

Introduction

Looking Forward and Backward

Heterogeneous Engineering of Social Media Software

sumoto.iki's web2diZZaster

Web artist sumoto.iki's "web2diZZaster" is a collection of bland, muted pastel images containing little more than rectangles and lines.¹ The images are unremarkable, even unattractive, and it is hard to determine what they represent. And yet, many of these images seem eerily familiar. A second glance reveals why: these muted rectangles take shape as common social media sites. Digg, the social bookmarking site, is identifiable by the peach tabs that indicate the number of "digs" that users have given to various stories. The tabs are empty, as is the rest of the page, but this largely empty frame is still recognizable. Myspace, the failed (and resurrected) social-networking site, is identifiable by its blue banner and log-in fields, and directly below it is Facebook's even more sedate (but much more successful) home page. YouTube is harder to recognize until the viewer sees the iconic red polygon and the two series of four rectangles where featured videos normally appear. Although its design has changed since sumoto.iki made this project, Twitter is perhaps the most recognizable because of its light blue field and narrow, prominent center column.

I am having trouble describing the momentarily unsettled response I had to sumoto.iki's art. However, after a moment of squinting at the

images and contemplating, I realized that sumoto.iki presents all these flagship social media sites *without any user content*. Without comments, videos, lists of friends, and editorial recommendations, these sites appear as “ghostly forms,” which sumoto.iki describes as a “first impression of a possible apocalypse where only HTML vestiges would remain inside a dehumanized network of all networks.” The disaster in “web2-diZZaster” is the social Web at the end of humanity. It is the tragic silence of a world without tweets, status updates, YouTube videos, diggs, and Facebook connections. When I viewed this art, all I had to go on were the surface manifestations of the most basic elements of HTML, PHP: Hypertext Preprocessor, Javascript, and CSS: div tags, positions and floats, colors, and lifeless scripts. Within the “dehumanized network” and without user-generated content, these social media sites appear as mere frames, and unappealing ones at that. Without content, these sites are lifeless shells. Without it, social media cannot work.

Of course, social media are working just fine precisely because users *do* contribute so much to these frames. Nearly a billion people populate the social network Facebook, creating constant streams of comments, links, “likes,” and applications. Twitter’s meteoric growth is also measured in user-generated content. Social bookmarking and link-sharing sites Digg and Reddit command millions of page views, and an upvoted link posted on their pages can drive traffic to websites. Amazon posts millions of user-written reviews of books and products. LinkedIn is filled with job-seekers and headhunters posting résumés and e-introducing one another. Flickr has billions of photos and comments, and Facebook’s servers contain the largest collection of user-uploaded photographs in the world thanks to its ownership of Instagram. And the company that is perhaps the exemplar of social media is Google, which relies on user-generated links, videos, social connections, and blogs to power its highly profitable search and advertising business. Investors who participated in the 2004 Google IPO have seen their investment grow substantially; the stock was offered at \$85 and as of this writing trades for nearly \$1,100. Although Facebook’s IPO was much less successful, it still generated \$1 billion. Twitter’s recent IPO avoided Facebook’s missteps to raise nearly \$2 billion. This rise in value comes directly from user-created content, all

within the opposite of sumoto.iki's "dehumanized" network. Perhaps we should borrow a phrase from networking company Cisco and call it the "Human Network."

Considering the history of the Web, it seems unlikely that social media would be such a commercial success. Just over a decade ago, the term "dot-com" (i.e., commerce on the Web) drew derision from anyone with an interest in business. The 2000–2001 financial/technology bubble burst resulted from irrational exuberance; investors in online commercial sites, such as Pets.com, lost millions of dollars when that business model failed to catch on. Direct-to-consumer sales of pet supplies, groceries, and gardening supplies had all the sustainability of a paper fire. In addition, investment in the networks themselves—that is, in the physical connections between sites—was drawn to oversaturated corridors, such as between New York and Boston. Just like the housing bubble burst of 2008, in the late 1990s there was a fiber-optic bubble.² When it burst, investors withdrew from the market almost immediately. According to PricewaterhouseCoopers, the first quarter of 2000 saw investment in IT peak at \$2.8 billion, and the first quarter of 2002 saw investment of only 10 percent of that peak level.³ Even in the post-Google IPO years, investment has come nowhere close to even 25 percent of the peak of the bubble years.

While the 2000–2001 market rejection of direct-to-consumer marketing of mass-produced goods and online commerce scared venture capitalists seeking to profit from the Web, another more sustained contemporary movement was attacking global capitalism, mass culture, and private property—and relying on the Web to do so. Writing about the Zapatista movement of the 1990s, Maria Elena Martinez-Torres notes that "a paradox has emerged from the revolution in communications: the same technology that has taken world capitalism to a new stage of development—corporate globalization—has also provided a significant boost for anti-corporate and anti-globalization movements."⁴ By co-opting the Internet as a space of spectacle and image politics,⁵ antiglobalization and progressive movements have been able to transmit their messages to worldwide audiences.⁶ Even in the midst of the dot-com euphoria of 1999–2000, protesters were able to use the Web to organize massive, coordinated demonstrations

against the World Trade Organization (WTO) meeting in Seattle. The actions of the estimated forty thousand protesters were supplemented with the advent of Indymedia, a user-led, anticapitalist news source that began as an alternative to mainstream coverage of the Seattle protests. These anticapitalist uses of the Web were engagements in what Nick Dyer-Witheford calls the “struggle for the general intellect.”⁷ Drawing on Karl Marx’s iconoclastic “Fragment on machines” in the *Grundrisse*,⁸ Dyer-Witheford argues that the Internet has simultaneously enabled extensions of the Taylorist domination of labor and the very means for labor to short-circuit global capital. On the one hand, the Internet might allow for “fast capitalist”⁹ flows of commodities and value realization, but on the other hand, it allows for the fast and space-eroding coordination of protest.

Here, we see two interweaving movements going online. On the one hand, capitalism’s cycles of boom and bust came to the Web as the irrational rationality of herdlike investment movements seized on cyberspace as the next great marketplace.¹⁰ On the other hand, the strong counterhegemonic possibilities of the Internet and Web were being explored by an increasing number of “hacktivists,” cybersocialists, and antiglobalization agitators seeking to appropriate the Web as a tool of revolution.

This seems to be an unlikely place for social media capitalism to thrive. However, in 2004, technology book publisher Tim O’Reilly and journalist and technology blogger John Battelle confidently stood on stage at the inaugural “Web 2.0 Conference” in San Francisco and argued that Web-based commerce was making a comeback.¹¹ Their proof came from companies that recognized the chaotic, unpredictable nature of user-generated discourse on the Web and were able to create sites that harnessed this “collective intelligence.” In O’Reilly and Battelle’s vision of Web 2.0, companies that were appropriating the socially articulated energies, passions, and labors of users—wherever those users might go with them—were not only surviving in the world of online commerce; they were building new media empires.

In this milieu, the practices—but certainly not the anticapitalist politics—of a user-generated site such as Indymedia are replicated

within the user-generated, for-profit CNN iReport, where “citizen journalists” produce and share news stories. The Zapatistas and their supporters are now welcome to connect via Facebook at the Chiapas Project.¹² The spectacle of mass protests drives news coverage, and thanks to the personalization of Google News and user-generated services Reddit and del.icio.us, we can easily keep up with the latest developments. Dyer-Witheford’s manifesto *Cyber-Marx* is available on Amazon; those undecided about spending \$25 on it can consider several glowing user-written reviews that appear on the listing. In short, social media are the corporate response to the mass creativity, collaboration, and desires of networked peoples. It is a tacit admission by large media companies: when given a choice, people prefer content produced and recommended by themselves and their friends to that recommended by editorial authorities. They are leery of mass culture and globalized corporations, so they seek to create their own culture.¹³ They express political opinions and offer frank assessments of commodities, corporations, and states, and they openly share these opinions with their friends and colleagues.

However, it is not as though new media capitalists are simply stepping aside and allowing users to lead the way, let alone slough off capitalist media production. Clearly, social media outlets are new media capitalism’s attempt to absorb and capture this explosion of user-generated content as objectified surplus value. Whatever the form—from fandom to antiglobalization manifestos—if the user-generated content occurs within what sumoto.iki calls the “ghostly frames” of many social media sites, it is being exploited for profit.

Sumoto.iki’s artwork, then, is a useful illustration of the central argument of this book: social media need to be understood not just as a collection of sites that place the users at the center, nor just as a radical reform to the top-down, authoritarian model of mass media. Social media also have to be understood as software engineered to privilege and enhance certain users while closing off others. To be sure, the “ghostly frames” that sumoto.iki depicts are notable for their absence of user-generated content, but they are also notable because they *are in fact a framework for users to inhabit and populate*. This software

framework contains a contradiction: on the one hand, social media allows for users to “be the media” and thus influence mass culture; on the other hand, social media sites are rigidly hierarchical, allowing certain uses and discouraging others, while site-owners constantly watch users’ movements and exploit users as what Tiziana Terranova has aptly described as “free laborers.”¹⁴ As she argues, free labor is rife with contradictions: it is freely given yet exploited; it is done for love, yet hypervalorization haunts and directs it; it is work, but it is play. Similarly, the “ghostly frames” sumoto.iki reveals are simultaneously sites of user freedom and rigid structures of exploitation. Social media must be understood along these lines, and to do so, we have to turn to computer architecture design as a key source for this contradiction. As Lawrence Lessig argues, “Code is law.”¹⁵ The architecture and software matter.¹⁶

However, despite the limitations of and problems with the social media architecture, what do we make of social media’s progressive aspects, started by the Zapatistas and the anti-WTO movement and progressing to the Arab Spring and Occupy Wall Street movements? It is myopic to only talk of exploitation while ignoring the power of new social movements. Here, I suggest that such movements indicate the pressing need for users to take an active role not only in producing online content but also in shaping the structure of the sites themselves. The fact that Occupy and the Arab Spring have achieved what they have *despite* the surveillance and exploitation of social media makes me wonder: what if social media were built to help *advance* activism and politics rather than constrain them? Again, the architecture matters. To alter social media to meet this goal, users and activists must be engaged early and often with the development of social media architecture and software to avoid many of the social inequalities and problems with privacy we currently are witnessing. Indeed, they would have to reverse engineer social media. I want this book to help point the way toward a better Web, one designed for progressive politics. In other words, I outline ways in which the structures and cultures of new media capitalism and social media software can be reverse engineered, hacked, altered, and maybe even renewed.

Software Studies and Engineering Metaphor(s)

Thus, this book is a contribution to the growing field of software studies. In the words of Rob Kitchin and Martin Dodge:

Complementing the work of computer scientists on the mechanics of software development and human computer interaction, and research on digital technologies more generally, social theorists, media critics, and artists have begun to study the social politics of software: how it is written and developed; how software does work in the world to produce new subjects, practices, mobilities, transactions, and interactions; the nature of the software industry; and the social, economic, political, and cultural consequences of code on different domains, such as business health, education, and entertainment.¹⁷

Software is ubiquitous. It structures many of our spaces, mediates much of our communication, helps shape our collective and individual memories, and underpins global economics. Software is an obvious part of our daily lives as we use computers and smartphones, but it is also a hidden substrate operating out of view: running automobiles, controlling infrastructure, algorithmically calculating credit and finance, storing data on myriad human and nonhuman flows, and collecting evidence of past behaviors. It provides us with new ontologies and epistemologies as we ponder what it means to be human in an age of smart machines and smarter phones.

Software studies is a critical interrogation of this multifaceted object. Theorists and scholars working in this field study how power operates in relation to software: who writes it? Why? To what purposes? What are the conditions of its production? How does it shape uses? What is its architecture, and why was that architecture selected over competing ones? How do users reappropriate, extend, build on, or break it? What can code, layers of software, the software/hardware relationship, network topologies, and the software/user relationship tell us about our assumptions about subjectivity and identity? Software studies does more than just analyze lines of code; rather, it evaluates

software as part of larger contexts, such as culture and history. Done well, a critical study of software can tell us much about the world around us.

Most histories of this new field trace it back to Lev Manovich's 2000 book *Language of New Media*, but its roots run deeper than that. Even when we limit the field to work produced after roughly 2000, there are many critical studies of software: philosophies of object-oriented programming¹⁸ and computationalism,¹⁹ hacking,²⁰ semiotic analyses of interfaces,²¹ artistic reinterpretations of software (such as sumoto.iki's, described at the beginning of this chapter), cultural and technology studies,²² legal analyses,²³ historiography,²⁴ spatial theorizations,²⁵ and critical code studies.²⁶ There is no single agreed-upon object or method of study, but this variety is a strength of the field; given the ubiquity of software, the field welcomes new works on new objects.

To this end, as a contribution to software studies, this book draws on three engineering metaphors, using them as methodological windows into social media software: software engineering, reverse engineering, and heterogeneous engineering.

Software Engineering

When studying software, an intuitive place to start is by running software on a machine in front of us. In the case of social media software, then, one might sign up for an account and critically consider the pedagogy and structure of the interface. I certainly have done this. However, following Matthew Kirschenbaum's warnings against "screen essentialism" and "medial ideology,"²⁷ I also want to extend such analyses by considering how software is made. What are its logics of production? How is the labor of designing, coding, testing, distributing, and using divided among people? Which materials and discourses make up software production and use? Who benefits from its production? How are ideal users constructed through its design, and how do concrete users deny that construction? This discussion takes us into the realm of software engineering.

Since the late 1960s, software production has been dominated

by the engineering metaphor. Software producers first debated, and then took up, the metaphor at a 1968 NATO-sponsored conference in Garmisch, Germany, titled “Software Engineering.” As the conference proceedings note, “The phrase ‘software engineering’ was deliberately chosen as being provocative, in implying the need for software manufacture to be based on the types of theoretical foundations and practical disciplines, that are traditional in the established branches of engineering.”²⁸ The engineering metaphor provided software producers with methodologies from more established fields, such as metallurgical and electrical engineering. In addition, rather than conceiving of software production as an art form or the production of knowledge for knowledge’s sake, the engineering metaphor oriented producers toward making objects for use and thus for users/customers. This producer/consumer orientation was compatible with capitalism and with software’s newfound independence from hardware (largely because of IBM’s “unbundling” of the software previously included when customers leased its machines).²⁹ Also compatible with capitalism was the engineering metaphor’s emphasis on the management of labor; as it is described in professional documents, such as the *Guide to the Software Engineering Body of Knowledge*,³⁰ and by such organizations as the Institute of Electrical and Electronics Engineers (IEEE), the Association for Computing Machinery (ACM), and the Accreditation Board for Engineering and Technology (ABET), engineering post-Garmisch is marked by hierarchical control and the division of labor to produce software commodities.³¹

To be sure, there have been alternatives and challenges to the engineering metaphor.³² However, it does provide software studies with a useful language with which to perform critical analysis. In a chapter in a key text for the field, *Software Studies: A Lexicon*, Andrew Goffey notes, “Software engineering . . . concerned as it is with the pragmatic efficacy of building software for particular purposes, might appear to offer a better starting point for factoring culture back into software.”³³ Goffey goes on to note that the basics of software engineering must be explored to complete this factoring. Similarly, Nick Montfort and Ian Bogost argue that software engineering’s emphasis on code and the organization of labor in the production of code provides a

rich set of concepts for a critical study of software platforms.³⁴ Indeed, following Kitchin and Dodge, if software studies “focuses on the etiology of code,”³⁵ then considering a dominant paradigm of software production—software engineering—should be fruitful. Throughout this book, I use such software engineering concepts as *architecture*, *implementation*, *abstraction*, and *decomposition* as entry points into social media software, specifically in terms of the ways in which social media software is produced, the relationship between producers and users, and the ways in which users are allowed to produce the content within social media software frames.

Reverse Engineering

The next engineering metaphor I draw on is reverse engineering. Kathryn Ingle offers a brief definition of this process: “If forward engineering is the traditional process of moving from high-level concepts and abstractions to the logical, implementation-independent design needed in a physical system, then reverse engineering is the design analysis of the system components and their interrelationships within the higher-level discrete system.”³⁶ While software engineering is the production of an abstract architecture followed by its implementation, reverse engineering starts with the final, implemented product and takes it apart, seeking clues as to why it was put together in the way it was and how it fits into an overall architecture.

I see three good reasons to engage with this metaphor. First, reverse engineering helps when we are confronted with closed code and proprietary formats. As such, it is a practical move in an age of Digital Rights Management (DRM), compiled binaries, and vicious lawsuits. If we are denied access to source code, or if software is running on a far-off server, we have to speculate on how a particular instance of software works. Drawing on knowledge of software engineering, we can do this by theorizing which sorts of uses and requirements the software is meant to meet and how the architecture and internal implementation of it might meet those needs. These may be visible in the interface; in this case, using the artifact and seeing how it shapes uses (and therefore users) through its design elements can lead us to

speculate as to why some uses are privileged while other technically and equally possible uses are denied. Outside the interface, we can also turn to white papers, user manuals, press releases, blog posts, and news stories to glean insights into why coders and architects built their particular closed software system and what their goals may have been. Reverse engineers have used all these techniques for decades.

Second, to reverse engineer is also to move back through time. Good reverse engineering takes temporality—if not history—seriously. “In spanning the life-cycle stages,” write Elliot Chikofsky and James Cross, “reverse engineering covers a broad range starting from the existing implementation, recapturing or recreating the design, and deciphering the requirements actually implemented in the subject system.”³⁷ Reverse engineering is thus a forensic process,³⁸ taking small parts of a system and theorizing as to which abstract purposes their creators may have intended. Concrete, particular instances of software are the starting points for analysis that drives toward an understanding of larger sociotechnical systems that precede it. And, in addition to tracing the associations from concrete implementation back to abstract architecture, this emphasis on temporality also helps us deconstruct the metaphors and language of social media and technology. Many scholars and popular writers place far too much emphasis on newness and novelty. Such terms as “new media” and especially “Web 2.0” posit a radical break with the past, eliding—even denying—history, even though the techno-utopian worship of the new has been with us for at least all of modernity.³⁹ This sort of thinking gives rise to the breathless proclamations of “revolutions” of “new technologies.” However, if we take the reverse engineering metaphor to heart, we ignore grand “revolutions” and instead do the practical work of tracing technologies back through time to uncover their associations with prior technologies and practices. The names and descriptions of current technologies often draw metaphorically on older ones, and tracing these metaphors back through time reveals hidden histories that get lost when we declare that our current time represents a radical break with the past.

Finally, reverse engineering provides a healthy perspective on the well-worn (and yet always compelling) debate about structure and

agency, particularly in terms of technology's relationship to agency. If one takes reverse engineering seriously, "every time you have a better idea you will consider all the positive design aspects before condemning an entire product."⁴⁰ This approach is pragmatic, dealing with the technology we actually have rather than dreaming of utopia. Contemporary social media have many advantages over older forms of mass media: protest movements, such as the Arab Spring and Occupy Wall Street, have used Twitter, Facebook, and YouTube to organize their movements and promote their ideas when other traditional media organizations have overlooked them. We have to acknowledge these progressive facets of social media, but we do not have to accept the system wholesale. Reverse engineering is useful here, because it is a critical dissection of existing technology with the goal of building a better system. While "positive design aspects" are not essential properties of a technology—that is, they are a matter of context and social structure—there is no reason they cannot be excised from the current architecture and shaped into new forms of media more compatible with radical democracy. I find reverse engineering's emphasis on actually existing, material technology useful to this end.

Heterogeneous Engineering

So now we have a few ways to answer some of the provocations raised by sumoto.iki's work, and we have a place to start for a critical study of software centering on the software engineering metaphor. However, we have to place these post-human frames and the processes of their production within an existing political-economic human context. To do so, we need a theory and a normative stance. Here, I turn to the science and technology studies theory of heterogeneous engineering, a subset of actor-network theory⁴¹ that is certainly useful to software studies.⁴² Perhaps the biggest champion of the theory of heterogeneous engineering as well as the provider of a heuristic method to apply that theory is John Law. Law argues that social analysis should start with the "metaphor of *heterogeneous network* . . . a way of sug-

gesting that society, organizations, agents, and machines are all *effects* generated in patterned networks of diverse (not simply human) materials.”⁴³ Law wants to break concrete social and technological totalities into their constituent elements to discover “the complexity and contingency of the ways in which these elements interrelate” and document “the way in which solutions are forged in situations of conflict.”⁴⁴ “Solutions” in heterogeneous engineering are not simply the best possible forms of technology or techniques for the job at hand. Rather, they are “associations” of technological and cultural elements, joined together despite myriad forces that constantly threaten to *dissociate* the object in question. Associated elements are “difficult to tame or difficult to hold in place. Vigilance and surveillance have to be maintained, or else the elements will fall out of line and the network will start to crumble.”⁴⁵ Every heterogeneous element is resistant to being put to the intended purpose, and anyone claiming to be an engineer must recognize the agency of the nonhuman.

Thus, the task of the critic is to start with the totality, seek out its constituent processes, and articulate the connections and relationships among them to (re)build a complex-concrete whole. The critic must seek out processes that are hidden, implied, or on the margins and connect them to those that are obvious and privileged in common-sense discussions of the object. Heterogeneous engineering is especially valuable for our purposes here in its conceptual grasp of technology; it reminds us (1) of the highly contingent nature of any technological artifact; (2) that “knowledge” and “technique” are the result “of a lot of hard work in which heterogeneous bits and pieces—[such as] test tubes, reagents, organisms, skilled hands, scanning electron microscopes, radiation monitors, other scientists, articles, computer terminals, and all the rest—that would like to make off on their own are juxtaposed into a patterned network which overcomes their resistance”;⁴⁶ and (3) that very often these heterogeneous bits and pieces must be *engineered* by some actor or organization.⁴⁷ The heterogeneous engineer must be an engineer of *differance*, not only associating the obvious elements but also deferring, eliding, displacing, and denying deviant elements, the Other, heteroclites, and noncoherences. The heterogeneous en-

gineer works with the immaterial and material, the abstract and the concrete, none of which is easily or necessarily associated together. If the engineer is good enough, then the system appears to be second nature, given, and immutable; the contradictions and resistances within the materials it comprises are hidden, and we celebrate the system as evidence of linear technological progress. Very often history is told as the story of these successful projects, while failures and contradictions (that is, messy phenomena that trouble our lovely narrative of linear and logical technological progress) are pushed outside the margins of historiography.

This theory reminds us that we cannot simply uncritically accept the proclamations or products of software (or reverse) engineers. We have to empirically examine them, produce litanies⁴⁸ of human and nonhuman objects, and critically trace their associations. We have to ask what is hidden within these engineering metaphors: beneath the surface of the interfaces driven by Asynchronous Javascript and XML (AJAX),⁴⁹ among the racks of server farms, inside client devices, in the wires and radio signals of network connections, and between the lines of IPO-filing documents, roadshow proclamations, and tech fan blogs. To analyze social media software and culture, we have to look forward and backward, in the abstract and concrete, in the now and in the vestiges of the past. We see contradictions between reverse and “forward” software engineering: there are competing movements through time, associations and dissociations, dreams of future solutions and the frustrating structures of the past. We find moments when the software misbehaves, when a security hole appears and a viral infection sneaks in, or when the subject so often Othered by engineering—the user—asserts him- or herself and disrupts the system. Heterogeneous engineering troubles the neat claims to agency made by engineers by hinting at the agency of nonhuman elements, such as machines and code. Heterogeneous engineering reminds us that an electron out of place can doom a network or that a shift in social politics can cause a reinterpretation of the ontological reality of a machine. Ultimately, then, this theory acts as a counterweight to the triumphalism of the software engineering literature.

A Normative Intermezzo

Finally, before I turn to chapter summaries, I need to take a moment to lay out a normative position. Because hegemonic social media is produced within informational capitalism, the inherent inequalities of that system are translated, ported, or simply replicated within new media software. As a growing number of media scholars, activists, and social media users now recognize, new media capitalism as practiced by such sites as Facebook, Google, and Twitter has had terrible consequences: it reduces online interaction to binary declarations of like-it-or-not consumer choices; it hides and inures us to the surveillance systems operating underneath its surface; it relies on the free labor of its users to build its content while the site owners make billions by selling user data and stock in their companies; its centralization (which always seems to be part of the political economy of capitalism, despite the repeated cries of creative destruction, disruption, innovation, and competition) provides an all-too-easy means by which states can gather data on citizens; and it promotes a culture of anxiety and immediacy over depth. In short, contemporary social media hardly seems compatible with democracy, and because of this issue, I need to move past the point where heterogeneous engineering analyses tend to leave off:⁵⁰ at the question of “what is to be done?” As Marx has famously argued, “The philosophers have only interpreted the world, in various ways; the point is to change it.”⁵¹ After we have explored software engineering, reversed it back through histories of computer science and modernity, and discovered the hidden Others and forces that threaten to dissociate that totality, we then may glimpse new potential associations of processes, technologies, and subjectivities. We begin to realize that these new associations can be heterogeneously engineered to reflect different values—specifically, those of the much-longed-for Habermasian public sphere of democratic debate and decision making. We start to see paths forward out of the seemingly unassailable realms of power into new spaces of freedom. Thus I turn, at last, to the normative stance of critical political economy of communication, particularly the Marxian strain, as I analyze the engineering of social media software. I also rely on

critical Marxian analysis as I reverse engineer this system and look for something better.

Thus, this book critiques social media by rigorously analyzing the ideas of social media engineers, reversing them to trace the path from concrete software to abstract desires, seeking contradictions among them, noting where meanings shift as audiences and interests shift, contrasting them with the underlying architecture, and looking for new, progressive possibilities and shapes within this structure.

The Plan of the Book

Chapter 1 begins this process by exploring the emerging phenomenon of the socialbot. Throughout many critical examinations of social media, there is a nagging sense that human activity is reduced to datasets within the templates of such sites as Facebook, Twitter, and Google. Socialbots—automated social media profiles that tweet, like, and friend—are concrete evidence that this nagging feeling is not unfounded. Socialbots are a reflection of our activities within social media; for these machines to work, we ourselves have to be trained to be machinelike as we produce images of ourselves via publicly confessed declarations. In keeping with the idea of reverse engineering as looking back, this chapter links contemporary socialbots to the construction of the ur-socialbot, Alan Turing's Universal Machine, a seminal idea in early computing. Turing's machine and his famous Turing Test of computer intelligence are founded on his idea that the human mind can be understood as a finite, discrete machine. As such, if various "states of mind" of humans can be encoded, they can be manipulated and imitated by computers, even to the point where a computer could appear intelligent by engaging in a conversation with a human. Chapter 1 then links Turing's work with the emerging critical literature on "noopolitics" (the politics of *nous*, or the mind). Next, it explores the cultures of transparency and authenticity in contemporary social media. Social media's "culture of confession" is producing a massive dataset of the internal, discrete states of mind of human beings. Once this codification is done, then the Universal Machine (in this case, socialbots) can imitate the human in a modern-day Turing

Test. Socialbots' success in this regard certainly raises the specter of machine intelligence. But more importantly, socialbots are emerging as a technology of noopower (i.e., power over thoughts) capable of shaping opinions and interactions within social media.

In Chapter 2, I explore a power imbalance between social media users and owners, particularly as the two meet in the "Web as platform." Looking backward once again, I link a division of labor apparent in social media to the internal divisions of labor built into computers, best exemplified by the Von Neumann Architecture. This architecture, which divides the processor from memory and allows the storage of programs, has been replicated in social media. In social media, users are granted the power of the processor: they are allowed to process digital objects, liking this, tweeting that, rating, commenting, and sharing. This is "collective intelligence" in action—at least as that intelligence is modulated within social media. However, social media companies maintain a strict division between the user/processor and memory, the other side of the computational equation. Social media companies derive power from storing the results of users' affective processing in archives. These archives can then be drawn on to construct "facts" about users: which consumer objects they desire, which trends they are tuned into, even whether they might agitate for political change. These "facts"—artifacts of a particular "harnessing" of collective intelligence—can then be sold to marketers or surrendered to states. This division between the processor and the archive helps maintain the social structure of social media.

Chapter 3 continues on the theme of the division of labor, this time considering how the software engineering practice of abstraction has shaped the processes of user labor. The central questions I consider here: how "user-led" is user-led production? Is there a limit to what users can do within social media? Who sets that limit? To answer these questions, I propose the concept of the "real software abstraction," a synthesis of the software engineering practice of abstraction and Marx's concept of the real abstraction. Starting with the seminal work of software engineer Frederick Brooks, I argue that the predominant architecture of social media has been drawn from the methods of managing labor in the production of software. Brooks's

work on the IBM System/360, detailed in his book *The Mythical Man-Month*, was based on his division between architecture and implementation. Architecture is a high-level abstraction, the blueprint of the system. Implementation is the labor of coding to the architectural specs. Drawing on the work of Alfred Sohn-Rethel and Alberto Toscano, I link architecture to Marx's real abstraction, arguing that software architecture may be an abstraction, but it has real, concrete effects in the world, particularly in how it enables the disciplining of coding labor. I use a case study of Myspace and Facebook to illustrate instances where this practice has been transferred from firm-based software production to social media. Social media site owners create real software abstractions and allow their implementation to be carried out by the free labor of users.

Chapter 4 is part of a larger conversation about networking protocols and standards. Many narratives discussing the rise of the Internet and Web as new forms of media center on the production of shared standards and protocols, such as Transmission Control Protocol/Internet Protocol (TCP/IP) and Hypertext Transfer Protocol (HTTP). In turn, social media are seen to be animated by the Application Programming Interfaces (APIs) of Google, Facebook, and Twitter. In Chapter 4, I complicate this narrative of standards by looking at the history of *advertising* standards, especially those developed by the Interactive Advertising Bureau (IAB). Our current social media have been designed to adhere to the standards produced by the IAB. Through these standards, social media sites link heterogeneous user-generated content of all kinds (from posts to status updates to the affective labor of constituting social networks) to networks of marketing and advertising. Advertising standards have thus played a major role in shaping the surveillance-based business models of such social media sites as Facebook, Twitter, and Google. In addition, in Chapter 4, I synthesize many of the arguments made in the previous chapters by outlining the subject imagined by the IAB and marketing-centric social media: the *sovereign interactive consumer*. This subject is the prime abstraction of social media. The IAB and social media sites build their discourses of self-regulation on this abstract consumer.

Of course, these chapters are quite pessimistic. As a way out, there

is the normative intervention to consider: what is to be done? The remaining chapters use the contradictions and gaps exposed in the previous ones to plot actual and potential ways forward. Chapter 5 details Wikipedia's evolution. This now-famous encyclopedia relies on many of the processes underpinning social media as a whole: it harnesses collective intelligence, it provides a platform on which users can build new applications, and it relies on a client-server architecture. And yet, Wikipedia is radically different from other social media sites. This chapter explores Wikipedia's difference in terms of an important event early in its history, the "Spanish Fork." In 2002, Wikipedia users in Spain left the site in droves in response to the potential use of advertising on the site. The leaders of the Spanish Fork believed that advertising would reflect Wikipedia's exploitation of their labor, so they transplanted the entire Spanish-language version of Wikipedia to their own servers hosted at the University of Seville. Clearly, this Spanish Fork was a labor strike; the participants in the strike were able to heterogeneously engineer a "class for themselves" out of the typical social media capitalist production of the user "class in itself." Their encyclopedia, *La Enciclopedia Libre Universal en Español*, became a rival to the nascent Wikipedia, forcing Wikipedia's founders to establish it as a nonprofit site rather than as the commercial venture they originally intended. I see this historical moment as proof that, with the right mix of technological and communication infrastructure, leadership, and discourse, users can recognize their position as free laborers and strike against social media exploitation.

Finally, in Chapter 6, I return to the reverse engineering metaphor. In the literature on reverse engineering, the impulse to do the work of disassembly and documentation is to improve the bottom line: products can be improved, technology and infrastructure can be maintained instead of thrown out when they break down, and new intellectual property can be claimed. In contrast, as a critical student of online media, I aim my work toward improving our media system, which is a fundamental element to democracy and citizenship. To this end, I offer a "Manifesto for Socialized Media" in two parts. The first is a design specification—or, to be more truthful, a *design speculation*—for a socialized media system that is decentral-

ized, transparent, encrypted, antiarchival, stored on free hardware, and geared toward collective politics over atomization and depth over immediacy and surfaces. The second part is a brief survey of current efforts to implement and realize such a system by heterogeneously engineering social media alternatives. Because I play the role of a designer, my ideal socialized media system remains just that—an ideal. However, looking at the production of media systems meant for activists, it is clear that serious efforts are underway to realize an ideal system, such as the one I have proposed. Moreover, because software is obdurate, and because it often asserts its agency by shaping and determining actions, the encoding of progressive politics into socialized media systems is also the preservation and extension of antistate, antipower, and anticapital politics.

This is the plan of the work. However, even after all this, social media will no doubt appear to be a complex, almost overwhelming phenomenon—and moreover, by the time this book appears, no doubt many elements will have changed! As Geert Lovink admits, “The object of study is in a permanent state of flux and will disappear shortly—the death of everything cannot be denied.”⁵² Book-based case studies and analyses of networked media, however timely, fall behind network time. No book or single author can hope to grasp social media in its entirety. But Lovink suggests a “possible way out” of this quandary: “the development of critical concepts that migrate from one generation of applications to the next, without falling back on speculative theory that merely celebrates the liberating potentialities of buzzwords while waiting to be translated into market value.”⁵³ I hope the process of reverse (heterogeneous) engineering and its associated concepts I offer here are a contribution to the work of activists, academics, and users who seek to change social media for the better.