

AFTERWORD: RECOMPILING

Christopher M. Kelty

Curiously, none of the articles in this issue of *Criticism* are about open source. For an issue boldly labeled “Open Source Culture and Aesthetics,” this might appear to be a problem. The claim is unfair, of course, but not inaccurate: everything depends on the scope of the variable. The local variable—very local—is a fairly mundane practice of software production and distribution (also called *free software*), diverse in itself to be sure, but really quite distinctive; and, in terms of the great swirl of cultural and technical production, not nearly as widespread or important as, say, standardized machine tools or spreadsheet accounting.

The global variable, however, is something more like a zeitgeist, a discourse, a social form, an aesthetic, and a political cipher. It is an unordered array of ideas, practices, capitalisms, technologies, moments, and movements, from the cravenly enthusiastic to the naively critical to the dismissive. Knowledge economies, distributed collaboration, Web 2.0, critical approaches to intellectual property law, crowdsourcing, fan culture, prosumption, peer production, new configurations of property and labor, changing conceptions of creativity, novelty and aesthetic norms, do-it-yourself (DIY) culture, and so on.

As the editors of this issue expertly explore in the introduction, over the last decade this local variable has been progressively redefined as a global one. *Open source*—the local, specific software development practice—has been modulated and transformed into something anyone can “call” (as it were) from within any function. Consider how open source was introduced on *Wikipedia* in 2001:

Open source computer software is nominally owned by one individual or entity and then licensed out according to an open source license; the license gives the user free use of the software as well access to the source code, so that

the software can then be further developed by whoever is interested.

(http://en.wikipedia.org/w/index.php?title=Open_source&oldid=271554)

And then consider the entry from 2011:

The term *open source* describes practices in production and development that promote access to the end product's source materials. Some consider open source a philosophy, others consider it a pragmatic methodology. Before the term *open source* became widely adopted, developers and producers used a variety of phrases to describe the concept; *open source* gained hold with the rise of the Internet, and the attendant need for massive retooling of the computing source code. Opening the source code enabled a self-enhancing diversity of production models, communication paths, and interactive communities. . . . Subsequently, the new phrase "open-source software" was born to describe the environment that the new copyright, licensing, domain, and consumer issues created.

(http://en.wikipedia.org/w/index.php?title=Open_source&oldid=418767638)

Aside from demonstrating what happens to clarity of thought in the hands of *Wikipedia*, this change, which occurred sometime around the summer of 2005, represents a moment of historical forgetting, in which something concrete (a bunch of programmer geeks and their Silicon Valley admirers rewriting the once-beloved tools of university computer science departments and distributing them freely on the Internet by using a clever legal license) is replaced by something abstract (a philosophy or a pragmatic methodology of production in which access to the sources is promoted), and in the process the causal relation between the concrete and the abstract is reversed.

On the one hand, this is simple misrecognition (Marx is rolling in his grave, but probably not uninterested in how this change can be so clearly captured in the detailed revisions of *Wikipedia*—a veritable cinema of ideological formation). There are concrete, empirically specifiable, and material realities to *open source* and the rush to replace them with a label like *peer production* can only obscure that there is something called open source software production, that it consists of specific practices and

technologies, and that it is, for the most part, an ongoing concern in the heart of the information technology industry. And what's more, it is increasingly a practice with none of the glamour or historical grandiosity attributed to it early on but with real effects on how that industry operates as part of contemporary capitalism.

Indeed, whatever open source was around 2001 (and ergo, whatever the Internet was), it's a dramatically different thing around 2011. In 2001, self-important geeks like Eric Raymond and Richard Stallman (the only people scholars ever seem to consult on the subject) controlled a discussion about the production of free software and the nature of the Internet, aided and abetted by semicapitalists and hybrid entrepreneurs like Bob Young (Red Hat software) and Tim O'Reilly (O'Reilly Publishers). In 2011, by some estimates, paid employees of blue-chip companies like IBM, Intel, Apple, Hewlett Packard, and Google perform more than 75 percent of the production of open source software, and most projects are in the process of formalizing their existence as corporations, nonprofits, or foundations.¹

Today, *open source* in the industry generally still means what it did in the past: openly licensed software applications for installation on a personal computer (PC) or server, often software of an infrastructural sort. Except that nearly every term in this definition has also changed, in some cases starkly. Perhaps the most dramatic shift has been the replacement of the architecture of the PC and its software by devices and their *apps* (applications). All this can still be conducted as open source software production—and very much is by the likes of Google and the hundreds of companies they have bought—but it hardly has the same meaning as it did in 2001. The discussion amongst geeks—such as it is, muted and subterranean, and primarily for those who still insist on the label *free software*—has turned to the problem of how free software has become impossible in the age of *cloud computing*, *software as a service*, or, more generally, the explosion of apps and their stores.² As our style of computing shifts away from the stand-alone PC and its applications to one where all our applications and data are hosted by corporations charging for the services (or giving it away gratis in hopes of winning advertiser dollars), the ideals of free software—grounded in individualism and self-help, mutual aid, and freedom to transform our technologies—become harder to implement or even imagine.

Thus has debate turned from a focus on proprietary applications and infrastructure to a general concern with the proliferation of closed platforms like Facebook and Google; a confused welter of concern about privacy, net neutrality, and security; and a lingering concern about the sustainability of the utopian alternatives that free software represents.

Despite Google's claim to use and value all open source technology, the fact remains that there is but one Gmail, and they control it absolutely. Gone from most debates is any discussion about the magical powers of self-organizing distributed collaboration. Software engineering, anyone involved eventually discovers, is still fiendishly hard to do well.

On the other hand, there is something like conceptual specification happening in this move from open source software to open source generally—and especially in pieces like those included here. It concerns whether this materially specific set of practices actually represents or stands in for a more general historical transformation—something that crosses all domains of production and consumption, not just those narrowly associated with software, networking, and information technology. In my own ethnographic work, I noted that this intuition (that open source is something more than just ways of producing and distributing software) follows remarkably quickly on the heels of understanding the details of the practice. What I just characterized as misrecognition is instead almost uniformly experienced as a profound recognition—indeed, even a revelation.

No doubt, some such recognition birthed this special issue of *Criticism* devoted to “Open Source Culture and Aesthetics.” It's too bad that the actual practice of open source software production (especially free software and its real and imagined utopias) is not the subject of more intense, sustained *cultural* analysis, but it is certainly no tragedy. I may be the only person in the world to feel this particular pang of loss. However, it is nonetheless curious that software and its material, technical, and economic gallimaufry are not more central to the general analysis of *open source culture* (open access, open commons, open science, open societies, or openness generally) in these pieces and others like them.³ As I said, the accusation is unfair (in this issue, both Stephen Voyce and Ben Roberts delve into the details of open source software circa 2001 as part of their arguments), and certainly the great majority of work in the humanities simply continues in blissful ignorance of software, open or not. Nonetheless, the point is whether software *matters* or not to the way culture turns out.

The desire to see open source *software* be relegated to the status of just one expression of open source *culture* stems no doubt from the sense that software is an instance of culture, not a cause of it. And as a result, the existence of open source (both local and global variables) instantiates a drive to search for its antecedents, its ghostly predecessors whose recognition now can help make sense of how open source must have come about, and how open source software will have turned out to be just an instance of this more general historical process.

It is almost as though software as a *problem* disappears in the way same way software as a *thing* does: by being compiled into a set of binaries that run in the background but are no longer legible as such.

Each of these pieces engages in, or observes, something like *recompilation*—in particular, a recompilation of the sort that renders the sources themselves virtually inaccessible. Ken Thompson (the coinventor of UNIX) described this process in his Turing Award speech, a very short, very intense meditation on the impossibility of trusting source code. The speech is concerned with hackers and security and trusting code, and has its own set of interests and contexts, but the point of his observation was that no matter how much scrutiny one undertakes, a program (in his example, one that produces its own source code as its output) can hide its origins in the process of being compiled, run, and the output recompiled.⁴ It is this moment of recompiling the source code that is akin to *historical forgetting*: the moment when one starts to see a general abstract cultural form instead of a concrete historical one. It is this moment that all these pieces observe, reflect on, or in some cases perform.

Roberts's piece most directly addresses this problem: the movement by which Yochai Benkler was able to recompile the practices of free software into something more general and ostensibly universal, called *peer production*. Benkler's work is curious: it has been incredibly influential in general without ever being taken seriously in its specifics by its intended audience of lawmakers (the Digital Millennium Copyright Act [DMCA] still stands, copyright still expands, and commons are as hard as ever to defend legally).

What Benkler's move from free software to peer production obscures is that all his examples (with the possible exception of car pooling) are either variations on free software or directly dependent on the tools, networks, licenses, and practices innovated therein. It is only by strategically ignoring the material and practical assemblage of things that emerged through free software and the Internet (which I continually insist are of a piece), in favor of a focus on the economic implications of different legal regimes (which is Benkler's bailiwick, after all), that he can reduce free software to one instance of a more general and new mode of production. Compile base into superstructure, discard source code, run superstructure to generate a new explanation of the base. Repeat as necessary.

But lawyers and economists are doing this *all the time*, with every phenomenon around us. We hardly need Marx or "immaterial labor" to diagnose this constant recompilation of what were once concrete material arrangements of power into simple binaries of legal and economic orthodoxy. But we do need, I think, to better resist it. Roberts's recognition

of *individuation* is a good step in this direction: free software projects are indeed *projects*—not firms, not nonprofits, and not voluntary associations. They have an individuated organizational component, but it is not reducible to any of the forms we know. And indeed a similar diagnosis might be made of the more recent reliance on the language of *platform* to identify this individuation problem.⁵ Nonetheless, it is the case that many of the existing free software projects are undergoing a transformation from this novel individuation to something more familiar (e.g., Star Office begat Open Office, which was bought by Sun Microsystems, which was bought by Oracle. While the software is still open source in a recognizable sense, this formalization of the project's status as one engineering project amongst others in Oracle's portfolio cannot be without consequence).

Voyce's piece participates in the misrecognition but only by way of prelude. His sources for explaining the rise of an open source poetics are capsule histories of open source software and the intellectual property struggles of the last quarter century (and its centuries-long prelude).⁶ At a surface level, all this introductory work is still quite necessary. Both the pathologies of intellectual property and the success or failure of open source remain a *mysterium tremendum* for the majority of scholars out there: they have yet to have their revelation. For Voyce's piece, recompilation is quick work—what really makes a difference is not open source software but open source poetics, or what used to be called found poetry, concrete poetry, appropriation, language poetry, and so on, but has since been modulated, upgraded, reverse engineered, and reconcocted by folks at UbuWeb or Poetic Research Bureau into a new brew. Open source software is a warrant and agent of legitimation for open source poetics. But I wonder how it would look if the story of open source software detailed what it looks like now (bureaucratized, nonprofit-ified, App-happy iPoetry?), not what it looked like ten years ago? But Voyce's intuition about this strikes me as accurate: one should turn to the work of poets first, and not the work of intellectual property lawyers (critical though they be) or free software demagogues, to make sense of what's happening both to poetry and through poetry to software and culture. Lest I be confused with a demagogue or seen as nostalgic for the heady golden days of free software, I actually think UbuWeb and its partisans represent a more chthonic practice than either free software or its peer production mirages. It's never a simple question of causation or precedence, but of the emergence of new forms, and contemporary poetry is (as hard as making this claim might seem) where open source gets some of its sources.⁷

There are two moments of recompilation in Leisha Jones's piece in this issue of *Criticism*: one is the relation between the historically proximate

problem of *fan fiction*, or fan culture after the Internet, and the more distant *Bildungsroman*. The piece is enlightening because, like Voyce's, it observes a difference between the kinds of writing that preceded open source and those that follow it, but in this case without any explicit reference to the actual practices and techniques of open source software.

If my own hazy ethnographic recall serves me, fan fiction evolved in close companionship with some open source software communities, with no small number of overlapping participants trading ideas, texts, problems, and issues back and forth—most obviously issues of copyright and fair use, but also concerns about distribution, the generation, and sustainability of communities and the necessarily uneasy relationship with the industry that can give or take its life (to use an admittedly clumsy *Twilight* metaphor). Stretch this process out in time, as an exemplar of *Bildung*, and one has the makings of a sociologically rich explanation of how hacking and fan-fiction (fanfic) writing together generate proper ways of becoming for this generation and the next: a problematic that Jones labels with that incredibly ugly word *prosumer* replete with the contradictions and deceptions therein.

But the appearance of the prosumer is the second moment of recompilation: the *open source novel* becomes less about a concrete set of new ways of being (new forms of property, new interactions with people, new tools for making videos or distributing texts or building one's own fanfic fan base) and more about a set of subjectivities and collectivities (girl hordes) enmeshed in the processes of capital and desire (sensate, brand, jouissance, friend, stereotype, polyglot). The prosumer is more a suspect in Jones's mystery than a protagonist, and this is good. But it fills the same role in the work of those who celebrate it (e.g., Ritzer and Jurgenson)⁸ as peer production in Benkler's work: a way of collapsing the material specificities of capitalist production into a figure that has always already been lurking in capitalism, a figure repressed by the binaries of production and consumption. Such a logic is entirely too complex to be useful to anyone, unfortunately—but, as Jones intuits, not irrelevant to contemporary *Bildung*.

Alexander Galloway's piece in this issue of *Criticism*, finally, is blunt about open source: "The open-source culture of new media really means one thing today: it means open interfaces" (para. 22). This might be an effect of recompilation rather than an instance of it: it's not hard to distinguish between this open source and that open source, provided one takes the time to do so. But recompilation discards the source (open source software), as it were, and introduces the new runtime instance of open source (culture) to which it becomes possible to object, probably for the right

intuitive reasons, as something that is blinding us to our own interests. Thus does Galloway counsel: “The interesting question is not so much whether open source is ‘more open’ or ‘less open’ than other systems of knowledge, but rather the question ‘How does open source shape systems of storage and transmission of knowledge?’” (para. 22). And, we are beguiled by the “short-term candy of openness as such” (para. 24) and risk mistaking a very local variable for the global one: the renewed spirit of capitalism. I think I agree.

Galloway’s quarry is elsewhere, however, though not unrelated: the state of academic media studies. Galloway’s critique of Lev Manovich’s work is also concerned with a moment of recompilation: Manovich’s focus on cinema as the original new media. For Manovich, new media in its contemporary specificity is compiled into a cinematic formalism. It becomes a variation on a theme that only now appears to have been the first instance of new media. Such a move strikes me as remarkably common in new media today (witness the spate of books about when old media were new), even if Manovich’s performance of it is somehow more influential and appealing to those with nostalgia for the age of the *auteur*. Galloway’s critique of the seeming disappearance of history is appropriate then: the work of thought in Manovich’s case—the reduction of new media to the thing that cinema was always about to give birth to—cannot even take place without a forgetting of the specificities of history.

To some extent, the fate of media studies hangs on the question of whether this history matters. In some ways, as in nearly all mainstream sciences today, the accurate and responsible recognition of history in media studies remains irrelevant—or at least only rears its head when a priority dispute or a prize is at stake. This could be the becoming-scientific of media studies, or it could be fodder for another generation’s critique and reformulation. For my money, reducing new media to the formalisms of cinema has a different and more troubling effect: it carves out for new media studies a space focused strictly on the aesthetics of interaction, especially those studies that involve perceptibility, the body, and affect. What does this carving leave out? Might it be possible to learn something from open source software that would highlight the negative image of this carve-out, the kinds of new media-enabled practices, problems, and art forms that are really not participating in quite the same set of problems as cinema or new media recompiled as cinema? What are the effects of this appeal to cinema, in contrast to an appeal to actually existing new media?

Finally, perhaps not unrelated, it is (to me) surprising that that none of these pieces reflect, even if only in passing, on whether *open source* means anything for scholarly production in the humanities itself. This could

unfold in several dimensions. In the simplest dimension, it is a question of whether open access is compatible with humanities scholarship? Why is there such resistance to it in the humanities? Of what advantage is it to allow giant corporations to control nearly every stage of the production of our work and to charge us unsustainable sums for the pleasure? How might it be possible to use our scholarship to help transform the system of hiring, promotion, and tenure into one that both values and validates the *availability* of work in any medium and not its inaccessibility and restriction to the standard-form written article? I'm all for critical suspicion around open access and its utopian promises, but there might be a baby/bathwater problem needing our attention here.

But there are other dimensions to this, as well: if open source means open interfaces today, what does that mean for humanities scholarship? I know what it means for so-called *digital* humanities scholarship, where the question of whether one can access Google maps or Google books in some dynamic and unconstrained way remains a hot one—though the question of whether one *should* seems a less lively part of the discussion. But what does it mean for writing, for theory, or for the organization and efficacy of thought today? Examples such as collaborative fan-fiction writing or the new poets and their explorations of the Internet and its possibilities should serve as key guides, if not harbingers, of what scholarship might become, if it is not to disappear entirely the way so much once-valuable software has. Where are the open interfaces in the humanities today? Or are we stuck with incompatibilities and lock-ins? Should we change more of our practices, or resist the song of openness ever more strongly? Are we sure we know, any longer, what open source means?

Christopher M. Kelty is an associate professor in the Department of Information Studies, the Center for Society and Genetics, and the Department of Anthropology at UCLA. He is the author of Two Bits: The Cultural Significance of Free Software (Duke University Press, 2008).

NOTES

1. For example, see Greg Kroah-Hartman, Jonathan Corbet, and Amanda McPherson, "Linux Kernel Development: How Fast Is It Going, Who Is Doing It, What Are They Doing, and Who Is Sponsoring It? An August 2009 Update" (San Francisco: Linux Foundation, 2009), www.linuxfoundation.org/sites/main/files/publications/whowrites-linux.pdf (accessed 6 April 2011).
2. For example, see "Toward Free Network Services," [Autonomo.us](http://autonomo.us/), <http://autonomo.us/> (accessed 22 August 2011); Open Cloud Manifesto, <http://opencloudmanifesto.org/> (accessed 22 August 2011); and Richard M. Stallman, "Who Does That Server Really Serve?" [Poet's Forum], *Boston Review*, 18 March 2010, <http://bostonreview.net/BR35.2/stallman.php> (accessed 22 August 2011).

3. Indeed, the field that has taken seriously the challenge of articulating a software-as-culture approach, that of *software studies*, is not particularly focused on what open source software represents—for software studies, it is but one instance of software, as opposed to one instance of open source culture (e.g., see Matthew Fuller, *Software Studies: A Lexicon*, Leonardo Books series [Cambridge, MA: MIT Press, 2008]).
4. Ken Thompson, “Reflections on Trusting Trust,” *Communication of the ACM* 27, no. 8 (1984): 761–63.
5. See Tarleton Gillespie, “The Politics of ‘Platforms,’” *New Media & Society* 12, no. 3 (2010): 347–64, <http://nms.sagepub.com/cgi/content/abstract/12/3/347> (accessed 28 March 2011). Similarities to *platform studies* as promoted by Ian Bogost and Nick Montfort are also apposite. See also Adam Fish, Luis Murillo, Lilly Nguyen, Aaron Panofsky, and Christopher Kelty, “Birds of the Internet: Towards a Field Guide to the Organization and Governance of Participation,” *Journal of Cultural Economy* 4, no. 2 (2011): 157–87.
6. On which, really see Adrian Johns, *Piracy: The Intellectual Property Wars from Gutenberg to Gates* (Chicago: University of Chicago Press, 2009).
7. It’s intuition, but it’s backed up by the case of Georgina Born’s ethnography of IRCAM (Institut de Recherche et Coordination Acoustique/Musique) in Paris, which in the 1980s was the center of both electronic music and much proto-free software (Georgina Born, *Rationalizing Culture: IRCAM, Boulez, and the Institutionalization of the Musical Avant-Garde* [Berkeley: University of California Press, 1995]).
8. George Ritzer and Nathan Jurgenson, “Production, Consumption, Prosumption: The Nature of Capitalism in the Age of the Digital ‘Prosumer,’” *Journal of Consumer Culture* 10, no. 1 (2010): 13–36, <http://joc.sagepub.com/cgi/doi/10.1177/1469540509354673> (accessed 22 August 2011).