albeit under a single authority. Linus' recommendation for following the FLOSS lead would be: "Early in the game, the company asked for my advice and I told them they needed to fight the urge to have decisions made internally" (Torvalds and Diamond, 2001: 231) Contrasting regressive strategies are: companies dual licensing code with a second license for taking the code private ("Dual-licensing as a Business Model" n.d.: unpaginated), or dual releasing code with subtle discrepancies between enterprise and community versions ("Alfresco Enterprise Network" n.d.: unpaginated). Studies of the hybrid adoption of open-source projects may not be testing the event itself when the model is stretched beyond its original eventual shape (O'Mahony, 2007: 140-144). This ambiguity and the confusion with mixed cases means that what we do not know about the FLOSS event may remain greater than what we know (Crowston *et al.*, 2012).

The preceding analysis demonstrated the shift to maximal appearance of the inexistent element as the mark of an event: a license that promotes inclusion, and organization without a formal organization. A side effect of this introduction of the inexistent element(s) is the rearrangement of the relations that construct the logic of the situation. When a rearrangement occurs in the post-event world, there is a destructive effect on some of the previous relations. When community-managed approaches include self-assignment of work, there is a corresponding rearrangement of the role of the manager from the former model. The relation of manager to developer is de-emphasized. When workers self-assign work, taking management into their own hands, there is a loss of significance for the hierarchical control structure. A study of the Debian Linux community bears out that the Linux model, including when a movement's leaders leave, remains marked by the restructured role for the top manager (O'Mahony and Ferraro, 2007).

The core directives or values behind community-managed governance which FLOSS promotes are transparency which includes traceability (accountability), non-exclusion (pluralism), autonomous participation with minimal distortion in representation, and decentralization in decision making and organizational control (O'Mahony, 2007: 145; David Berry and Giles Moss, 2006: 27). These values, embodied in the unique configuration FLOSS organization models, have generated change that goes

beyond strictly technological developments. In extending the relevance of this phenomenon to other domains, the core values correspond to values that would dominate the interests for any generic participatory governance models. Using FLOSS as a guide the ground has been worked for experiments and research into how parliamentary forms could be altered that could range from the software mediating this governance to the practice of governance itself (Berry and Moss 2006; Walt Scacchi, 2002). “This concept [open governance] seeks to explore the potential and opportunities that can emerge when one views the purpose of digital government as also including how to empower and engage an interested public” (Scacchi, 2002: 4). The cautionary note raised in this study is that a distinction should be made between open-source as event site and the FLOSS elements that compose this event if eventual status is to hold. Because open-source was initiated as a reactionary attempt to accommodate the reception of the phenomena behind FLOSS in commercial circles, it suffers from misrepresenting the deeper implications of the genuine event. If there is no event, then it is business as usual.

The impact region of this event reaches into the study of organizations as tackled by organizational sociologists. In a review of a half-century of organizational sociology, W. Richard Scott notes the transformative phase when open system models were introduced into this field (2004). Open system models focus on the dynamics of a system, here an organization, that is reacting to what is beyond its borders and ways that the organizational form adapts to these external environmental forces (Scott, 2004: 10) Yochai Benkler's study of FLOSS (2002), revealing how FLOSS breaks with Coase's firm-based transaction cost model, one of the branches of open system models, is promising research but it stays within a narrow understanding of organizations. What happens when we turn to open system models in a more generalized study of boundaries? “Although it seems premature to declare the advent of the 'boundaryless' organization, as have some overenthusiastic observers, there are many indicators that boundaries have become more permeable and less fixed” (Scott, 2004: 10). FLOSS demonstrates a radicalization of the boundary-less organizations:

Scholars have proffered multiple conceptions and examined various indicators in their study of boundaries, including actors (distinctive roles, membership criteria, identity), relations (interaction frequency, communication patterns, networks), activities (tasks, routines, talk), and normative and legal criteria (ownership, contracts, legitimate authority). (Scott, 2004: 10)

At each point where the scholars have identified potential boundaries for study, FLOSS has presented organizational adjustments that gravitate towards reformed boundaries that are more permeable than previously imagined possible.

We have been studying the reduction in stability of the boundaries in organization structures: actors, relations, activities, and normative or legal criteria. Where previously there was an organization functioning as the umbrella, we now have no single organization. However, the open systems model for organizational sociology itself fits within the more general tension in contemporary sociology of explaining organization from a relational, or transactional, perspective over the previously dominant perspective with a substantialist base (Mustafa Emirbayer, 1997). Here things become more awkward for Badiou also. With a relational perspective thinking changes from conceptualizing positions in structures as somehow fixed entities, even if only at an ontological level, to how relations themselves construct the situation (Emirbayer, 1997: 286-291). As FLOSS demonstrates, the loosening and unbinding of organizational structures threatens to destabilize any conception of organization that begins or reduces to entity-ness. What now must be thought is whether the new relations spawned by an alternative license model do not themselves generate new structures. There is not only more permeable boundaries and a new rearrangement brought on by the inexistent, but relations themselves now generate new organization, structural forms. The composition of the structural elements are themselves operations of relations in formation. Not only is organization not a noun (Scott, 2003: 98), but the appearance of positions is obtained itself from new relations. Membership is not obtained by filling a job position; it is the operation of contributing to the source code repository. It is not an attribute of actor nor an attribute of position.

## Conclusion

Thinking about a concrete situation as a Badiouian event necessitates beginning again at the beginning. By revisiting the location where the world of commercial software development experiences its brush with FLOSS allows us to identify the components of an event. If we think the new in this event from its underside, FLOSS, and not the hybrid models submerging the new within the commercial world, open-source, then it is possible to recognize the evental status. Our intervention remains incomplete, as the experimentation still continues on, for there is no guarantee the evental impact will be fully adopted by the commercial world. This effort is surging ahead as Crowston et al., have the number at one hundred eighty-four published empirical studies on FLOSS (Crowston *et al.*, 2012: 5). This arguably offers further proof of an evental impact of a maximum degree on this world if there is so much attention. If what Marx discovered with capitalism is at bottom a relationship (Emirbayer, 1997: 290), we can conclude that FLOSS is the introduction of a change in relations at the foundation of commercial practices .

## The International Journal of Badiou Studies

According to Badiou events think, and this thinking must be allowed to alter and reshape even his own philosophical frameworks. The concrete case of FLOSS attests to the possibility of an alteration in the systematic thinking of relations. If one of the lessons from FLOSS is that relations between nodes themselves restructure the world they become exposed in, then ontological thought that reduces entities to holding down positions in structures is thrown in doubt. Badiou will be forced into making some adjustments so that the halting point for relational thinking moves even deeper into foundational thought, past the restrictive boundaries of entity-based substance.

Notes

1. Situation is the term Badiou uses when the focus is on the ontological dimension of a scene, and world is used when the focus is on the phenomenological level of the consistency of relations. In this paper, world and situation are used synonymously depending on the level of the analysis or for readability.
2. To consider the legitimacy of the new elements belonging the reader should familiarize themselves on the role of a generic extension (Oliver Feltham, 2008:108-113).
3. The following site provides a primer on the history of Libre software and some background on the term FLOSS: <http://sinetgy.org/jgb/articulos/libre-software-origin/libre-software-origin.html>
4. GNU, an recursive acronym for GNUs Not Unix, is meant to pay tribute to the role that Unix played in operating system development while at the same time distinguishing GNU as an alternative approach. GNU/Linux is the combined system with Linux as the kernel and the GNU suite of shell tools that can run on this kernel.
5. The event is an exception or impossibility in a world. It is likewise in its conceptual development an exception or impossibility from within Badiou's systematic ontological and phenomenological philosophical frameworks (2011: 49-50; 2005a: 184). Impossibility is drained of any determinism taking it beyond a Deleuzian model of emergence that Manuel DeLanda outlines as a “concrete space of possibilities with a definite structure” (DeLanda, 2011: 17).

**Bibliography**

- "Alfresco Community Network," Alfresco Datasheet Enterprise Network. Available at: [http://storage.pardot.com/1234/46851/Alfresco\\_Datasheet\\_Community\\_Network\\_eng.pdf](http://storage.pardot.com/1234/46851/Alfresco_Datasheet_Community_Network_eng.pdf). Accessed March 27<sup>th</sup>, 2012.
- Badiou, A. (2011/2009), *Second Manifesto for Philosophy*, trans. by Louise Burchill, Cambridge: Polity Press.
- (2009/2006), *Logics of Worlds*, trans. by Alberto Toscano, London: Continuum.
- (2005a/1988), *Being and Event*, trans. by Oliver Feltham, London: Continuum.
- (2005b/1998), *Metapolitics*, trans. by Jason Barker, London: Verso.
- Benkler, Y. (2002), "Coase's Penguin, or, Linux and The Nature of the Firm," *The Yale Law Journal*, 112.3: 369-446.
- Berry, D. M and G. Moss (2006), "Free and Open-Source Software: Opening and Democratising e-government's Black Box," *Information Polity*, 11: 21-34.
- Bosteels, B. (2011), *Badiou and Politics*, Durham: Duke University Press.
- Crowston, K., Wei, K., Howison, J and A. Wiggins (2010), "Free/Libre Open-source Software Development: What We Know and What We Do Not Know," *ACM Computing Survery*, 44.2.7: 1-35.
- DeLanda, M. (2011), *Philosophy and Simulation: The Emergence of Synthetic Reason*, London: Continuum.
- Demil, B. and X. Lecocq (2006), "Neither Market nor Hierarchy nor Network: The Emergence of Bazaar Governance," *Organization Science*, 27.10: 1447-1466.
- DiBona, C., Ockman, S. and M. Stone (1999), *Open-sources: Voices from the Open-source Revolution*, Sebastopol: O'Reilly.
- "Dual-licensing as a Business Model," OSS Watch. Available at: <http://www.oss-watch.ac.uk/resources/duallicence2.xml>. Accessed Mar 28<sup>th</sup> 2012.

- Emirbayer, M. (1997), "Manifesto for a Relational Sociology," *American Journal of Sociology*, 103.2: 281-317.
- Feltham, O. (2008), *Alain Badiou: Live Theory*, London: Continuum.
- Hamerly, J., Paquin, T & Walton, S., "Freeing the Source: the Story of Mozilla," in DiBona, C., Ockman, S. and M. Stone (1999), *Open-sources: Voices from the Open-source Revolution*, Sebastopol: O'Reilly: 197-206.
- "History of the OSI," Open-source Initiative. Available at: <http://www.opensource.org/history>. Accessed January 14<sup>th</sup> 2012.
- Lee, G. K and R. E. Cole (2003), "From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development," *Organization Science*, 14.6: 633-649.
- "Licenses by Name," Open-source Initiative. Available at: <http://www.opensource.org/licenses/alphabetical>. Accessed January 14<sup>th</sup> 2012.
- McGowan, D. (2005), "Legal Aspects of Free and Open Source Software," in Joseph Feller, Brian Fitzgerald, Scott A. Hissam, Karim R. Lakahani, *Perspectives on Free and Open Source Software*, MIT Press.
- Moglen, E. (1999), "Anarchism Triumphant: Free Software and the Death of Copyright," *First Monday*, 4.8. Available at: <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/684/594> Accessed April 3<sup>rd</sup> 2010.
- Moody, G. (2001), *Rebel Code: The Inside Story of Linux and the Open-source Revolution*, Cambridge: Perseus Publishing.
- Moon, J. Y and L. Sproull (2000), "Essence of Distributed Work: The Case of the Linux Kernel," *First Monday*, 5.11. Available at: <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/801/710>. Accessed June 22<sup>nd</sup> 2010.
- O'Mahony, S. (2007), "The Governance of Open-source Initiatives: What does it mean to be community managed?" *Journal of Management and Governance*, 11: 139-150.

## The International Journal of Badiou Studies

- O'Mahony, S. and F. Ferraro (2007), "The Emergence of Governance in an Open-source Community," *Academy of Management Journal*, 50.5: 1079-1106.
- "The Open-source Definition," Open-source Initiative. Available at: <http://www.opensource.org/docs/osd>. Accessed January 14<sup>th</sup> 2012.
- Perens, B., "The Open Source Definition," in DiBona, C., Ockman, S. and M. Stone (1999), *Open-sources: Voices from the Open-source Revolution*, Sebastopol: O'Reilly: 171-181.
- Raymond, E. S. (2000), *The Cathedral and the Bazaar*. Available at: <http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>. Accessed May 3<sup>rd</sup> 2010.
- Rosen, L. (2005), *Open-source Licensing: Software Freedom and Intellectual Property Law*, Upper Saddle River: Prentice Hall.
- Scacchi, W. (2002), "Understanding the Social, Technological, and Policy Implications of Open-source Software Development." Available at: <http://www.ics.uci.edu/~wscacchi/Papers/New/OSS-Policy.pdf>. Accessed March 19, 2012.
- Scott, W. R. (2003), *Organizations: Rational, Natural, and Open Systems*, Upper Saddle River: Prentice Hall.
- Scott, W. R. (2004) "Reflections on a Half-Century of Organizational Sociology," *Annual Review of Sociology*, 30: xxi, 1-21.
- Shah, S. K. (2006) "Motivation, Governance, and the Viability of Hybrid Forms in Open-source Software Development," *Management Science*, 52.7: 1000-1014.
- Shirky, C. (2008), *Here Comes Everybody: The Power of Organizing Without Organizations*, New York: The Penguin Press.
- Stallman, R. (2010), "The GNU Manifesto," GNU. Available at: <http://www.gnu.org/gnu/manifesto.html>. Accessed March 29<sup>th</sup>, 2012.
- Terranova, T. (2004), "Free Labor: Producing Culture for the Digital Economy," *Social Text*, 18.2: 33-58.
- Torvalds, L and D. Diamond. (2001), *Just for Fun: The story of an accidental revolutionary*, New York: HarperCollins Publishers.

## Daryl Campbell

Wayner, P. (2000), *Free For All: How Linux and the Free Software Movement Undercut the High-Tech Titans*, New York: HarperCollins Publishers.

Weber, S. (2004), *The Success of Open-source*, Cambridge: Harvard University Press.

Williams, S. (2002), *Free as in Freedom: Richard Stallman's Crusade for Free Software*, Sebastopol: O'Reilly.

Yee Lee (2011) "How Facebook Ships Code," FrameThink. Available at: <http://framethink.wordpress.com/2011/01/17/how-facebook-ships-code/>. Accessed February 7<sup>th</sup> 2012.