Le foucaldien
Open Access Journal for Research along Foucauldian Lines

Published: April 3, 2017
DOI: 10.16995/lefou.26

Robert W. Gehl · University of Utah · robert.gehl@utah.edu

(Critical) Reverse Engineering and Genealogy

This paper articulates "critical reverse engineering" with Foucauldian genealogy. It first explores the theoretical bases for reverse engineering, drawing mainly on software engineering literature. The author suggests that reverse engineers do more than simply take apart and document systems; they also trace technical artifacts back to preceding versions, theories of the user, and the organizations that created the artifacts. Next, the article justifies an interest in technologies by connecting reverse engineering with Science and Technology Studies. Finally, the paper considers the ways in which reverse engineering might inform genealogical inquiry, as well as the limitations of this approach.

KEYWORDS

actor-network, digital media, foucault, genealogy, infrastructure, reverse engineering, science and technology studies

HOW TO CITE

Robert W. Gehl: "(Critical) Reverse Engineering and Genealogy", in: *Le foucaldien*, 3/1 (2017), DOI: 10.16995/lefou.26

Contents

Open Library of Humanities

1. Introduction

In some of my previous work, including my book[1] and a couple of my articles[2], I've made the claim that a valuable approach to the study of technoculture is what I call "critical reverse engineering." As I define it, critical reverse engineering "is a method of producing knowledge by dissociating human-made artifacts. This knowledge is then used to produce new artifacts that simultaneously improve upon the old and yet also bear a relation to the old."[3] This approach, I argue, is a way to take apart the technical artifacts we confront on a daily basis. However, drawing on the example of reverse engineers, I suggest that critics cannot stop at critique of technology; I argue that we must take the knowledge made during our critiques to *produce something better*.

I've already published work exploring how makers of what I call "alternative social media"[4] are doing exactly this. Many people have criticized corporate social systems such as Facebook and Twitter for a range of reasons, including their emphasis on consumption and their use of surveillance and algorithms to shape online communications. As I argue, however, alternative social media makers do more than just critique: they make alternatives. They critically assess corporate social media, attempting to take what they view as positive about them and stave off the negative. This is, in a nutshell, critical reverse engineering.

In this line of work, I've suggested that there are four key aspects of critical reverse engineering:

- *Legal*: Reverse engineering, like "Fair Use" exceptions in copyright law, enjoys some legal protection.
- *Pragmatic*: Reverse engineering is about working with the technology we have at hand, rather than with an idealized, hoped-for sociotechnical system.
- *Normative*: Reverse engineers do their work in order to make something better (by whatever criteria they might have).
- *Genealogical*: In "forward" engineering, there is an implied order of steps to produce an object. These steps can be traced back — reversed — by starting with a concrete object.

However, despite writing all these pieces, I have not yet had the chance to fully articulate *reverse engineering* and *genealogy*. To be honest, I made a rather quick-and-dirty claim that reverse engineering might have something to offer to critical genealogy. I've based this claim on

[1] Robert W. Gehl: *Reverse Engineering Social Media: Software, Culture, and Political Economy in New Media Capitalism*, Philadelphia: Temple University Press 2014.

[2] Robert W. Gehl: Building a Better Twitter: A Study of the Twitter Alternatives GNU Social, Quitter, Rstat.us, and Twister, in: *The Fibreculture Journal*, 26 (22 December 2015), pp. 60–86; Robert W. Gehl: Critical Reverse Engineering: The Case of Twitter and TalkOpen, in: Ganaele Langlois, Joanna Redden, and Greg Elmer (eds.): *Compromised Data: From Social Media to Big Data*, New York: Bloomsbury 2015, pp. 147–169.

[3] Gehl: Critical Reverse Engineering: The Case of Twitter and TalkOpen, p. 148.

[4] Robert W. Gehl: The Case for Alternative Social Media, in: *Social Media + Society*, 1/2 (July 1, 2015), pp. 1–12.

the idea that genealogy is the work of doing a history of the present, attempting to trace conditions of possibility that made the present moment thinkable.[5] For Foucault, critical genealogy is "a historical investigation into the events that have led us to constitute ourselves and to recognize ourselves as subjects of what we are doing, thinking, saying [...] it will separate out, from the contingency that has made us what we are, the possibility of no longer being, doing, or thinking what we are, do, or think"[6]. In that sense, the contemporary subject — who we are — is the starting point, and the approach is to trace the historical processes, accidents, and events that have helped constitute who we currently are. Reverse engineering, likewise, starts with a concrete object and attempts to work backwards to the abstract concepts that helped shape it. Rather than starting with a subject, reverse engineering starts with a technical object and then engages in tracing how that object came to be.

Until recently, however, that's about as far as I had taken this connection. Fortunately, I was invited to the inaugural Critical Genealogies Workshop, held at the University of Denver in June 2016, and I used that opportunity to more deeply explore the connections between reverse engineering and genealogy. Thanks to feedback from the workshop participants, I have a stronger sense of how critical reverse engineering and Foucauldian genealogy may work together.

To explore this connection, this paper does the following:

- It explores the theoretical bases for reverse engineering, with a focus on software reverse engineering;
- justifies focusing on technology because of the overdetermination between technology and society;
- connects reverse engineering to critical genealogy;
- and concludes with some possible future directions for genealogists who might take up critical reverse engineering.

2. Theoretical Bases of Reverse Engineering

First of all, I should note that I mainly focus on reverse engineering of software, and much of what I cite in this paper comes from the field of software engineering. There is a long history of reverse engineering of "hardware;" in many liberal legal regimes, the sale of a piece of hardware is considered to be akin to publication, and thus if you legally acquire a hardware tool, you are allowed to make a replica of it. The situation with software is more complex, because as Samuelson and Scotchmer argue, there's "higher quantum of applied know-how" within digital prod-

[5] Michel Foucault: *Discipline and Punish: The Birth of the Prison*, trans. Alan Sherdian, 2nd ed., New York: Vintage 1995 [French 1975], p. 31.

[6] Michel Foucault: What Is Enlightenment? [French 1984], in: Paul Rabinow (ed.): *The Foucault Reader*, New York: Pantheon Books 1984, pp. 32–50.

ucts.[7] In other words, while a mechanical object certainly carries know-how, software's malleability and complexity means that more such knowledge is embedded in it. That said, some of the general principles hold across artifacts.

Basic Definition: Documenting a System

A basic definition of reverse engineering is provided by Chikofsky and Cross: "analyzing a subject system to identify the system's components and their interrelationships and create representations of the system in another form or at a higher level of abstraction."[8] As Kathryn Ingle puts it,

> *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.*[9]

To put this another way: a reverse engineer starts with a concrete, fully-implemented technical system, such as a machine or software installation, and through various means, attempts to describe that system in abstract terms. The concrete system is understood to be a particular implementation of those abstract conceptualizations. The abstract terms produced through reverse engineering tend to resemble the original abstract design plans that preceded the implementation of the concrete system. In other words, this is like analyzing a house in order to make a blueprint for it, aiming to make one's blueprint as close a replica of the original blueprint as possible.

In terms of our interest in genealogy, this general definition of reverse engineering seems somewhat limited; there's no mention of anything like conditions of possibility or problematization. Here, the goal is simply to *document* the object, especially in terms of what it does, representing the object in more abstract, diagrammatic terms. This is the somewhat boring process of writing a missing owner's manual or making a diagram. From this basic definition, reverse engineering does not seem to get at the wider social, political, economic, or technical conditions that made that object possible, a key object of inquiry for genealogy.

Looking Back in Time

However, a subtle move towards our interest in genealogy — that is, doing a history of the present by tracing conditions of possibility to see what made something possible — can be seen in reverse engineering's relationship to "forward engineering." Reverse engineering assumes a specific "forward engineering" process that involves creating abstractions and then implement-

[7] Pamela Samuelson and Suzanne Scotchmer: The Law and Economics of Reverse Engineering," in: *The Yale Law Journal*, 111/7 (2002), pp. 1575–1663, here p. 1579.

[8] E. J. Chikofsky and J. H. Cross: Reverse Engineering and Design Recovery: A Taxonomy, in: *Software, IEEE*, 7/1 (1990), pp. 13–17, here p. 15.

[9] Kathryn A. Ingle: *Reverse Engineering*, New York: McGraw-Hill 1994, p. 14.

ing them. "In spanning the life-cycle stages," Chikofsky and Cross note, "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."[10] "Design recovery," as discussed by Ted Biggerstaff, is intuiting the design process from "only the barest of clues"[11] contained within the technical object under observation. How can we move from "bare clues" to something that approximates the original design concepts? I suggest that this is possible because of the relationship between reverse and forward engineering. Engineering as a field of knowledge production has general practices that get repeated when engineers seek to create a thing. Knowing these general practices, a reverse engineer can work backwards from that thing to the design ideas of the original engineer(s).

Moreover, as that Chikofsky and Cross quote emphasizes, we also push *further back* past the design stage to the "requirements" that preceded the design of the system. In engineering, "requirements" refers to the needs of the intended user of the technology: what is the user trying to accomplish with this technology? Or more properly: who did the original designers imagine the users to be, and how did the designers imagine those users would use the technology? And thus, what must this technology do to satisfy these (imagined) needs and users? In fact, engineers suggest that requirements themselves be "engineered," a process called "requirements engineering," which involves gathering and systematizing potential uses before the design process begins.[12] Thus, if forward engineering ideally entails requirements gathering, design, and then implementation, reverse engineering starts with implementation and works backwards to those previous stages.

While the original design and requirements may not be the broad conditions of possibility that critical genealogists seek, they certainly get closer than mere documentation of the technical artifact. I will expand on this below in the "Genealogy" section.

Beyond Ideals, Beyond Code

So far, I've spoken of engineering (both forward and reverse) in idealized terms. However, there are two key challenges to this ideal conception found in the reverse engineering literature, and both of these challenges open up more possibilities for an articulation of genealogy and reverse engineering.

First, reverse engineering can expose the less-than-ideal flaws in a given artifact's production. For example, in their work on reverse engineering databases, Blaha and Premerlani note "idio-

[10] Chikofsky and Cross: Reverse Engineering and Design Recovery, p. 15.

[11] Ted J. Biggerstaff: Design Recovery for Maintenance and Reuse, in: *Computer*, 22/7 (1989), pp. 36–49, here p. 36.

[12] B. Boehm: A View of 20th and 21st Century Software Engineering, in: *Proceedings of the 28th International Conference on Software Engineering*, New York: ACM 2006, pp. 12–29.

syncrasies" in databases they reversed.[13] They argue this denies the idealized vision of a clear, logical design process by which something is made. In other words, reverse engineering reveals quirks, accidents, and idiosyncrasies of human/technical development. Whereas we might imagine a flawless and almost inevitable process of finding a user's needs, designing something to meet those needs, and precisely implementing that design, what Blaha and Premerlani find is that the implementers (coders, builders, etc.) take shortcuts, make myriad (sometimes bad) decisions, and make mistakes. This doesn't necessarily prevent the implemented system from working, but it reflects the micro-interactions between humans and material as something is built. If we dig beneath the surface of a working system, we will uncover specific, messy human/technical articulations.

Secondly, and more importantly, in their quest to work from a concrete implementation to abstract design and requirement concepts, reverse engineers tend to do more than just look at technical artifacts such as code. The reverse engineering literature is full of these methodological expansions:

- Canfora Harman and Di Penta discuss looking at embedded comments and variable names within software code.[14] Comments are human-readable notes within software that are not run by the computer and are thus not part of the code. They are often notes to other designers, explanations of functions, or justifications of specific implementation choices. Using a software engineering concept of "traceability," these code-level elements can be linked to abstract designs (i.e., a comment could refer to how the code implements a part of the overall design).
- Both Leite and Cerqueira[15] and Aiken et al[16] move past "code" to consider the structures of the organizations that gave rise to the code in question. Aiken et al achieve this by reverse engineering "structured specifications" — documents that specify what the software does — in order to recover more abstract organizational rules and practices. In other words, they argue we can learn how an organization works by studying one of the artifacts it produces.
- Similarly, Patricia Lutsky looks at software documentation to recover requirements to aid in making systems to test software reliability.[17]

[13] Michael R. Blaha and William J. Premerlani: Observed Idiosyncracies of Relational Database Designs, in: Linda Wills, Philip Newcomb, and Elliot Chikofsky (eds.): *Reverse engineering: 2nd Working conference*, Los Alamitos: IEEE Computer Society Press 1995, pp. 116–125.

[14] Gerardo Canfora Harman and Massimiliano Di Penta: New Frontiers of Reverse Engineering, in: *2007 Future of Software Engineering*, IEEE Computer Society 2007, pp. 326–341, URL: http://dl.acm.org/citation.cfm?id=1254728.

[15] Julio Cesar Sampaio do Prado Leite and Paulo Monteiro Cerqueira: Recovering Business Rules from Structured Analysis Specifications, in: Linda Wills, Philip Newcomb, and Elliot Chikofsky (eds.): *Reverse Engineering: 2nd Working Conference*, Los Alamitos: IEEE Computer Society Press 1995, pp. 8–12.

[16] Peter Aiken, Alice Muntz, and Russ Richards: DoD Legacy Systems: Reverse Engineering Data Requirements, in: *Communications of the ACM*, 37/5 (1994), pp. 26–41.

[17] Patricia Lutsky: Automated Testing by Reverse Engineering of Software Documentation, in: Linda Wills, Philip Newcomb, and Elliot Chikofsky (eds.): *Reverse Engineering: 2nd Working Conference*, Los Alamitos: IEEE Computer Society Press 1995, pp. 8–12.

Most intriguingly, both Ahmed Hassan[18] and Luigi Cerulo[19] look at software repositories such as Subversion and CVS. Software repositories are systems that allow multiple coders to make changes to a software system; they track versions of the software, allowing engineers to revert to an older version if new changes do not work. Hassan and Cerulo take advantage of these systems in order to to study software evolution and developers' approaches over time. As Hassan argues,

> *Such repositories contain a wealth of valuable information for empirical studies in software engineering: source control systems store changes to the source code as development progresses, defect tracking systems follow the resolution of software defects, and archived communications between project personnel record rationale for decisions throughout the life of a project. Such data is available for most software projects and represents a detailed and rich record of the historical development of a software system.*[20]

Notably, Hassan points out "archived communications" as key data for insight into software development and design decisions. Hassan, who worked in software development for firms such as IBM, notes the value of software repositories as historical archives of design decisions:

> *In the role of a developer, we faced the daunting task of understanding large complex software systems which were developed by others, enhanced by many and patched frequently to meet tight deadlines or critical emergencies – we found ourselves along with other developers falling back to the initial version of a complex piece of code to understand it. In many cases, the initial cut of a piece of code, which is stored in the source control system, was easier to understand and was cleaner than the current code. In addition, we often investigated prior changes to code segments to gain a better understanding of the rationale for their current complexity or to clarify the design choices.*[21]

Thus, the literature on reverse engineering is far more than simply documenting a technology. It's more than making the missing owner's manual: it is a forensic process (akin to that described by Matt Kirschenbaum[22]) that uncovers a wealth of empirical information, including mistakes, off-color jokes in comments,[23] organizational rules and structures, internal communications, and the evolution of software over time.

[18] Ahmed Hassan: *Mining Software Repositories to Assist Developers and Support Managers* (2004), URL: https://uwspace.uwaterloo.ca/handle/10012/1017; Ahmed E. Hassan: The Road Ahead for Mining Software Repositories, in: *Frontiers of Software Maintenance*, 2008. FoSM 2008 (IEEE, 2008), pp. 48–57, URL: http://ieeexplore.ieee.org/document/4659248/.

[19] Luigi Cerulo: On the Use of Process Trails to Understand Software Development, in: *Reverse Engineering*, 2006. WCRE '06. 13th Working Conference on Reverse Engineering (IEEE, 2006), pp. 303–304, URL: http://ieeexplore.ieee.org/document/4024004/.

[20] Hassan: Mining Software Repositories to Assist Developers and Support Managers, p. 2.

[21] Ibid., p. 4.

[22] Matthew G. Kirschenbaum: *Mechanisms: New Media and the Forensic Imagination*, Cambridge: MIT Press 2012.

[23] For a collection of comment humor, see Roy Lachica: Fuzzzy Blog: 40 Most Funny Code Comments Ever, in: *Fuzzzy Blog* (September 14, 2014), URL: http://fuzzzyblog.blogspot.de/2014/09/40-most-funny-code-comments.html.

In sum, it's clear there's a great deal of meticulous, rigorous work in reverse engineering. It's clear that a reverse engineer will look at whatever she can get her hands on in order to understand how the technical artifact reflects more abstract design goals, requirements, and even organizational practices. And especially with the more expansive works, we see a move past mere documentation of how technical artifacts work, and even past the recovery of design or requirements; we start to see ways in which reverse engineers begin to reconstruct the sequence of technical development, including design decisions, internal debates over development, and the resolution of — not to mention the *defining* of — faults in the design.

I would suggest these moves do get us closer to tracing the conditions that give rise to specific technical artifacts. The connections between the artifact and organizational practices (communication and hierarchies) is especially intriguing. I will explore this further in the "Genealogy" section of this essay. But first, I want to draw on Science and Technology Studies to further stress the importance of studying technical artifacts, especially for those of us interested in Foucauldian genealogy.

3. Articulation between Technology and Society

Why is it important to consider reverse engineering as a source of genealogical methods and ideas? The relationship between technology and the social is key: technology is, as Jonathan Sterne pithily puts it, society crystallized.[24] To justify my interest in reverse engineering as a possible resource for genealogical criticism, I turn to the field of Science and Technology Studies (STS). Here, I offer three key articulations between the reverse engineering literature and STS.

Punctualization and Depunctualization

There are two key concepts in actor-network theory that bear directly on reverse engineering: punctualization and depunctualization.[25] Michel Callon argues that "the process of punctualization […] converts an entire network into a single point or node in another network."[26] In other words, a network (in ANT, understood to be social, technical, natural processes) can be "simplified" into a punctualization — an obdurate, coherent object that stands in for the network that made it possible. A punctualization is thus a point of contact between networks. We can easily think of specific technical artifacts (an installation of an operating system, or a particular smart phone) in this way. For Bruno Latour, "punctualization" is the pinnacle of the process of association; it is the construction of the apocryphal "black box" of technology.[27]

[24] Jonathan Sterne: Bourdieu, Technique and Technology, in: *Cultural Studies*, 17/3–4 (2003), pp. 367–389.

[25] Robert W. Gehl: The Politics of Punctualization and Depunctualization in the Digital Advertising Alliance, in: *The Communication Review*, 19/1 (2016), pp. 35–54.

[26] Michel Callon: Techno-Economic Networks and Irreversibility, in: John Law (ed.): *A Sociology of Monsters: Essays on Power, Technology and Domination*, New York: Routledge 1991, pp. 132–165, here p. 153.

[27] Bruno Latour: *Pandora's Hope: Essays on the Reality of Science Studies*, Cambridge: Harvard University Press 1999, p. 184.

It should be clear that the basic impulse of reverse engineering is to *depunctualize* an artifact: to reverse the process of punctualization, to take the black box apart and see how it all hangs together. Here, documenting the object is a means of inscribing the knowledge gleaned from this depunctualizing process. However, the more complex impulses of reverse engineering — that is, the moves beyond mere documentation — including *tracing the associations of the artifact* with its network: organizations, theories of the user, previous versions, and communication networks. This is depunctualization with a goal of doing a sort of sociology of objects and networks, an "ethnography of infrastructure" as Susan Leigh Star[28] puts it: what other networks made this object possible? What previous efforts had to take place to make this object seem so simple, durable, and coherent?

Technology and Time

The initial definition of reverse engineering — the basic documentation of a system — ties in with Bernard Stiegler's suggestion that the technologies we confront have a major influence on "protentions," or our capacity to imagine or expect particular futures. Stiegler suggests technical artifacts are "exteriorized" thoughts or memories (or, to use his better term, "retentions"). He refers to technical retention as "tertiary retention":

> *Tertiary retention is a mnemotechnical exteriorization of secondary retentions [memory of time past] which are themselves engendered by primary retentions [experience of time passing] […]. All technical objects constitute an intergenerational support of memory which, as material culture, overdetermines learning [apprentissages] and mnesic activities.*[29]

In other words, as carriers of specific memories/retentions/ways of knowing and being, the ensembles of technology we confront have an influence over how we perceive the world, the past, ourselves, and our possible futures. James Ash sums this up quite well:

> *Stiegler suggests that the 'now' of perception is not simply 'there', or apparent to human perception, but only 'appears' to humans through the sets of equipment and technology which make up the ecology of an environment. Clocks, timetables, hammers, and so on implicitly create an experience of the present and a way of relating past memory to future experience.*[30]

Reverse engineering's basic definition of documenting a technical artifact reflects this insight. We can be content to simply use a piece of software or technology; or, we can *take it apart and document it*, trying to understand it better, knowing (perhaps intuitively) that the technology has a major influence on how we conduct ourselves and our practices.

[28] Susan Leigh Star: The Ethnography of Infrastructure, in: *American Behavioral Scientist*, 43/3 (1999), pp. 377–391.

[29] Bernard Stiegler: *For a new critique of political economy*, Cambridge: Polity 2010, p. 9.

[30] James Ash: Technology, Technicity, and Emerging Practices of Temporal Sensitivity in Videogames, in: *Environment and Planning A*, 44/1 (2012), pp. 187–203, here p. 191.

Pre-scribing for the "Configured User"

Similarly, if we consider how reverse engineers attempt to work their way back to "requirements" (the imagined uses to which the technology will be put), we see a connection between reverse engineering and the concepts of the "configured user" or pre-scriptions. Steve Woolgar's conceptualization of a technology as "text" is useful here: although technologies are subject to interpretation (i.e., different uses) by end users, they are not radically open-ended. Like any text, their authors imagine particular readers and address them *qua* the artifact.[31]

Another way of thinking about this is in terms of "scripts," a concept developed first by Madeleine Akrich.[32] Friz and Gehl describe this perspective:

> *Akrich considers technological artifacts as being "pre-inscribed" or carrying "pre-scriptions" placed there by their designers. "Designers thus define actors with specific tastes, competences, motives, aspirations, political prejudices, and the rest, and they assume that morality, technology, science, and economy will evolve in particular ways."*[33] *A vision of the user's relationship to and interactions with the object and its consequent actors is inscribed such into the object itself. Presenting a technological artifact as something that bears a "script" — metaphorical "stage directions for the performance of using the technology — draws attention to artifacts as actants, things that can make a difference in a situation.*[34]

In both cases — technology as text with imagined readers, or technology as script with prescribed uses — designers of technology "configure" users[35] via the medium of a technology: they imagine uses and users and build their technologies accordingly. Although the technology can be appropriated, altered, modified, or used in previously unimaginable ways, certain uses are in fact easier than others simply by dint of the design. Reverse engineering, then, is a means to start with an artifact and consider it as a "text" inscribed with particular visions of users and uses.

4. Genealogy

So STS suggests that the confrontation between human and engineered environments is a key moment to study. I think it is clear that reverse engineering may be a useful technique to do that work. With that justification in mind, I want to formally connect reverse engineering with gene-

[31] Steve Woolgar: Configuring the User: The Case of Usability Trials, in: John Law (ed.): *A Sociology of Monsters: Essays on Power, Technology and Domination*, New York: Routledge 1991, pp. 57–99.

[32] Madeleine Akrich: The de-Scription of Technical Objects, in: Wiebe Bijker and John Law (eds.): *Shaping Technology/Building Society*, Cambridge: MIT Press 1992, pp. 205–224.

[33] Ibid., p. 208.

[34] Amanda Friz and Robert W. Gehl: Pinning the Feminine User: Gender Scripts in Pinterest's Sign-up Interface, in: *Media, Culture & Society*, 38/5 (2016), pp. 686–703.

[35] Woolgar: Configuring the User: The Case of Usability Trials.

alogy[36]. Genealogy is the rigorous search for actual, historical practices that provide conditions of possibility for an object, practice, or way of being in question. For Foucault, critique should be "genealogical in the sense that it will not deduce from the form of what we are what is is impossible for us to do and to know; but it will separate out, from the contingency that has made us what we are, the possibility of no longer being, doing, or thinking what we are, do, or think."[37] In other words, Foucault is interested in historically tracing how our ways of thinking and being came to be, the historically-developed limits of our thinking and being, and how our ways of thinking and being might be different. For this, he turned away from the transcendental and towards the concrete and specific.

Those concrete and specific practices are the fields from which new practices, objects, and ways of being emerge. This emergence is not a simple, inevitable process, but one that is messy. As Colin Koopman puts it,

> *genealogy tracks complex histories of alliance, support, and reinforcement that facilitate the production of spaces of practical possibility. The point is not to discern how the intentions of those in the past effectively gave rise to the present, but rather to understand how various independently existing vectors of practice managed to contingently intersect in the past so as to give rise to the present.*[38]

To trace these alliances, a first step is to *problematize*, or to do two things. First is to answer a question: what are the already-existing problems in contemporary life? This involves naming, clarifying, and describing problems and tracing their emergence over time. It also involves finding agents who have defined phenomena as problems. Such problems are thus not neutral; power/knowledge relations come into play. Here we can see resonance with Michel Callon's use of "problematization," where particular actors attempt to establish themselves as "obligatory points of passage" in networks.[39] For Callon, successful actor-networks are constructed in part by actors who define, enroll, and contain elements, including definitions of what is problematic in contemporary life. In doing so, these actors are able to establish themselves as authorities or gatekeepers for anyone wanting to address the problem in question. The linkage between this concept and Foucault's articulation of power/knowledge should be obvious: the production of knowledge is accompanied by the production of power, because whomsoever is able to produce knowledge of problems (and no doubt proffer solutions in the form of specific knowledges and practices) becomes dominant.

[36] Michel Foucault: Nietzsche, Genealogy, History [French 1971], trans. D. Bouchard and S. Simon, in: Paul Rabinow (ed.): *The Foucault Reader*, vol. 1, New York: Pantheon Books 1984, pp. 76–100.

[37] Foucault: "What Is Enlightenment?", p. 45.

[38] Colin Koopman: *Genealogy as Critique: Foucault and the Problems of Modernity*, Bloomington: Indiana University Press 2013, p. 107.

[39] Michel Callon: Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St Brieuc Bay, in: John Law (ed.): *Power, Action, and Belief: A New Sociology of Knowledge?*, London/Boston: Routledge & Kegan Paul 1986, pp. 196–233, here p. 204.

Indeed, it is this articulation between ways of knowing and instantiations of power that must be problematized. This brings us to the second part of problematization: the act of alerting us to danger. The genealogist should show how contemporary problems are *dangerous* (and therefore are in need of critical attention). It goes without saying that Foucault and Nietzsche saw problems such as discipline or Christian morality as dangerous because they could either have deleterious effects or because many of the responses to them created new problems. Here, then, critique is not making prescriptions, but is instead concerned with taking danger seriously and convincing all of us to do so, too.

With problematization (in its nominal and verbal forms) established, the genealogist turns to the various practices that have, over time, given rise to contemporary problems. "This means that the genealogist will seek to narrate and explain historical processes by reference to the problems that motivate certain processes and the specific practices that develop in response to these problems."[40] As Foucault puts it, the genealogist "must be able to recognize the events of history, its jolts, its surprises, its unsteady victories and unpalatable defeats — the basis of all beginnings, atavisms, and heredities."[41] In other words, how have past practices contributed, even if accidentally, to a contemporary problem? And, when something is constructed as a problem, what practices are introduced, adopted, or modified to deal with this problem? What individuals and institutions position themselves as the obligatory points of passage (in Callon's sense), as the only ones qualified to mitigate, solve, or at least sustain the problem? How did they successfully arrive at that position? When others confront such problems or points of passage, what do they do in the face of them? What objects, technologies, modes of inquiry, or instruments became entangled within the problem? Following Chandra Mukerji's reading of Nietzsche, how do "utility, habit, forgetfulness, and error" figure into these practices?[42]

Thus, Foucauldian genealogy, a genealogy of problematization, helps us see the ingredients, elements, practices, accidents, and threads that come together over time to assemble in the form of the complex problems we face today. More importantly, it helps trace the development of conditions of possibility that simultaneously allow us to think of and act out solutions as well as constrain our actions and thoughts. In other words, this is about the age-old structure/agency tension. As Mukerji puts it, such a method

> *assumes that cultural forms have deep roots, and that we ourselves are by necessity carriers of those traditions. We enter a pre-existing world that seems natural to us, and whose culture we pick up intuitively. We mimic what we see in others, learn to act as members of our communities, and take on predispositions that we do not recognize in ourselves.*[43]

[40] Koopman: *Genealogy as Critique*, p. 97.

[41] Michel Foucault: Nietzsche, Genealogy, History, in: *Semiotexte*, 3/1 (1978), pp. 78–94, here p. 80.

[42] Chandra Mukerji: Cultural Genealogy: Method for a Historical Sociology of Culture or Cultural Sociology of History, in: *Cultural Sociology*, 1/1 (2007), pp. 49–71, here p. 53.

[43] Ibid., p. 52.

I hope that the articulation between critical genealogy and reverse engineering is clear. If technical artifacts a) stand in for larger networks, b) have an influence on how we perceive the present and possible futures, c) are built as responses to requirements — another way of saying "problems", and d) prescribe particular uses and actions, and if reverse engineering is capable of tracing these practices by starting with technical objects, then reverse engineering ought to be useful for a larger genealogical project. It's certainly not the only approach, and I don't think it would be a means for a complete genealogy of a technical practice or object, but I would suggest that we can imagine critical reverse engineering as a key approach for any genealogy that includes examination of particular technologies.

5. Conclusion: Possibilities, Limitations

Reverse engineering is the critical examination of the technical artifacts we confront. This is often, but not always, in the absence of access to: the designers; the organization that produced them; technical documentation; earlier versions and blueprints. As such, I would suggest that it offers a great deal of insight for critical genealogists who attempt to trace histories of the present as that present is made visible in the form of technologies. However, there are limits in the reverse engineering approach. I ought to address them. Specifically:

- Reverse engineers acknowledge that they cannot recreate the exact decisions and documents of the original designers.
- Reverse engineering's scope is often small, focusing on specific technologies.
- Finally, reverse engineers desire a way to automate this process, removing human judgment and intervention as much as possible. In other words, they seek to eradicate subjectivity (as they define that term) in order to build technologies that can "objectively" reverse engineer other technologies.

I take the first as a caution for genealogy: if reverse engineers cannot reverse a standardized process — if they cannot move from a concrete artifact to the original abstract design — how could genealogists? I suggest that genealogists might follow in the footsteps of reverse engineers by also avoiding the claim that they are constructing an externally verifiable history of the present and instead consider how their histories intervene as much as they reveal. I believe that most genealogists, if pressed, would suggest this is the case with their work.

The second limitation requires us to do something that reverse engineers do not claim to do, but that genealogists are particularly well-suited for: articulate technologies into larger historical moments. Reverse engineering cannot address all of the questions a genealogist might pose, but I still believe it can be very useful to a genealogist.

Moreover, the smaller focus of reverse engineering may also be an asset to genealogy, especially as genealogy relates to possible solutions to social problems. Returning to my initial definition of critical reverse engineering, I must note that critical reverse engineering is more than the dissociation of technical artifacts; it is more than tracing the histories of alliances and accidents that gave rise to them. Critical reverse engineering also involves the *production* of new artifacts from the knowledge acquired through reverse engineering the previous artifact. As I argue elsewhere,

> *Whereas critical engagements [. . .] 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 [. . .] 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.*[44]

I realize this grates against a key caution inherent in the genealogical method: that our current practices and technologies have emerged from incredibly complex historical processes, and that any changes we might make to them are dangerous because they may not be adequate to address the original problems and they will inevitably lead to unexpected outcomes. In other words, Foucauldian genealogy is often about diagnosis, not prescriptions. Thus, the idea that we can simply take apart a technology, learn how it works, and make something "better" appears naïve. However, I would suggest that a more humble way forward is in terms of experimentation: if we are dissatisfied with our current sociotechnical systems — say, we are angry about surveillance, dissatisfied with the algorithmic shaping of our daily lives, or disappointed in how our sociality is limited by technical mediators — then critical reverse engineering could provide clues about new ways to experiment and construct alternative technical systems, which in turn could be articulated with other experiments in different domains (political, economic, cultural, etc.).

The final limitation is actually quite intriguing. What if genealogists take inspiration from engineers' desires to automate and systematize reverse engineering? What would a machine-assisted genealogy look like? Indeed, *Le foucaldien*'s special issue on *Distant Reading and Discourse Analysis* offers a key point of articulation between Digital Humanities and genealogical work. Further articulations might be made possible by adapting techniques of automated reverse engineering for humanistic inquiry.

[44] Gehl: Critical Reverse Engineering: The Case of Twitter and TalkOpen, p. 155.