“Soft” wears: Immaterial Labor and the New Sites of Conflict

There are politics of a point and a click. It is no shock to anyone that digital media has seized a central role in all things superstructural: culture, politics, law, and even religion. But, what is becoming more difficult to sort out is just how digital computers mediate, process, and enable global economics, and thus—from many Marxist perspectives—social and political life. The Western world produces different things than it did in Karl Marx’s time; instead of “material” goods, the common sense notion holds that the West now produces ideas, knowledge, and styles. Instead of building the machine, the hardware, we produce the produce the software that runs it. Even Raymond Williams’ inversion of base and superstructure no longer holds. Like the problems of ideology and ideology critique, “software” as a stand-in for the larger systems of global exchange remains misleading, opaque, and ineffectual as an entry point into leftist politics. Software is both a commodity and a text that continues to be misunderstood and under theorized. We have lived through many digital revolutions in the last twenty years, but too many things have stayed the same.

This project attempts to read software—and digital media more broadly—through the lens of labor. Computers promise to lessen workload, but really workers just donate more of their time than ever before. Bodies are not supposed to matter like they used to, but this is a fallacy. Bodies matter differently, just as “material” labor has changed. Software poses an interpretation paradox of labor that, if better answered, could better position critics to understand the rifts within contemporary, cosmopolitan, global, networked society. If the production of value in software and other digitally mediated objects remains fetishized, we cannot hope to understand the possibility of the global equity of labor.
Two pieces of software that seem the most overtly political are web browsers and word processors. For these ubiquitous computing needs there are plenty of options for the consumer. Of course the proprietary options are the most well known, and not surprisingly, owned by the same company, Microsoft Internet Explorer (IE) and Microsoft Word. Against these hegemonic programs there are non-propriety, “open-source” alternatives, the two most popular being Mozilla Firefox and the OpenOffice Suite. The open-source products differ in many ways from their analogs, but most notably, due to the fact they are completely free, forever. Mozilla and OpenOffice accept money through donations, surely, but these companies have created a business model that does not require buying and selling. In order to create better products than companies built for these purposes, these companies consist primarily of volunteers. Instead of hiring a staff of programmers to work 40-60 hours a week, Mozilla hires one or two and relies of the rest of the surplus labor to be contributed from the community. Incomprehensible to Marx, or any orthodox economist, the Firefox browser is extremely competitive; It accounts for 22.48% of Internet traffic, while less than IE’s 66.10%, it is three times as possible as Apple’s Safari browser, 8.25% (“Browser Marker Share”). Traditional Marxist labor theory cannot account for the quality of free labor producing a widely popular product. As much as commentators have tried to read this quality of gift economy as an expression of production outside of capital, or at least a possible outlet, this is not the case. Capital has changed to require and engulf a new form of labor; labor has not averted capital.

This paper will explore the changes in labor and the theories of labor in the digital age, particularly through specific heroic renderings of “knowledge workers”, the programmers, testers, and users of software, and the movement against copyright (also known as the copyleft/copyfight movements). While there are exciting developments in these fields for social theory, they are also the most prone to delusions of revolutionary grandeur. The major problem is that the majority of “immaterial labor” is hidden in its form. This paper hopes to make these sites of labor visible in
order to contextualize these phenomena so that we can begin to understand how to proceed towards understanding network society.

The Problem(atic) of Software

Wendy Chun traces a gendered and ideological history of software since its invention around the time of World War II. She explains the problem in a technical but illuminating way, “It is truly remarkable that software—designed to obfuscate the machine and create a virtual one on buried commands—has led to the overwhelming notion of computation as transparent” (Chun, 43). Software is envisioned as granting access to the inner processes of the machine; it’s supposed to make desirable things happen, whether its surfing to a specific webpage or inputting text that brings this writer closer to a conclusion, software grants the interface the manipulate the electronic basis of the machine, the hardware. She thus calls software, “a functional analog to ideology” in the sense of “false consciousness”, mentioning both Louis Althusser and The Matrix in the same sentence (Chun, 43). After showing how the original programmers and “computers” were the women who physically plugged in the wires and flipped the switches that drove the early computers, she makes another claim for the naturalization of the current iteration of software. She writes, “Software has become a commonsense shorthand for culture and hardware a shorthand for nature…Our interactions with software have disciplined us, created certain expectations about cause and effect, offered us pleasure and power that we believe should be transferrable elsewhere” (Chun, 47). Beyond the point-and-click fluency of computer users, “masters” or “experts” in the field fall into the trap of seduction of the veiled translation of limited physical labor into immense bits of immaterial labor. Software is anything but transparent, just like ideology. She wants to argue that belief in the power of software, to balance your budget, play your music files, or Twitter your way to a revolution mirrors the invisibility by which ideological processes of hegemony and soft domination occur.
Alexander Galloway takes this discussion even further by explicating addressing the aspects of both politics and interpretation. A former protégé of Frederic Jameson, Galloway compares his reading of software to the ways in which Jameson teases out moments of “utopia” and “reification” as counterpoints of 1970s blockbuster films in *Signatures of the Visible*. Galloway diagrams both ideology and software dialectically and as problematics, in the Althusserian sense. Following Chun, he explains “the fundamental paradox of software: what you see is not what you get. Code is a medium that is not a medium. It is never viewed as it is, but instead is compiled, interpreted, parsed and otherwise driven into hiding by still larger globs of code” (“Language”, 325). Though code is a language, perhaps comparable to the speech-act theory, it is a non-human language, even when read and written by human programmers. In fact, Galloway is often cited for giving voice to a slogan, “code is the only language that is executable” (“Language”, 325). Performance is not the same as execution. While illocutionary acts cause things to happen, such as pronouncing man and wife, code propels a machine into action, causing literal and symbolic material change. The motherboard of my computer is a little different each time I press a key, or each time I open a file on my desktop. Though many of these actions seem visual, they are in fact linguistic and machinic.

Galloway, in responding to Chun, tries to suture her concern for revitalizing ideological critique, while updating the process to account for not only history, but contemporary digital and social processes. To understand this it is necessary to cite him at length:

My task here is not to claim that software has a ‘meaning’, political or otherwise, that can be revealed through a convenient methodological scaffolding called the political interpretation, quite the opposite. A certain networked relation is at play: software, the social and the act of interpretation combine in ‘an intense mimetic thicket’ and it is this thicket, in its own elaboration, can be called the political (“Language”, 329, emphasis added).
Galloway is not reading the genre of software as meaning one thing or the other. That would be as silly as deeming all novels progressive or reactionary, not to mention the murkiness of trying to pin down a cultural objects politics in the first place. Instead, Galloway suggests that the nested layers of meaning, execution, and code resemble and enact the political, or material ideology in an immaterial labor world. Much like how Jaws and The Godfather exhibit the changes in global labor and politics in the late 1970s, Galloway argues that the modern problems of software instantiate the contemporary crises in labor and production that rely so heavily on the culture industry of software.

Clearly software coding and information architecture projects have become some of the largest business endeavors of the 21st century. One need only to think of the possibility of a Microsoft monopoly, Google’s omnipresence, or the preponderance of iPods to remember how central software is in our global culture. But what’s really ironic about these huge software endeavors is that, in order to correctly distribute manpower for either paid or unpaid labor, the workers are put on a need to know basis with the secrets of the machine. Galloway explains: “perhaps the major challenge—in the construction of any large body of software is how to divide the effort among programmers in such a way that work can proceed on multiple fronts simultaneously…[which] depends critically on the notion of information hiding, which makes objects and algorithms invisible, whenever possible” (“Language”, 324). Whether Mozilla or Microsoft, “open-source” or closed, to construct the insanely intricate languages of code, they need to exceed human compression, and thus, a single laborer need only see one part at a time so that she will not be overwhelmed. The idea of “Open access” and the “information age” are built on misnomers. To create, store, and transfer, that data that fuels the “information age” we have to hide it. Software, just like politics or ideology, cannot be transparent, or they seize to serve the functions that empower and manipulate subjects.

Interface Labor
The buzzwords of at least the past ten years of discussion around the democratic and socialist possibilities of the Internet have revolved around participation: participatory (direct) democracy and participatory (social) media like blogs and networks like Facebook, but maybe it should include participatory labor. Christian Fuchs explicitly tries to understand the production of surplus value through software engineering. His major contribution to the study outlines a shift of production philosophy that makes software, in particular, such a good model to study. He explains the state of the business in 2002, though it still holds today as a shift from “hierarchical software design, for which the waterfall model was characteristic, towards participatory and evolutionary software engineering reflects an ideological shift in capitalism which no longer involves dominance…but a strategy of integration that is expected to result in a rise of profits” (Fuchs, 38).

In the program is designed to continually be improved as it goes along instead of proceeding down the assembly line like in the waterfall model that Fuchs references, possible flaws in design or limits to marketability will not be obvious until a substantial amount of labor and time has already been invested.

Decentralizing the work process installs more safety valves for the enormous endeavor. Of course, this idea of participation is not neutral. Fuchs argues, “in informational capitalism, ‘participation’ is participation in the second, instrumental sense, which serves economics interests and hence is pure ideology” (Fuchs, 38). This model of design reflects a larger trend in immaterial labor, the incorporation of free labor, free content, and free development within grander schemas of value production. Christian Marazzi heralds this shift in by attempting to redefine contemporary productivity. He argues, “production neither starts nor ends in the factory…[it] begins even before the worker arrives in the office” (Marazzi, 29). The fetish of the commodity form is written into the labor practice. Fuchs’ description of this trend as “pure ideology” is misleading. Though he correctly dispels this “participation” as non-inherently democratic in any way, he remains insistent on
deeming it “false consciousness.” He feels compelled to do this by demanding that intellectual labor needs a “substantial-material carrier of knowledge” to remain productive, such as the material sale of a product analogous to selling some kind of container (a CD, a packaged piece of software, etc.) the open-source software movements seem to dispel this requirement. Mozilla Firefox is not, to my knowledge, even available in “material” form, only by free download. Though the labor required to develop and maintain a successful product of this scope is abstracted from laborers, is it not the same old mechanism of false consciousness from The German Ideology or even Althusser. “The classical definition of productivity which relates the value of the finished product to the cost of the factors of production (labor and/or invested capital)” Marazzi argues, “no longer has any operational meaning” (Marazzi, 29). Immaterial labor may not be productive in the same way, but it still produces value, content and culture. But when the workday has no limit, and the labor doesn’t live behind sweat, can we still consider it exploitation and alienation? Do those terms have any political traction?

Hidden labor, participation, and the problem of productivity are all other ways of explaining free labor. Unconvinced by the arguments for the hypertextual nature of the Internet empowering the websurfer like Barthes’ concept of the “writerly text” and comparing the decentralized architecture of the Internet to the Deleuzian rhizome, Tiziana Terranova configures the knowledge worker in 2000 as providing free labor that is not simply the extraction of surplus value like in alienated labor. She explains that “the increasingly blurred territory between production and consumption, work and cultural expression…does not signal the recomposition of the alienated Marxist worker [though it also] does not turn every user into an active producer, and every worker into a creative subject” (“Free Labor”, 35). What’s intriguing about this is that Terranova illuminates the possibilities that leftist discourse and corporate conglomerates want the same thing, free labor. The left in terms of freed labor, alternative projects, and capital in terms of uncompensated labor.
She goes as far as to claim that “free labor is structural to the late capitalist cultural economy” which, in her perspective, “embraces some crucial contradictions” which enables the “holy truths” of Marxism to be updated to deal with the contemporary problems at hand” (“Free Labor”, 53,55). Free labor, or the gift economy is strange example of an overlapping supra-consciousness of work. Whether it is the wunderkind of Google who get free meals and a pool table in order to coax him into staying in the office as long as possible to donate their great ideas, or the 13 year old girl compulsively adding content to Facebook—content which Facebook has recently made legal claim that they own—the “knowledge economy” is built off of donated time, and the interfaces of software time-sharing, cooperative, and distributed architecture both require and facilitate this.

Before we go any further, it is helpful to get a broad conception of other popular “Neo-Marxist” readings of production. The face of labor has changed in the last half-century and so has the face of Marxist theories of labor. The central component of Marxist critique shares little with the systems of production that Marx saw and predicted in the mid-19th century. It follows that a failure to understand labor means a failure to perform Marxist critique. Since the 1970s critics and thinkers have tried to grasp these changes with the terms “post-industrial”, “weightless”, “postmodern”, “globalized”, “post-Fordist”, or “knowledge” economies, none of these have succeeded in the same ways the Marxist analysis encapsulated and understood the tendency of capitalist regimes of accumulations of Marx’s time. Michael Hardt and Antonio Negri, following the Italian autonomists, argue that we should understand this shift towards a “new” economy as a shift towards immaterial labor. While many critics of this kind of thinking have accused this vein of thinkers as dismissing the material basis of production, and forgetting the human bodies still toil across the globe to produce the goods that lubricate the means of global trade, Hardt and Negri in fact argue for a materialist understanding of this kind of immaterial labor which “remains material—it involves our bodies and brains as all labor does. What is immaterial is *its product* such as knowledge, information,
communications, a relationship, or an emotional response” (Hardt and Negri, 108). The thorny history of software as a cultural product brings out the tensions in understanding the new aspects of labor and value. They claim to update the problem of Fuchs by explaining that immaterial labor is hegemonic, not simply black and white ideology. They write, “immaterial labor has become hegemonic in qualitative terms and has imposed a tendency on other forms…[and] is today in the same position that industrial labor was 150 years ago when it accounted for only a small fraction…but nonetheless exerted hegemony over all other forms” (Hardt and Negri, 109). The kind of labor they document here is what I will call interface labor: interaction, communication, and cooperation. What they call “affective” I think is better understood as “interface” because it negotiates both the form and content of the processes involved. Immaterial labor involves, at some stage of its development, interfacing or networking between people and between human and computer. One needs to engage the interface to labor, and this is productive labor because it reproduces the interface.

Hardt and Negri bear witness to the revolutionary potential of this now hegemonic laborers which they call the “multitude” instead of the proletariat or any other vanguard party. Rather than rehearse the mostly negative evaluations of this concept, I will try to salvage their enthusiasm towards a more accurate understanding of global labor and its prospects. The update of the proletariat to the multitude means the shift from the public to the common. Seemingly a semantic difference, it resonates with our discussion here. Hardt and Negri explain their critique of economic policy reform, “In our view [progress will entail] will have to be not a return to the public, with the state control of industries, services, and goods, but a creation of the common…This notion…is the basis for a postliberal and postsocialist political project” (Hardt and Negri, 303). The commons has become a key word/concept for the purveyors of “progressive” thinking on the Internet, most notably the Creative Commons (CC) project to reform copyright by renowned Internet legal scholar and public intellectual, Lawrence Lessig. Hardt and Negri directly refer to this project at this
important reiteration of their conception of the commons, so we know that they are referring to an alternative way to license cultural goods as a nexus of the production of a third solution to privatization and state socialism. A quick summary of the Creative Commons license will suffice for these purposes. Copyright is granted automatically, whether you like it or not. When a piece of work is finished, published, or submitted anywhere, it is already copyrighted. Opposed to this strict copyright law that can suspend individual ownership for up to 150 years after the author’s death, Creative Commons intervenes and allows the author/creator to relinquish some of the rights that may not have wanted to claim. For example, you can choose to distribute your work freely for any purpose as long as you are credited, or if the user promises to share whatever they create just as you have. Because of the ease of this license, and the popularity it has gained alongside Lessig’s Internet fame, this license has become an industry standard: Giorgos Cheliots estimated that 61 million CC licenses existed online in early 2007, but the number is probably much larger now (Cheliots).

Clearly the CC project is actively investing in creating new common culture, but some of the many vociferous critics, on the left and more towards the middle, dispute its ability to reform copyright instead of abolishing it. Echoing some of the most obvious critical theory language, David M. Berry and Giles Moss lambaste CC for what they see as empty promises. In sum, their critique reads, “simulacrum of a commons. It is a commons without commonalty…these goods are neither really shared in common, nor owned in common, nor accountable to the common itself. It is left to the whims of private individuals and groups to permit reuse “ Berry and Moss, 2-3). Berry and Moss distrust the promise of CC since its licenses gives the owner choices on what to do with their content, finding a middle ground in “some rights reserved” instead of the “all rights reserved” that copyright de facto grants the author. Centrism is not commonalty, it is just a wide solution to make the most people happen, and the CC license obviously had. One interesting thing to point out here is that though CC is becoming more popular throughout Internet content (even Wikipedia is
currently voting on whether to change the licenses of their content to CC), the FAQ page of the project recommends against using the license for software. The FAQ explains, “We recommend considering licenses made available by the Free Software Foundation or listed at the Open Source Initiative. Unlike our licenses, which do not make mention of source or object code, these existing licenses were designed specifically for use with software” (“FAQ”). Basically the CC license does not speak the language of machines. This disconnection illuminates the political valences of both the Creative Commons project, the movement against copyright more generally, and the Free/ Open Source Software Movements. After the hubbub about democracy on the Internet and the blogging revolution have died down, these movements have maintained some currency in the questions of progressive uses of the Internet.

Much of the rhetoric surrounding these projects and initiatives have boiled down whether free software is anti-capitalist or not. Before we can accurately evaluate this claim, we need to quickly distinguish between free software and open-source software. The Free Software Foundation, developed by software guru Richard Stallman in the 1970s and 80s at MIT, was modeled around a so-called hacker ethic; anything that could be shared should be. Phrased as a question of ethics, the foundation relied on the mantra, “information is born free but exists everywhere in chains.” Flash forward to today and we see the Open-Source Initiative broke this mold. Instead of an ethics it is more of loosely connected reformist practices for software creation and distribution that cares more about keeping the underlying code of the product accessible and modifiable instead of any pretenses of being against hegemonic economic practice; Free Software seems like an anarchist ethics, but Open Source is unabashedly an alternative business model. While this relationship is not as polar as I’m making it out to be here, the difference is provocative. Indeed, Milton Mueller casts this as a rehashing of the ideological cold war of free-market capitalist versus state Leninism, history repeated as farce, which, for him, derails many of the progressive aspects of either type of large scale cultural
or economic endeavor. First off, to be fair, many outside either of these movements have made headlines making this exact claim. Most notably, in early 2005, Microsoft tycoon Bill Gates called “intellectual property reformers ‘new, modern-day sort of communists’” (qtd in Mueller). His argument rests upon the problem of the moral requirement to share information as being a movement against private property more generally. He tries his best interweave the flaws in each position, “you can’t justify an opposition to proprietary information solely on an ethical imperative to share, unless you are willing to extend that ethical imperative to all resources and thus embrace true communism” (Mueller). While I was not aware of a “true” communism, I presume by he’s warning against letting the West resemble the horrors of the Eastern Bloc but this argument sounds very strange if we are to believe that this argument is against setting up old ontological binaries of East and West or Capitalist and Communist.

What’s really the heart of the issue here is the multitudinous uses of the loaded ideologically word “free”. “Freedom of Information” and “Open Access” are surely the catchphrases used by almost all reformers and apologists for digital culture. But Mueller is much more concerned with the definition of freedom in the liberal sense, “A justification for free software grounded in individual choice could serve as a basis for a more consistent, realistic philosophy of freedom of information” (Mueller, emphasis added). While we should be conscious about the rights of individual on the Internet, the loss of ethical solid footing, or a community of users/developers suggests that digital “freedom” is a problematic term. More free market, than a free lunch. In fact, Stallman’s own most notable axiom describing his vision is "Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer” (Stallman). While Mueller is right to try to dispel the cold war mentalities of free versus open source software, we have to remain conscious of the free labor that is injected into either of these projects, though they are outwardly done for different political aims, they accomplish the same goals of production. What we
see from the free software movement is techno-libertarianism not Marxism; freedom to access does not mean freedom from wage labor.

Immaterial/ Invisible labor

This paper has highlighted the inevitability and invisibility of interface labor so it would be unfair to neglect to consider the corporeal labor that is made most invisible by outsourcing, causalizing and “feminizing.” In an ostensibly favorable review of *Empire*, Nick Dyer-Witheford suggests that “we could, very schematically, suggest that the attention paid to “immaterial” laborers be balanced by equal attention to at least two other groups: ‘material’ and ‘immiserated’ workers” (Dyer-Witheford, 76). While I will not have the space to address these two other categories with equal space that I have given immaterial labor through the lens of software, I will try to quickly address both of these other categories. Andrew Ross reads the Free and Open Software movements not as ethical bankrupt, but as shortsighted. He argues that, “for all its admirable political advances, it has to done little to address the suspicion that a predominantly volunteer labor model poses a threat to the livelihoods of future engineers” (Ross, 763). Labor is already invisible, it will be even more spectral when it does not exist. What Ross addresses here is the dystopian vision that the “casualization” of labor will continue all the way up the ranks to prestigious, technically advanced, and desirable jobs just like it has left those behind the technological poverty line in its wake. For examples he provides the all to close to home example of the plights of adjuncts at online colleges like University of Phoenix and the DeVry institute, but also the legal escapades of a company who—until this current economic crisis—had not had a single large lay off in its 34-year history: Microsoft (Vance). One of the reasons Microsoft could afford to keep people on their payroll was refusing to give them permanent jobs with benefits. Ross explains the “landmark, decade-long court case brought against Microsoft by its ‘permatemps’…because they were classified as ‘independent
contractors’” (Ross, 754-55). While less than ethical business practices in any big industry should never be too much of a surprise, it is shocking that this kind of exploitation of American labor occurring on U.S. soil. Even the best jobs one can get are not protected against the realities of invisible exploitations of labor.

The inequity of the lives of many people in the global south will only be mocked by trying to narrate them here. Needless to say, there are gradations of comfort and freedom from exploitation: there is a burgeoning “middle-class”, but their lives do not compare to those in the West. So instead I want to assume a kind of “new imperialism” of big business without taking the time to explore it. What is more germane to our discussion is bringing up one more example of techno-business’ operative paradox. In The Silicon Valley of Dreams, David N. Pellow and Lisa Sun-Hee Park, point to the global health disasters from technological factories’ byproducts. The damning irony of this all is that this particularly company who makes semiconductors for use in the so-called “knowledge economy” intentionally obscure relevant data. The authors write, “in our interviews with workers…many reported that they were never give the results of the medical exams performed on them by corporate personnel…[which] appears consistent with the industry’s overall policy of restricting access to information” (Pellow and Park, 183). The producers of the channels of information sometimes have the most to hide in the new economy: where there labor comes from, whether outsourced or donated, what there profits are, and what havoc their factories wreak on their workers and the environment more broadly. The “knowledge economy” is inherently paradoxical, just like its chief operator, software. Information is hidden and so is labor, both in the sense that the material labor is sent to the periphery or the global south, and that free labor is expected and needed to sustain knowledge capital.

Conclusion
In identifying software as the hegemonic commodity form of the so-called new economy of immaterial labor, I have tried to ferret out the invisible aspects of work in order to isolate new sites of contestation. If we cannot come to grips the commodity, we cannot understand the ideology that permeates it. Without an interface to these sites of tension, we miss the larger conflicts of global culture more broadly. While software is a problematic text—in the sense that it is not just linguistic—there seems to be ways to read it as one. “Software” may be too broad to read comprehensively, interpreting it constructs an interface with larger mechanisms of economy, production, and ideology in ways that few other contemporary texts can. Though the free-software and open-source movements may not herald a critical investment in the knowledge economy, they do bring up important new problems. Berry and Moss end their own essay on saccharinely optimistic note, “This is a social-factory of immaterial labour where all of life—loving, thinking, feeling and sharing—is subject to the corruption of privatisation and property…there has to be a political dimension” (Berry and Moss, 3). However you feel about their prose, clearly inspired by Hardt and Negri, their final point is vital: a politics of digital culture has to be constructed and maintained. Though partisan blogs, meet-up groups, social participation and citizen news are all facets of this politics, they are only political, not a politics, in the sense that Galloway suggests in his reading of software. We cannot de-politicize the culture of the commons, but we neither can we blindly trust in it. This commons does not provide the hope of “info-communism” as Mueller mockingly suggests, but a new site for hermeneutic investigation. Technology is not a savior, it’s a battleground.
Works Cited


“FAQ”. Creative Commons. <http://wiki.creativecommons.org/FAQ#Can_I_use_a_Creative_Commons_license_for_software.3> Retrieved 1 May 2009


