The definitive version of this paper will appear in the edited collection *Compromised Data*, Greg Elmer and Ganaele Langlois, editors.

Earlier versions of this paper were presented at the Compromised Data symposium (http://www.infoscapelab.ca/news/oct-28-29-colloquium-compromised-data-new-paradigms-social-media-theory-and-methods) and at the 2014 Cultural Studies Association conference.

Critical Reverse Engineering: The Case of Twitter and TalkOpen

Social media present the critical media scholar with a quandary. First of all, a growing number of scholars are faulting sites such as Twitter for their erosion of privacy, their desire to turn every thought into monetizable, quantifiable evidence of individual consumer desires, their reduction of human interaction to 140 character chunks, their promise of openness while viciously protecting their algorithms and datasets with lawsuits, and their susceptibility to (if not downright compliance with) government surveillance programs. On the other hand, it is difficult even for the most hardened social media critic to deny the pleasures of using these systems, of making friends and connections, and of bathing in a stream of new ideas as it flows across the screen. Nor can the critic deny social media's utility in political, cultural, and social organizing, their centrality in the erosion of mass media gatekeepers, or their part in the larger epistemological shift that is happening due to big data.

The critic's relationship to social media is thus compromised in many senses. For some, the sites compromise critical inquiry just as a virus compromises bodily integrity: critics become infected with laudatory love for viral videos, memes, trends, and the quantified self, ignoring the ways in which such practices reify existing power relations. For others, such sites are compromised, full stop: they cannot be trusted. For them, the dual-headed surveillance system, comprised of states and transnational corporations, is simply too dominant in corporate social media, and the only clear course is to get out. Revel or get out: it appears that these are our options, and these options are largely reflected in the academic literature on social media.

However, there is another compromise to make with social media: build your own. We often forget that we're working with a network of universal machines. As the old saying goes, another world is possible.

To this end, I propose a methodology of 'reverse engineering' to critique contemporary social media software as well as establish criteria for alternatives to that software. Drawing on the reverse engineering literature as it is found in fields such as engineering, law, and economics, and inspired by critical science and technology studies and software studies, I will argue that this approach is valuable because of four orientations: the pragmatic, the genealogical, the legal, and the normative. These orientations can guide the critic through the false choices of technophilia or technophobia, uncritical love or endless critique with no way out. To illustrate reverse engineering as a critical method, I will use the examples of Twitter and TalkOpen (a short-lived alternative to Twitter), specifically the ways in which TalkOpen reverse engineered Twitter's interfaces, architectures, and imagined users. Ultimately, I argue that instead of simply giving in or getting out, critical social media scholars have a chance to take their theories and put them to the test by constructing (or at least using) new social media systems that challenge the power of centralized, corporate social media.

Reverse Engineering and the Humanities

There are a variety of definitions for reverse engineering across the literature I'm working with here (engineering, law, and economics). Drawing on them, I would argue for this basic conceptualization: *Reverse engineering is a method of producing knowledge by dissociating human-*

made artifacts. This knowledge is then used to produce new artifacts that improve upon the old and yet also bear a relation to the old.

In other words, during reverse engineering, a human confronts an object. The object bears information: how it was assembled, the material used in its production, the choices made to realize it. This information gestures towards – but can never completely reveal – a process of 'forward engineering' that implemented the object. The reverse engineer brings tacit knowledge (these days, often produced in him/her by way of training and study) to bear on that object in order to open it up, take it apart, probe it, test it, stress it, break it, peer inside, and learn how it works. In other words, the reverse engineer uses many techniques to read information the object bears. The object thus mediates a relationship between the producer, who used tacit knowledge to produce the object, and the reverse engineer, who uses tacit knowledge to take it apart. Such movements across the engineering processes are also oscillations between the concrete and the abstract, objects and theories: the object mutates, going from conception to concrete instantiation to conception again.

After reading this object-text, the reverse engineer creates new knowledge and information which can be used to 'forward engineer' a new object. The process of reverse engineering helps create new tacit knowledge in the reverse engineer, and very often the process begets new information in the form of documentation and manuals. When this knowledge and information are used to build a new object, that new object bears both the traces of the original object of inquiry and the traces of the reverse engineer-cum-forward engineer's desires and intentions. The new object might work in concert with the old; it might be meant to replace it; or it might be something that offers new affordances that on the surface bear little relation to the old object. However, it is always internally linked to the old.

Reverse engineering is, of course, an essential process in any engineering field. Pedagogically, it is one of the key ways that engineers learn their craft. Economically, it provides a tool for firms to either compete with other firms, maintain their stocks of artifacts (Ingle, 1994), or generate monetizable information about their artifacts (especially in the form of patents). Legally, as I will explore below, it provides a limit on laws that grant protection to trade secrets, and thus for engineers reverse engineering is a sort of promise: go ahead a build something, but know that others will take it apart.

Given the 'new materialist' turn in the humanities, marked by theories such as Actor-Network Theory (ANT), Object-Oriented Ontology (OOO), and assemblage theory, and given the turn to the digital and to computation in the humanities (i.e., the digital humanities), I would argue that reverse engineering is not only valuable to fields like engineering or business; it also provides a wealth of methodological, theoretical, and practical paths the critical humanist can take. There are many ways to critically delve into technoscience. Bruno Latour famously tells us to 'follow the actors' (Latour, 1987). This is of course a call for ethnography, for the direct observation of the actors who do the associative work to construct technoscience systems and discourses. But, as Susan Leigh Star suggests, we can also conceive of an ethnography of objects (Star, 1999). The technical objects we confront contain within them traces of the associations that brought them into being. They are, to use a term from John Law, 'punctualizations' standing in for the networks of materials, politics, and discourses that constantly structure them (Law, 1992). Those objects that appear stable, become infrastructural, or even fade into the background can be fruitfully and critically reverse engineered to trace their inner heterogeneity. In this sense, to return to Latour's aphorism, we can 'follow the actors' by turning to the artifacts they produced and looking for traces of their activities, ideals, and politics within them.

To build on this, I next turn to the four orientations found in the reverse engineering literature: the pragmatic, the genealogical, the legal, and the normative, connecting these orientations to a possible critical methodology.

The Pragmatic Orientation

Reverse engineers are pragmatic: they consider the technology they have at hand, not an ideal technological assemblage. They do not simply throw away older or poorly designed technology, but modify, shape, and alter it. They accept the positive side of a technology while working against the bad. As Katherine Ingle puts it, when a you're a reverse engineer confronted with a technical object,

> every time you have a better idea you will consider all the positive design aspects before condemning an entire product. And for every time you have had a bitter experience with a real lemon you will also roar. The roaring will die down when you realize that you have the power and skill to change this ugly duckling of a design into a graceful swan (Ingle, 1994, p. 2).

In other words, for the reverse engineer, technical and design shortcomings are not excuses to throw away an object, but to make a better one.

For example, in terms of software, most software used in firms is bought, not built. Moreover, most of the costs of software use center on maintenance. When a firm has software that does not meet the firm's needs or requirements, it does not just make new software. Very often, it modifies the old. This practical approach of course entails reverse engineering.

A great deal of this process entails the discovery of 'facts' discerned through thick, close, detailed empirical analysis of technical objects. How does the technology work? What is it comprised of? Who built it? What might their intentions have been? What can we do with it? However, this is not to say that reverse engineering is a simple positivism, a search for one right way to construct a technology. Rather, reverse engineering holds that there are many ways to solve a problem. Unlike positivistic science, which in the ideal seeks the one right, verifiable answer to a question, engineering is a field of heterogeneity. There are many multiplicities involved: multiple problems, multiple users, multiple implementations, multiple patterns, and multiple products of multiple reverse engineering approaches. As software engineers Baxter and Mehlich put it, when one reverse engineers software code, 'different abstract concepts map to the same code within one application' (Baxter and Mehlich, 1997, p. 105). In other words, code is not only heterogeneous in how it operates across various settings and times, as Adrian Mackenzie (2006) argues; it is also heterogeneous in terms of the abstract, theoretical ideas we might have about its functions, as well as in its theories of the user/subject. In their prescription for reverse engineering, Baxter and Mehlich (1997) argue for the production of an abstract specification from a concrete system, but they recognize that such abstractions never map directly back onto the original designer's intentions. There are too many permutations, oscillations, algorithms, optimizations, paths through lines of code, subjective design decisions, use cases, subjective requirements, and arbitrary technical choices involved at every stage of the engineering process to ever hope for a complete description of an software system. Unlike positivistic science, however, reverse engineers by and large recognize this, accept this, and simply go for what works (by whatever standard of judgment they've decided upon).[1]

Reverse engineering provides this pragmatic approach to the techno-scape we confront: look to the technology you have in front of you. Learn it. Work with it. Tinker. Alter it. Shape it. Imagine possibilities, but don't hold to some ideal best way. Thus, reverse engineering avoids overly idealist approaches to socio-technical problems. It helps us avoid moralizing discourses about technologies, as in 'we need a wholly new form of technology to solve problems X, Y, and Z,' as if the latest version of the technology will magically erase history, habit, and error. Such hope for novel, 'revolutionary' technologies is rendered naïve by the reverse engineering perspective. I will explore this further below.

[1] To be sure, this pragmatism only goes so far. In that same Baxter and Mehlich essay, they share a lamentation: if only forward engineers would document every decision in a machine-readable, consistent fashion, we would never need reverse engineering! Given both the existence of proprietary software and the multiplicities of interpretation of designs and code – just to name two problems – their solution seems quite far-fetched, to say the least (Baxter and Mehlich, 1997, p. 106).

The Genealogical Orientation

Reverse engineers also consider the historical development of a technology. First of all, as its name implies, reverse engineering is forward engineering in reverse. The hegemonic engineering process involves starting with an abstract architecture, decomposing it into components, and then implementing each component in whatever material it will be made out of, including code. This takes time, and moreover involves many subjective decisions, heuristics, accidents, coincidences, and acts of labor. This process is one that can be comprehended temporally – even historically – and one way to do this is to start with the artifact and reverse the process.

As Baxter and Mehlich state, 'hidden in this creative construction of the program from the specification are a set of obvious as well as non-obvious design decisions about how to encode certain parts of the specification in an efficient way using available implementation mechanisms to achieve performance criteria (the why of the design decisions)' (Baxter and Mehlich, 1997, p. 105).The artifact, in other words, contains within it traces of the design decisions made to bring it about. For reverse engineers, the designer's intentions, conceptions of the user, and skills can be traced in and through the artifact. To do this, reverse engineers take an existing technology and trace its genealogy backwards, looking at a whole host of artifacts and practices to uncover how the technology was developed: documentation (Lutsky, 1995), white papers, press releases, organizations (their structures, histories, and strategies) (Aiken et al., 1994), theories of the user, previous versions, databases (Hainaut et al., 1996), vanquished competitors, handwritten specifications (Leite and Cerqueira, 1995), older technologies and techniques, and of course lines of code. This is downright Nietzschean in its insistence upon actual documents and utterances and a search for historical a priori to the current object. The goal is to work backwards towards something approximating (but likely never exactly reflecting) the intentions of the designer, which requires a historical/genealogical sensibility.

When this happens, the artifact's heritage is traced. Because reverse engineering reverses the hegemonic engineering approach (begin with an abstract architecture; implement it in a concrete artifact), an intermediate result of reverse engineering is the uncovering of 'thought before thought,' of the abstractions that shape future actions. This is to say that reverse engineering reconstructs the abstract ideas the original designers *may* have held in the construction of the artifact. In a noopolitical sense, what this means is reverse engineering is a process of uncovering the thoughts that incite, induce, or constrain future actions (Foucault, 2003, p. 138) by way of what Foucault calls the 'body-object articulation,' the productive linkage between an artifact and a body (Foucault, 1979, p. 152). In Lazzarato's (2006) theory of noopower, 'thought before thought' is a process by which one mind (say, the mind of a designer) may influence the thoughts of another (say, the mind of an end user). Reverse engineering traces this backwards from the object, speculating as to the contours of power desired by the original designer. This is, in this admittedly grandiose sense, a genealogy of software artifacts as instruments capable of shaping our thoughts.

This is useful not only because it traces the contours of power, but also because reverse engineering provides an antidote to technological hype which consistently holds that new things are radical breaks with the past. In contrast to hype and the obsession with the new, reverse engineering understands technology to have a history that is contingent, a history steeped in power relations, discernible in part through genealogical inquiry. And because history and power are the products of social struggle, any technology is open to changing and reshaping.

The Legal Orientation

We live in what Lawrence Lessig (2004, p. xiv) has aptly called a 'permission culture.' That is, when we are confronted with many of the technologies and texts in our lives, we are hesitant to do more with them than simply consume them on the producer's terms (Gillespie, 2007). We fear lawsuits or criminal prosecution if we remix texts, quote ideas, copy and distribute items, or put a technology to uses other than those intended by their creators. In the case of software, the lengthy Terms of Service and End-User License agreements we click through are full of prohibitions, stipulations, and thinly-veiled legal threats. Thus, despite having a bewildering array of flexible, networked software systems at our command, we hesitate to probe, alter, or use software in ways that the original producers did not intend.

Here we can turn to the legal aspects of reverse engineering. Reverse engineering has a tradition of legal protections in contexts such as the United States and the European Union (Samuelson and Scotchmer, 2002). In traditional manufacturing, U.S. courts have held that the sale of an object is akin to its publication, and thus the legal owner of any object is free to take it apart, study it, and even produce a copy of it for sale. This situation is a bit more complex with software, because software is simultaneously ideational (in the sense that it is a materialization in code of the idea of its creator) and functional (Mackenzie, 2006). In terms of the ideational side, software is protected by copyright; I cannot copy lines of code from a copyrighted program into a new program. But in terms of the functional side, unless it has been protected by a patent, we are free to replicate the functionality of existing software. And in order to do that, we are in fact allowed to open up, probe, test, and disassemble software – even to make copies of it in the process.[2]

In this sense, the traditional legal protections afforded to reverse engineering is similar to the fair use exception in copyright law in many countries. The fair use exception allows people to quote, copy, remix, and reprint texts for the purposes of education, critique, and transformation. Despite their power and desire, the originators of intellectual property (IP) simply do not have absolute control over the uses of that IP. The fair use limitation sees to that. Likewise, reverse engineering is a way to prevent

[2] Two important cases that have established this in the United States context are *Sega Enterprises Ltd. v. Accolade, Inc*., 1992, and *Sony Computer Entertainment v. Connectix Corp*., 2000.

firms from monopolizing a technology (Samuelson, 2002; Samuelson and Scotchmer, 2002). Any unpatented technology can be taken apart and replicated by others; without this exception, a simple trade secret would be akin to a limitless monopoly. Courts have seen both fair use and reverse engineering as means to encourage new ideas and to limit the power of firms that originate ideas.

Moreover, like fair use, reverse engineering is an exception that only works when people use it. Established firms have consistently lobbied legislators to ban the use of reverse engineering (Samuelson and Scotchmer, 2002)0. The Digital Millennium Copyright Act is one such troubling example. The only way to maintain the reverse engineering exception is to keep using it (and, in some cases, keep fighting for it in the courts), just as fair use activists have done. To be sure, software firms such as Microsoft, Apple, Oracle, Google, and Facebook have become incredibly economically and politically powerful, which means they have high-octane lawyers who will use intellectual property laws to prevent critical access to their systems. One such technique is the use of Terms of Service which explicitly prohibit reverse engineering. However, the enforceability of such license terms is in question: courts tend to look askance at firms that use them to prohibit reverse engineering for the purposes of building interoperable system or competing systems. Additionally (and I find this very intriguing), there are explicit exceptions in the DMCA for reverse engineering of any digitally protected software that invades one's privacy.[3]

Finally, in a rhetorical sense, because reverse engineering has more acceptance in courts than terms such as 'hacking' or 'copying,' the term could provide rhetorical power to the critic. I realize 'hacking' is a fashionable term; I also know full well that 'hacking' does not mean malicious intrusion into networks to steal data. However, this negative connotation is unfortunately popular. By adopting

[3] See Samuelson and Scotchmer, (2002). Imagine if, especially after the popular revelations about government surveillance and the collusion of major social media firms with the state in the summer of 2013, one could use this exception in the DMCA to reverse engineer social media systems like Google or Facebook! They are, in many estimations, quintessential privacy-invading software systems.

'reverse engineering,' a technology critic can fend off some of the negative associations of terms like 'hacking' while drawing on the legal heritage I discuss above.

The Normative Orientation

Finally, reverse engineers do not simply study existing technologies; they do their work in order to build new systems. Thus, they have a normative position, albeit often a very immediate and rationalized one: usually they want to create a competing version of a product in order to sell it, create software that can interoperate with an existing platform or service, or gain the knowledge needed to maintain a system themselves rather than rely on the original producer for maintenance. As interested as they are in discerning the empirical properties of a technology (as I explored above in my discussion of the pragmatic orientation), reverse engineers also seek to change technological systems to meet their needs.

We can take this further and suggest a *critical* form of reverse engineering dedicated to political/economic and media justice. Whereas critical engagements of social media often stop at critique, the reverse engineering approach urges us to keep going. It calls for supporting and even becoming activists and technologists who are seeking to create software alternatives. Reverse engineering allows us to trace the path between alternatives and their less equitable predecessors, to see how the new alternatives take positive aspects of the old while avoiding the negative. Again, as I argue above in the discussion of the pragmatic orientation, reverse engineering isn't about idealized technologies, but rather taking the technology we have, decomposing it, discerning abstractions from it, and using this knowledge to make something better.

This maps onto Marx's observation in the *18th Brumaire* that 'Men [sic] make their own history, but they do not make it as they please; they do not make it under self-selected circumstances, but under circumstances existing already, given and transmitted from the past. The tradition of all dead generations weighs like a nightmare on the brains of the living' (Marx, 2008, p. 15). As Andrew Feenberg (1986) writes, a philosophy of praxis can be found in the overdetermination between the concrete circumstances of the present and the abstract, historically-developed conceptualizations used to both comprehend it and shape it. Rather than start with an ideal and bemoaning a world that doesn't live up to it, critical theory seeks out contradictions within the objects it encounters. These fissures provide us with tangible ways forward, new possibilities that are not simply abstract, timeless ethical ideals. Drawing on this, we can imagine a critical reverse engineering: start with the concrete technological system, derive abstractions from it, and use those abstractions to plot and create a better system.

Again, this is a bit grandiose in relation to the actual existing reverse engineering literature, but I propose that critical humanistic inquiry can learn a lot from reverse engineering. In the next section, I will explore this further with the case of TalkOpen, a short-lived Twitter alternative.

Reverse Engineering Twitter: The Case of TalkOpen

Building off of Lessig's point that we live in a culture of permission, I would argue that in the case of social media we live in a culture that denies us permission to push past the interface, to see how such social media systems are structured. Yet, as Ganaele Langlois (2013) rightly argues, social media software provides us with infrastructures to live our lives; their structures have a major influence on our online interactions. Therefore, despite the anxiety we might feel when we push past the interface, we must reverse engineer social media systems in order to see how we are producing ourselves within them.

One such effort along these lines was a site called TalkOpen. TalkOpen doesn't exist anymore, but it is like a black hole. You cannot see it, but you know it exists because of the gravitational pull of links, Tweets,[4] Facebook posts (OpESR, 2012), Pastebin chat dumps (“TalkOpen.info & Anon Communications,” 2012, “TRG - th3j35t3r & talkopen.info,” 2012), Imgur screenshots (@JackalAnon, n.d., n.d.), Internet Archive caches (“Talk Open - Welcome to The Talk Open Community,” 2012), and a brief mention in an information security blog (Schwartz, 2012) it left behind. These Web traces tell us that there was once something more to talkopen.info than what we see now: 'This website is for sale!'[5]

From roughly 31 March 2012 until early June of that year, TalkOpen.info was a microblogging site specifically built to be an alternative to Twitter. In that span of two months, TalkOpen had about 250 people sign up. Almost all of the members had usernames like 'Darknet,' 'Cpher,' 'DoxxyDox,' and 'AnonHooker,' and their posts were about topics like Anonymous, the Occupy movement, information security, and hacking. Although the user base was small, it was quite active, with new posts happening every hour.

Two months on the Internet? A user base of 250? New posts by the hour? Admittedly, these are not impressive numbers. And moreover, because the site did not have a large user base, there are only a few screenshots of it, and no accounts of its use beyond what I offer here. However, the case of TalkOpen provides a window into the methodology of reverse engineering I'm describing in this paper, and moreover I think there is value in preserving the memory of a failed technology in order to avoid reifying successful ones as the only or best possible ones. Moreover, by critically interrogating a project that reverse engineered Twitter, we also learn more about Twitter itself. Thus, I hope you will indulge me as I work with such 'small data.'

As TalkOpen founder XCpherX noted in an interview with me, 'Twitter alternative is the focus of the site.'[6] What does it mean to be a 'Twitter alternative'? I will explore this through the four

[4] For a collection of tweets about TalkOpen, contact the author.

[5] This is as of 22 April 2014. Previously, talkopen.info was comprised of a screen that said, 'This account has been suspended.'

[6] This interview was conducted via TalkOpen. Initially I intended to provide a link to this conversation, but now of course that link is lost. All quotes from XCpherX included in this essay come from that interview.

orientations I've described above.

Pragmatic

TalkOpen was a pragmatic response to the problems of social media, problems I will explore below. Rather than crafting a wholly new online communication system, it drew on something quite familiar: the conventions of Twitter. We would recognize the site as a 'micro-blogging' site, which contains many conventions:

- a character limit on posts,
- an interface that emphasizes the new (Gehl, 2011),
- A client-server architecture,
- individual-oriented account structures (i.e., the fields required for signup are biased towards an individual, not a group or other social entity),
- the follower-followed relationship,
- structured profile layouts (your profile image is 50x50 pixels; your username is limited to a certain number of characters, etc.),
- navigational layout with links such as Home, Members, Sign Out, and Settings,
- and the use of hashtags and @ signs in tagging and messaging.

These conventions could appear to be 'intuitive,' but if we act like John Law's (2002) 'naïve reader' as this reader appears in the book *Aircraft Stories*, we can start to see how this heterogeneous assemblage of objects and processes must be constantly organized in order to cohere. None of it is intuitive; for users to work with it, there needs to be a large amount of pedagogical training that happens via the interface. Thus, for any would-be social media alternative, there is a problem: do we create an assemblage of elements so new that users won't know how to use it? Or do we simply draw on the conventions forged by previous ones, even if those conventions represent the very problems we're seeking to overcome with a new system? TalkOpen opted for the latter. Thus, even though TalkOpen was a Twitter alternative, it did not abandon the old form. As XCpherX explained to me, 'im afraid that [the Twitter elements] may be taken away if we fiddle with it too much so to speak... just have to be careful what we do/remove/add.'

Moreover, TalkOpen was linked to Twitter from the outset, allowing TalkOpen users to post to both TalkOpen and Twitter simultaneously via RSS and the Twitter API. Thus, although TalkOpen was a 'Twitter alternative,' XCpherX made many pragmatic decisions to maintain a link to Twitter. This allowed users to move to TalkOpen without feeling as if they were abandoning Twitter altogether. Indeed, many of the posts TalkOpen users made appeared in both services.

Genealogical

TalkOpen was articulated into a larger historical moment – its potential place in the Occupy movement and in the network politics of the hacker group Anonymous. As such, TalkOpen might have been animated by the Twitter that was: the Twitter that was large enough to have a critical mass of users involved in Anonymous or Occupy, yet small enough to have not attracted the attentions of states. In other words, TalkOpen might have been redolent of Twitter before it was 'friended' by the state, to use an idea of Jack Bratich's (2011, p. 629). Moreover, this was a Twitter far removed from an Initial Public Offering (IPO) of stock. If there is any moment in the history of a social media site that reveals the site for what it truly is – a system to gather data on users and sell it for profit – the IPO has to be it.

In other words, TalkOpen might have been intended to be the pre-26 January 2012 Twitter, the Twitter of #jan25 and #egypt, the Twitter that in many popular accounts fomented revolutions: the pre-Wall Street orgy Twitter. In a famous blog post, Biz Stone (2011) wrote 'The Tweets Must Flow,' in

> which he stated, Some Tweets may facilitate positive change in a repressed country, some make us laugh, some make us think, some downright anger a vast majority of users. We don’t always agree with the things people choose to tweet, but we keep the information flowing irrespective of any view we may have about the content.

Wrapped in the language of free speech, Stone's post was seen as a promise to never censor expression in Twitter.

However, one year later on 26 January 2012, Twitter announced their intention to block certain Tweets and Twitter accounts by country (“Tweets still must flow,” 2012). It turned out that, in order to grow into international markets, the 'Tweets Must Flow' in certain contexts only. Alongside this announcement, there was a growing perception among users – right or wrong – that topics like Wikileaks, Occupy Wall Street, or Anonymous were being algorithmically excluded from Twitter's Trending Topics list (Johnstone, 2010a, 2010b). In other words, this was a concern that certain Tweets might not flow as well as others. Users protesting these changes used the hashtag #twittercensorship starting in late January 2012.

This was the moment in which TalkOpen reverse engineered Twitter to recreate the older Twitter, the Twitter of 'The Tweets Must Flow.' We can see this easily when we compare Twitter's Terms of Service[7] to TalkOpen's Terms of Service (TOS) statement, which in its entirety read

> By entering this site you take full responsibility for whatever you say or do and acknowledge that TalkOpen.info neither in whole or part is responsible for your actions. With that being said, TalkOpen will NOT sell your information to third parties or give up anyones [sic] information to law enforcement unless it is in regards to murder or cp [child pornography]. You have our word. Take it or leave it :)[8]

[7] Available at https://twitter.com/tos

[8] This was Terms of Service Statement as it appeared in May 2012. As far as I am aware, it was not altered during the

Here, TalkOpen's promise is that the users – not the site – would be 'responsible' for their statements. In that sense, although in very different words, TalkOpen repeated Biz Stone's 2011 promise to keep the statements flowing even if the site owners disagreed. Twitter users who defected to TalkOpen picked up on this and began to recruit users to TalkOpen in early 2012. They did so by articulating the Twitter hashtag #twittercensorship with others, including #OpMigrate and variations on #anonymous, to promote the site to any users concerned that post-26 January 2012 Twitter was losing its way as a site of free expression.

In this sense, TalkOpen reverse engineered Twitter by comparing its more recent version to older iterations, studying news reports and other para-documentation on the service, and using these insights to build a different, yet redolent, microblogging service.

Legal

TalkOpen was built on StatusNet, which is an open source alternative to Twitter with roots dating back to the late 2000s. StatusNet reverse engineered ideas such as 'microblogging' to create a Twitter alternative with the added functionality of federation across multiple servers via a social networking protocol (now called Ostatus). Using an open protocol and open source software would allow for a more distributed network architecture than the centralized Twitter system. As such, StatusNet is part of a long line of software projects that reverse engineer first-comers to expand interoperability and provide new platforms for users.[9]

In a sense, a major part of StatusNet's reverse engineering of Twitter is to graft the Free and Open Source Software (FOSS) model onto existing social media systems. This is true of other social media alternatives such as GNU Social and to a lesser extent Diaspora. Being built on top of StatusNet

lifetime of the site. The emoticon was always included.

[9] Important instances of reverse engineering of novel systems include Accolade's reverse engineering of the Sega Genesis and Connectix's reverse engineering of Sony's Playstation system (*Sega Enterprises Ltd. v. Accolade, Inc.*, 1992, *Sony Computer Entertainment v. Connectix Corp.*, 2000).

meant TalkOpen could rely on the long and successful history of legal engineering that is the GNU suite of licenses. As Gabriella Coleman (2009, p. 424) explains, such licenses use

> copyright law, a [United States] Constitutional mandate, to undermine the logic of copyright law. The [GNU General Public License] is built on copyright, but disables the restrictions of copyright to allow for modification, distribution, and access; it is also self-perpetuating because it requires others to adopt the same license if they modify copylefted software.

But this use of FOSS copyleft doesn't exhaust the legal story. In order for a software system to replicate the functionality of another, the history of protections of reverse engineering has to be in place. Part of FOSS production is what Coleman (2009, p. 425) calls a culture of 'legal exegesis' and 'legal training' that helps FOSS developers navigate the murky waters of intellectual property law. Such exegesis and training results in the legal jujutsu of copyleft, which is precisely written to use copyright against itself.

This has implications for reverse engineering. Although the emphasis on FOSS since the earliest days of Richard Stallman's work has been on free speech, a corollary emphasis has been on the right to open up and alter the software one uses: in other words, in the FOSS ethos, one has the right to reverse engineer software. Moreover, even if the Digital Millennium Copyright Act (DMCA) is often presented as merely a copyright law, its restrictions on reverse engineering coupled with the fact that FOSS activists universally despise it means that FOSS developers must engage with reverse engineering as a legal category of action to resist the DMCA.

Thus, being built on StatusNet meant that Twitter drew on these legal tactics to reverse engineer Twitter. It enjoyed the protection of being a part of a larger project within a legal context where reverse engineering is protected and with a long legal history of copyleft licensing. These legal protections are 'built in' when we start social media alternatives using copylefted platforms like StatusNet, Diaspora, or

Crabgrass.

However, TalkOpen partially failed to build on top of this legal history. While well-engineered legal documents like GNU licenses can protect a project, the far more blunt TalkOpen license (quoted in its entirety above) simply doesn't play the legal language game. Bad grammar, unclear abbreviations ('cp'), and a strange transition ('All that said') rendered TalkOpen's Terms of Service to be somewhat comical. Here, instead of reverse engineering social media terms of service in the same way that copyleft reverse engineers copyright, the TalkOpen TOS appeared to be a flat-out repudiation of legalese, likely – although I cannot be certain – setting TalkOpen up for dissociation if it were to come into contact with the world of lawsuits, National Security Letters, and prosecutions.

Normative

Finally, why build a Twitter alternative unless one has a goal in mind? The goal might be just to make another competitor in the business of inciting, digitizing, and valorizing user emotional labor (see, for example, Pinterest). Or it might be to build a new layer of abstraction on top of the 'platform' that a site like Twitter provides developers (see, for example, Topsy). But in the case of TalkOpen, it was more about #TwitterCensorship and Anonymous network politics. The hashtag #OpMigrate has a normative and pedagogical sense: make the shift to TalkOpen. There, you can do all the same things you would in Twitter (because TalkOpen is pragmatic and built on that model) but you get new freedoms.

In this sense TalkOpen was another in a long list of projects meant to be alternatives to mainstream social media: Diaspora, GNU Social, FreedomBox, Lorea, and Crabgrass, to name a few. As I argue elsewhere (Gehl, 2014, 2013a), the technologists and activists building these projects recognize many of the problems that social media critics have pointed out: ubiquitous surveillance (Andrejevic, 2007); the modulation of affect, emotion, and communication for the purposes of linking interaction to consumption (Elmer, 2004; Langlois, 2011); the centralization of the Web (Zittrain, 2008); the templated nature of the interfaces (Arola, 2010); the exploitation of user creativity for massive profits (Terranova, 2000); the lack of democratic control over the social media systems (Gehl, 2013b). To build an alternative that does not have these problems is indeed a worthy goal.

And yet, as I explore above in the 'Pragmatic' section, TalkOpen does not simply reinvent 'social media,' totally disregarding the design conventions of previous systems; this would be an impossible goal, and moreover it would ignore the progressive aspects of social media. TalkOpen was a recognition that much good can come out of even the most centralized social media system. Although I would never reduce Occupy Wall Street or the Arab Spring to 'Twitter Revolutions' or 'Facebook Revolutions,' research has shown that social media played an important part in organizing movements and shaping public perceptions about them (Morris, 2013). Reverse engineering's normative move is to take that good, maintain design conventions that people recognize and are comfortable with, and avoid the problems of centralized social media. This is what TalkOpen attempted to do, albeit unsuccessfully.

What went wrong with TalkOpen?

But of course, TalkOpen failed. What went wrong?

As I argue elsewhere (Gehl, 2014), one of the great animating ideas of the social media alternative builders is decentralization and distribution. These network architectures are based on the quasi-mythical founding principle of the Internet: redundancy in network paths. That is, rather than having all communications flow through a central hub, the goal is to have them flow through multiple paths and nodes. This reduces the likelihood that any one node becomes a weak spot in the network. Mainstream social media sites, such as Twitter, Facebook, and Google, are not distributed; they are highly centralized, using a client-server architecture. As we have seen in recent revelations about government institutions such as the National Security Agency (U.S.A), the Government

Communications Headquarters (U.K.), and the Communications Security Establishment (Canada), such centralized databases of user communications are tempting targets for state surveillance. And of course the sites themselves exploit their centrality in the day-to-day communications of billions to collect data on their users and sell them to the highest bidders. Thus, to build an alternative to mainstream social media, one commonly stated goal is to create distributed (or federated) systems, allowing users to host their own node of a social media system on their own computers and link these nodes across the Internet.

And yet, this is exceedingly difficult work. Major software engineering problems in the construction of building distributed social media sites include authenticating users, allowing users to find each other across installations, and encrypting communication to prevent 'man in the middle' attacks. Moreover, because mainstream social media is 'easy to use' (through many years of training users in their particular interfaces), alternatives would have to do all of the above *and* make the system as 'intuitive' as the mainstream sites. This is a tall order for any would-be social media reverse engineer.

Thus it is not a surprise that social media alternatives often end up replicating the client-server architecture. TalkOpen did just this, thus making the site vulnerable to determined attackers. Indeed, because TalkOpen was built for members of hacker groups Anonymous and LulzSec, it attracted the attention of the 'patriotic hacker' The Jester, who claimed in Internet Relay chat to have taken down TalkOpen.info on 31 May 2012 (“TRG - th3j35t3r & talkopen.info,” 2012). I cannot say for certain if this is true, but the timing is right. TalkOpen might have been too centralized to survive a determined attack by The Jester.

Setting aside the question of centralization, I also wonder if TalkOpen failed to become *enough* of a Twitter alternative. In other words, in its attempt to reverse engineer Twitter, TalkOpen might have kept too many Twitter conventions. Perhaps it failed to account for the highly individualized, personal branding and micro- or macro-celebrity features of Twitter. The rise of Twitter in the popular imagination was marked by the oddity of Ashton Kutcher being a highly followed celebrity. Because TalkOpen reverse engineered Twitter, and because Twitter's core organizing principle is the relationship between the individual follower and followed, perhaps this organizing primitive is not compatible with the politics of TalkOpen's users. To a degree – although certainly not completely – individual-to-individual social media grates against the networked politics of Anonymous and Occupy. We have of course seen 'celebrities' heralded by the anons: Julian Assange, Chelsea Manning, Aaron Schwartz, and now Eric Snowden. But the most compelling aspects of Anonymous and Occupy are the politics of networks and aggregations: *we* are the 99%. Expect *us*. The iconography of Anonymous includes anons wearing Guy Fawkes masks or suits with no heads. This is not a celebration of individuality but rather a statement about interchangeability and hiding in plain site.

Perhaps then TalkOpen's replication of the individualizing structure of Twitter was not the solution to the riddle of producing a social media alternative that more fully meets the organizing principles and politics of Anonymous. Perhaps, then, other social media alternatives – that is, other ways of reverse engineering mainstream social media – can learn from the culture and architecture of TalkOpen to produce something more viable.

Conclusion: Critical Reverse Engineering as a Bridge Between Engineering and Humanities

In sum, as I hope I've shown here, the methodology of reverse engineering could provide social media critics with a bridge between the critique of contemporary, proprietary social media and the production of better media systems. Reverse engineering does more than either celebrating Twitter, Facebook, or Google, or refusing to take part in them. It also does more than simply criticize social media systems. Instead, it provides a method to both disentangle all of the threads that go into any specific social media system *and* build criteria for the production of better systems. While the normative goals of traditional reverse engineering tend to be instrumental and usually tied to industrial ends, critical reverse engineering can involve normative goals that tie into the goals of democracy and media justice. Moreover, the methodology is only complete when the critic becomes a maker and attempts to build a new system that is related to the old while striving to ameliorate the problems of the old. All of this is achieved through a mix of pragmatic, genealogical, legal, and normative orientations that all appear in the reverse engineering literature. This chapter, of course, is not a description of a system I myself built; rather it describes TalkOpen, which I participated in but did not help construct. Indeed, I do not claim to have the technical ability to easily construct a social media alternative on my own. However, reverse engineering is not often a solitary activity. It involves multiple steps, from identifying an artifact to reverse engineer, decomposing it into parts, researching its history across different discursive domains (such as popular, business, legal, and academic), creating new abstractions from the old system, and then implementing those abstractions in a new system. As such, it is well-suited to collaboration. Critical reverse engineering thus has the potential to be a practice that brings together humanists and engineers. For example, critical scholars of social media who have done the work of dissociating Twitter, Facebook, Pinterest, etc., might consider partnering with technologists who are building nascent social media alternatives. This might mean helping with design, broad site goals including social and cultural elements, or specific elements of user-to-user or user-software interaction. While none of these practices is explicitly tied to coding, they all can be found in the best traditions of humanities scholarship. Reverse engineering, understood in this way, becomes a potential 'boundary object' (Barley et al., 2012; Turner, 2008) between engineers and humanists, allowing these fields to bridge discursive and conceptual gaps to collaborate.

Based on the case of TalkOpen elaborated here, were I to work with technologists creating a social media alternative, I would suggest several broad, tentative goals for successful critical reverse engineering of social media. Any new social media system would bear traces of the old, but would be built for social justice. It might be compatible with the old (in the traditions of software reverse engineering) and yet add layers of functionality. It would prevent the problems of contemporary social media (i.e., surveillance, reduction of interaction to consumer choices, and the valorization of free labor). It could include an interface that draws on existing design conventions but would also provide a pedagogy for deeper understanding of social media systems, data, privacy, and value production. The new system could expand the meaning of 'the social' by building on the older social metaphors (friend, follower, etc.). It could provide a level of abstraction above the network layer – thus hiding implementations of decentralization and encryption from end-users – while providing access to the inner workings of the social media system, allowing users to not only inspect the code but also alter it to improve the system.

TalkOpen failed to do these things, but its failure does not invalidate the potentials of reverse engineering social media.

References

Aiken, P., Muntz, A., Richards, R., 1994. DoD legacy systems: reverse engineering data requirements. Commun. ACM 37, 26–41.

Andrejevic, M., 2007. iSpy: Surveillance and Power in the Interactive Era. University Press of Kansas, Lawrence, KS.

Arola, K.L., 2010. The Design of Web 2.0: The Rise of the Template, The Fall of Design. Computers and Composition 27, 4–14.

Barley, W.C., Leonardi, P.M., Bailey, D.E., 2012. Engineering Objects for Collaboration: Strategies of Ambiguity and Clarity at Knowledge Boundaries. Human Communication Research 38, 280–308.

Baxter, I., Mehlich, M., 1997. Reverse engineering is reverse forward engineering, in: Baxter, I., Quilici, A., Verhoef, C. (Eds.), Proceedings of the Fourth Working Conference on Reverse Engineering, October 6-8, 1997, Amsterdam, the Netherlands. IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 104–113.

Bratich, J., 2011. User-Generated Discontent. Cultural Studies 25, 621–640. doi:10.1080/09502386.2011.600552

Coleman, G., 2009. CODE IS SPEECH: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers. Cultural Anthropology 24, 420–454. doi:10.1111/j.1548-1360.2009.01036.x

Elmer, G., 2004. Profiling machines: mapping the personal information economy. MIT Press, Cambridge, Mass.; London.

Feenberg, A., 1986. Lukács, Marx, and the Sources of Critical Theory. Oxford University Press, New York.

Foucault, M., 1979. Discipline and Punish. Vintage, New York.

Foucault, M., 2003. The subject and power, in: Rabinow, P., Rose, N.S. (Eds.), The Essential Foucault : Selections from Essential Works of Foucault, 1954-1984. New Press, New York, pp. 126–144.

Gehl, R.W., 2011. The archive and the processor: the internal logic of Web 2.0. New Media & Society 13, 1228–1244.

Gehl, R.W., 2013a. What’s on your mind? Social media monopolies and noopower. First Monday 18.

Gehl, R.W., 2013b. “Why I Left Facebook”: Stubbornly Refusing to Not Exist Even After Opting Out of Mark Zuckerberg’s Social Graph, in: Lovink, G., Rausch, M. (Eds.), Unlike Us Reader: Social Media Monopolies and Their Alternatives. Institute of Network Cultures, Amsterdam, pp. 220–238.

Gehl, R.W., 2014. Reverse Engineering Social Media: Software, Culture, and Political Economy in New Media Capitalism. Temple University Press, Philadelphia, PA.

Gillespie, T., 2007. Wired shut. MIT Press.

Hainaut, J.L., Englebert, V., Henrard, J., Hick, J.M., Roland, D., 1996. Database reverse engineering: From requirements to CARE tools. Reverse engineering 9–45.

Ingle, K.A., 1994. Reverse engineering. McGraw-Hill, New York.

@JackalAnon, n.d. mA5dL.png (PNG Image, 984 × 516 pixels).

@JackalAnon, n.d. 1Suu1.png (PNG Image, 963 × 621 pixels) - Scaled (94%).

Johnstone, A., 2010a. Why Isn’t #Wikileaks Trending on Twitter, and Should We Care? Student Activism.

Johnstone, A., 2010b. New Questions About Why #Wikileaks Isn’t Trending On Twitter. Student Activism.

Langlois, G., 2011. Meaning, Semiotechnologies and Participatory Media. Culture Machine 12, 1–27.

Langlois, G., 2013. Social media, or towards a political economy of psychic life, in: Lovink, G., Rausch, M. (Eds.), Unlike Us Reader: Social Media Monopolies and Their Alternatives. Institute of Network Cultures, Amsterdam, pp. 50–60.

Latour, B., 1987. Science in action : how to follow scientists and engineers through society. Harvard University Press, Cambridge, Mass.

Law, J., 1992. Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. Systems Practice 5, 379–393. doi:10.1007/BF01059830

Law, J., 2002. Aircraft stories: decentering the object in technoscience. Duke University Press, Durham, NC.

Lazzarato, M., 2006. The concepts of life and the living in the societies of control, in: Fuglsang, M., Sørensen, B.M. (Eds.), Deleuze and the Social. Edinburgh University Press, Edinburgh, Scotland, pp. 171–190.

Leite, J.C.S. do P., Cerqueira, P.M., 1995. Recovering business rules from structured analysis specifications, in: Wills, L., Newcomb, P., Chikofsky, E. (Eds.), Reverse Engineering: 2nd Working Conference. IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 8–12.

Lessig, L., 2004. Free culture : how big media uses technology and the law to lock down culture and control creativity. Penguin Press, New York.

Lutsky, P., 1995. Automated testing by reverse engineering of software documentation, in: Wills, L., Newcomb, P., Chikofsky, E. (Eds.), Reverse Engineering: 2nd Working Conference. IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 8–12.

Mackenzie, A., 2006. Cutting Code: Software and Sociality. Peter Lang, New York.

Marx, K., 2008. The 18th Brumaire of Louis Bonaparte. Wildside Press LLC.

Morris, L., 2013. Did the kill-switch really kill the internet?

OpESR, 2012. http://talkopen.info/doxers group for doxers [WWW Document]. Facebook. URL https://www.facebook.com/permalink.php?

id=132449120164059&story_fbid=204518512990354 (accessed 9.3.13).

Samuelson, P., 2002. Reverse engineering under siege. Communications of the ACM 45, 15–20.

Samuelson, P., Scotchmer, S., 2002. The Law and Economics of Reverse Engineering. The Yale Law Journal 111, 1575. doi:10.2307/797533

Schwartz, M.J., 2012. Anonymous Builds New Haven For Stolen Data [WWW Document]. InformationWeek. URL http://www.informationweek.com/news/security/vulnerabilities/232900590 (accessed 9.4.13).

Sega Enterprises Ltd. v. Accolade, Inc., 1992. , F. 2d.

Sony Computer Entertainment v. Connectix Corp., 2000. , F. 3d.

Star, S.L., 1999. The ethnography of infrastructure. American behavioral scientist 43, 377–391.

Stone, B., 2011. The Tweets Must Flow. Twitter Blogs.

TalkOpen.info & Anon Communications: A Simple Method for safER (not safe) Social [WWW Document], 2012. . Pastebin. URL http://pastebin.com/72aHMGb9 (accessed 9.3.13).

Talk Open - Welcome to The Talk Open Community [WWW Document], 2012. . Archive.org. URL http://web.archive.org/web/20120403045749/http://talkopen.info/

Terranova, T., 2000. Free Labor: Producing Culture for the Digital Economy. Social Text 18, 33–58.

TRG - th3j35t3r & talkopen.info [WWW Document], 2012. . Pastebin. URL http://pastebin.com/K9weNjUD (accessed 9.3.13).

Turner, F., 2008. From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism. University Of Chicago Press.

Tweets still must flow, 2012. . Twitter Blogs.

Zittrain, J., 2008. The future of the Internet and how to stop it. Yale University Press, New Haven [Conn.].