



Nick Seaver

Follow

automatic anthropologist, sound/media/technology studier, resonator

Jan 27, 2014 · 15 min read

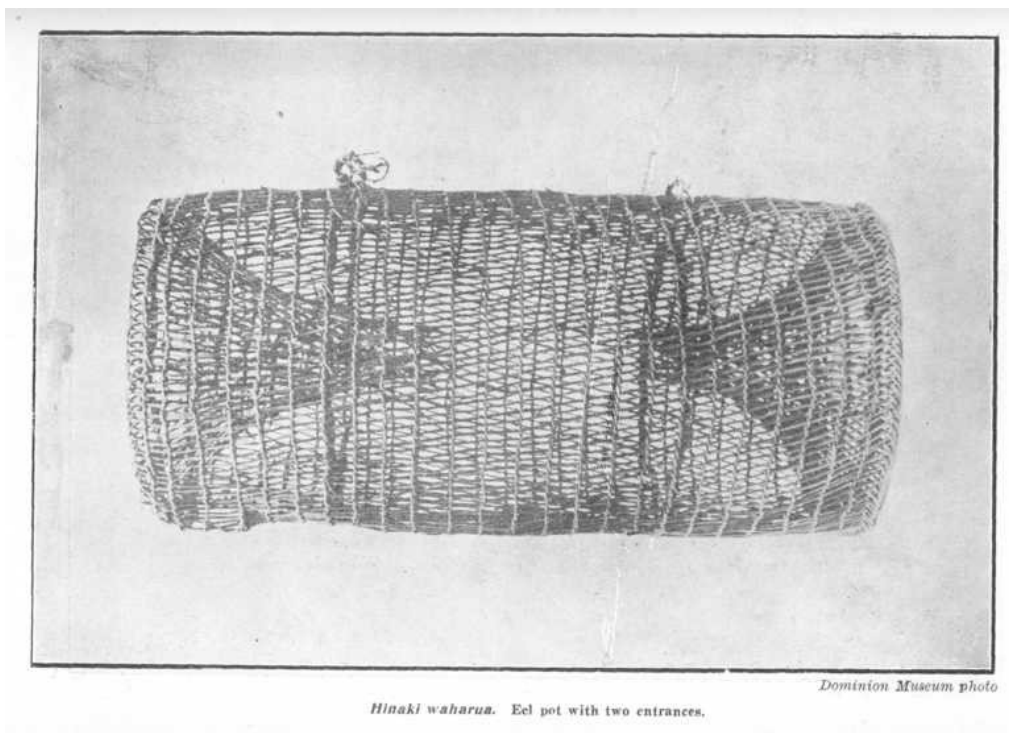
On Reverse Engineering

Looking for the cultural work of engineers

The Atlantic welcomed 2014 with a major feature on web behemoth Netflix. If you didn't know, Netflix has developed a system for tagging movies and for assembling those tags into phrases that look like hyper-specific genre names: Visually-striking Foreign Nostalgic Dramas, Critically-acclaimed Emotional Underdog Movies, Romantic Chinese Crime Movies, and so on. The sometimes absurd specificity of these names (or "altgenres," as Netflix calls them) is one of the peculiar pleasures of the contemporary web, recalling the early days of website directories and Usenet newsgroups, when it seemed like the internet would be a grand hotel, providing a room for any conceivable niche.

Netflix's weird genres piqued the interest of *Atlantic* editor Alexis Madrigal, who set about scraping the whole list. Working from the US in late 2013, his scraper bot turned up a startling 76,897 genre names—clearly the emanations of some unseen algorithmic force. How were they produced? What was their generative logic? What made them so *good*—plausible, specific, with some inexpressible touch of the human? Pursuing these mysteries brought Madrigal to the world of corpus analysis software and eventually to Netflix's Silicon Valley offices.

The resulting article is an exemplary piece of contemporary web journalism—a collaboratively produced, tech-savvy 5,000-word "long read" that is both an exposé of one of the largest internet companies (by volume) and a reflection on what it is like to be human with machines. It is supported by a very entertaining altgenre-generating widget, built by professor and software carpenter Ian Bogost and illustrated by Twitter mystery darth. Madrigal pieces the story together with his signature curiosity and enthusiasm, and the result feels so *now* that future corpus analysts will be able to use it as a model to identify texts written in the United States from 2013–14. You really should read it.



A Māori eel trap. The design and construction of traps (or filters) like this are classic topics of interest for anthropologists of technology. cc-by-sa-3.0

As a cultural anthropologist in the middle of a long-term research project on algorithmic filtering systems, I am very interested in how people think about companies like Netflix, which take engineering practices and apply them to cultural materials. In the popular imagination, these do not go well together: engineering is about universalizable things like effectiveness, rationality, and algorithms, while culture is about subjective and particular things, like taste, creativity, and artistic expression. Technology and culture, we suppose, make an uneasy mix. When Felix Salmon, in his response to Madrigal's feature, complains about "the systematization of the ineffable," he is drawing on this common sense: engineers who try to wrangle with culture inevitably botch it up.

Yet, in spite of their reputations, we always seem to find technology and culture intertwined. The culturally-oriented engineering of companies like Netflix is a quite explicit case, but there are many others. Movies, for example, are a cultural form dependent on a complicated system of technical devices—cameras, editing equipment, distribution systems, and so on. Technologies that seem strictly practical—like the Māori eel trap pictured above—are influenced by ideas about effectiveness, desired outcomes, and interpretations of the natural world, all of which vary cross-culturally. We may talk about technology and culture as though they were independent domains, but in

practice, they never stay where they belong. Technology's straightforwardness and culture's contingency bleed into each other.

This can make it hard to talk about what happens when engineers take on cultural objects. We might suppose that it is a kind of invasion: The rationalizers and quantifiers are over the ridge! They're coming for our sensitive expressions of the human condition! But if technology and culture are already mixed up with each other, then this doesn't make much sense. Aren't the rationalizers expressing their own cultural ideas? Aren't our sensitive expressions dependent on our tools? In the present moment, as companies like Netflix proliferate, stories trying to make sense of the relationship between culture and technology also proliferate. In my own research, I examine these stories, as told by people from a variety of positions relative to the technology in question. There are many such stories, and they can have far-reaching consequences for how technical systems are designed, built, evaluated, and understood.

. . .

The story Madrigal tells in *The Atlantic* is framed in terms of "reverse engineering." The engineers of Netflix have not invaded cultural turf—they've reverse engineered it and figured out how it works. To report on this reverse engineering, Madrigal has done some of his own, trying to figure out the organizing principles behind the altgenre system. So, we have two uses of reverse engineering here: first, it is a way to describe what engineers do to cultural stuff; second, it is a way to figure out what engineers do.

So what does "reverse engineering" mean? What kind of things can be reverse engineered? What assumptions does reverse engineering make about its objects? Like any frame, reverse engineering constrains as well as enables the presentation of certain stories. I want to suggest here that, while reverse engineering might be a useful strategy for figuring out how an existing technology works, it is less useful for telling us how it came to work that way. Because reverse engineering starts from a finished technical object, it misses the accidents that happened along the way—the abandoned paths, the unusual stories behind features that made it to release, moments of interpretation, arbitrary choice, and failure. Decisions that seemed rather uncertain and subjective as they were being made come to appear necessary in retrospect. Engineering looks a lot different in reverse.

This is especially evident in the case of explicitly cultural technologies. Where “technology” brings to mind optimization, functionality, and necessity, “culture” seems to represent the opposite: variety, interpretation, and arbitrariness. Because it works from a narrowly technical view of what engineering entails, reverse engineering has a hard time telling us about the cultural work of engineers. It is telling that the word “culture” never appears in this piece about the contemporary state of the culture industry.

Inspired by Madrigal’s article, here are some notes on the consequences of reverse engineering for how we think about the cultural lives of engineers. As culture and technology continue to escape their designated places and intertwine, we need ways to talk about them that don’t assume they can be cleanly separated.

. . .



Ben Affleck, fact extractor.

There is a terrible movie about reverse engineering, based on a short story by Philip K. Dick. It is called *Paycheck*, stars Ben Affleck, and is not currently available for streaming on Netflix. In it, Affleck plays a professional reverse engineer (the “best in the business”), who is hired by companies to figure out the secrets of their competitors. After doing this, his memory of the experience is wiped and in return, he is compensated very well. Affleck is a sort of intellectual property conduit: he extracts secrets from devices, and having moved those secrets from one company to another, they are then extracted from him. As you might expect, things go wrong: Affleck wakes up

one day to find that he has forfeited his payment in exchange for an envelope of apparently worthless trinkets and, even worse, his erstwhile employer now wants to kill him. The trinkets turn out to be important in unexpected ways as Affleck tries to recover the facts that have been stricken from his memory. The movie's tagline is "Remember the Future"—you get the idea.

Paycheck illustrates a very popular way of thinking about engineering knowledge. To know about something is to know the facts about how it works. These facts are like physical objects—they can be hidden (inside of technologies, corporations, envelopes, or brains), and they can be retrieved and moved around. In this way of thinking about knowledge, facts that we don't yet know are typically hidden on the other side of some barrier. To know through reverse engineering is to know by trying to pull those pre-existing facts out.

This is why reverse engineering is sometimes used as a metaphor in the sciences to talk about revealing the secrets of Nature. When biologists "reverse engineer" a cell, for example, they are trying to uncover its hidden functional principles. This kind of work is often described as "pulling back the curtain" on nature (or, in older times, as undressing a sexualized, female Nature—the kind of thing we in academia like to call "problematic"). Nature, if she were a person, holds the secrets her reverse engineers want.

In the more conventional sense of the term, reverse engineering is concerned with uncovering secrets held by engineers. Unlike its use in the natural sciences, here reverse engineering presupposes that someone already knows what we want to find out. Accessing this kind of information is often described as "pulling back the curtain" on a company. (This is likely the unfortunate naming logic behind *Kimono*, a new service for scraping websites and automatically generating APIs to access the scraped data.) Reverse engineering is not concerned with producing "new" knowledge, but with extracting facts from one place and relocating them to another.

Reverse engineering (and I guess this is obvious) is concerned with finished technologies, so it presumes that there is a straightforward fact of the matter to be worked out. *Something* happened to Ben Affleck before his memory was wiped, and eventually he will figure it out. This is not *Rashomon*, which suggests there might be multiple interpretations of the same event (although that isn't available for streaming either). The problem is that this narrow scope doesn't capture everything we might care about: why *this* technology and not another one? If a technology is constantly changing, like the algorithms and data structures under the hood at Netflix, then why is it

changing as it does? Reverse engineering, at best, can only tell you the what, not the why or the how. But it even has some trouble with the what.

. . .



“Fantastic powers at his command / And I’m sure that he will understand / He’s the Wiz and he lives in Oz”

Netflix, like most companies today, is surrounded by a curtain of non-disclosure agreements and intellectual property protections. This curtain animates Madrigal’s piece, hiding the secrets that his reverse engineering is aimed at. For people inside the curtain, nothing in his article is news. What is newsworthy, Madrigal writes, is that “no one outside the company has ever assembled this data before.” The existence of the curtain shapes what we imagine knowledge about Netflix to be: something possessed by people on the inside and lacked by people on the outside.

So, when Madrigal’s reverse engineering runs out of steam, the climax of the story comes and the curtain is pulled back to reveal the “Wizard of Oz, the man who made the machine”: Netflix’s VP of Product Innovation Todd Yellin. Here is the guy who holds the secrets behind the altgenres, the guy with the knowledge about how Netflix has tried to bridge the world of engineering and the world of cultural production. According to the logic of reverse engineering, Yellin should be able to tell us everything we want to know.

From Yellin, Madrigal learns about the extensiveness of the tagging that happens behind the curtain. He learns some things that he can't share publicly, and he learns of the existence of even more secrets—the contents of the training manual which dictate how movies are to be entered into the system. But when it comes to how that massive data and intelligence infrastructure was put together, he learns this:

"It's a real combination: machine-learned, algorithms, algorithmic syntax," Yellin said, "and also a bunch of geeks who love this stuff going deep."

This sentence says little more than "we did it with computers," and it illustrates a problem for the reverse engineer: there is always another curtain to get behind. Scraping altgenres will only get you so far, and even when you get "behind the curtain," companies like Netflix are only willing to sketch out their technical infrastructure in broad strokes. In more technically oriented venues or the academic research community, you may learn more, but you will never get all the way to the bottom of things. The Wizard of Oz always holds on to his best secrets.

But not everything we want to know is a trade secret. While reverse engineers may be frustrated by the first part of Yellin's sentence—the vagueness of "algorithms, algorithmic syntax"—it's the second part that hides the encounter between culture and technology: What does it look like when "geeks who love this stuff go deep"? How do the people who make the algorithms understand the "deepness" of cultural stuff? How do the loves of geeks inform the work of geeks? The answers to these questions are not hidden away as proprietary technical information; they're often evident in the ways engineers talk about and work with their objects. But because reverse engineering focuses narrowly on revealing technical secrets, it fails to piece together how engineers imagine and engage with culture. For those of us interested in the cultural ramifications of algorithmic filtering, these imaginings and engagements—not usually secret, but often hard to access—are more consequential than the specifics of implementation, which are kept secret and frequently change.

. . .

"My first goal was: tear apart content!"

. . .

While Yellin may not have told us enough about the technical secrets of Netflix to create a competitor, he has given us some interesting insights into the way he thinks about movies and how to understand them. If you're familiar with research on algorithmic recommenders, you'll recognize the system he describes as an example of content-based recommendation. Where "classic" recommender systems rely on patterns in ratings data and have little need for other information, content-based systems try to understand the material they recommend, through various forms of human or algorithmic analysis. These analyses are a lot of work, but over the past decade, with the increasing availability of data and analytical tools, content-based recommendation has become more popular. Most big recommender systems today (including Netflix's) are hybrids, drawing on both user ratings and data about the content of recommended items.

The "reverse engineering of Hollywood" is the content side of things: Netflix's effort to parse movies into its database so that they can be recommended based on their content. By calling this parsing "reverse engineering," Madrigal implies that there is a singular fact of the matter to be retrieved from these movies, and as a result, he focuses his description on Netflix's thoroughness. What is tagged? "Everything. Everyone." But the kind of parsing Yellin describes is not the only way to understand cultural objects; rather, it is a specific and recognizable mode of interpretation. It bears a strong resemblance to structuralism—a style of cultural analysis that had its heyday in the humanities and social sciences during the mid-20th century.

. . .

Structuralism, according to Roland Barthes, is a way of interpreting objects by decomposing them into parts and then recomposing those parts into new wholes. By breaking a text apart and putting it back together, the structuralist aims to understand its underlying structure: what order lurks under the surface of apparently idiosyncratic objects?

For example, the arch-structuralist anthropologist Claude Lévi-Strauss took such an approach in his study of myth. Take the Oedipus myth: there are many different ways to tell the same basic story, in which a baby is abandoned in the wilderness and then grows up to unknowingly kill his father, marry his mother, and blind himself when he finds out (among other

things). But, across different tellings of the myth, there is a fairly persistent set of elements that make up the story. Lévi-Strauss called these elements “mythemes” (after linguistic “phonemes”). By breaking myths down into their constituent parts, you could see patterns that linked them together, not only across different tellings of the “same” myth, but even across apparently disparate myths from other cultures. Through decomposition and recomposition, structuralists sought what Barthes called the object’s “rules of functioning.” These rules, governing the combination of mythemes, were the object of Lévi-Strauss’s cultural analysis.

Todd Yellin is, by all appearances, a structuralist. He tells Madrigal that his goal was to “tear apart content” and create a “Netflix Quantum Theory,” under which movies could be broken down into their constituent parts—into “quanta” or the “little ‘packets of energy’ that compose each movie.” Those quanta eventually became “microtags,” which Madrigal tells us are used to describe everything in the movie. Large teams of human taggers are trained, using a 36-page secret manual, and they go to town, decomposing movies into microtags. Take those tags, recombine them, and you get the altgenres, a weird sort of structuralist production intended to help you find things in Netflix’s pool of movies. If Lévi-Strauss had lived to be 104 instead of just 100, he might have had some thoughts about this computerized structuralism: in his 1955 article on the structural study of myth, he suggested that further advances would require mathematicians and “I.B.M. equipment” to handle the complicated analysis. Structuralism and computers go way back.

. . .

Although structuralism sounds like a fairly technical way to analyze cultural material, it is not, strictly speaking, objective. When you break an object down into its parts and put it back together again, you have not simply copied it—you’ve made something new. A movie’s set of microtags, no matter how fine-grained, is not the same thing as the movie. It is, as Barthes writes, a “directed, *interested* simulacrum” of the movie, a re-creation made with particular goals in mind. If you had different goals—different ideas about what the significant parts of movies were, different imagined use-cases—you might decompose differently. There is more than one way to tear apart content.

This does not jive well with common sense ideas about what engineering is like. Instead of the cold, rational pursuit of optimal solutions, we have

something a little more creative. We have *options*, a variety of choices which are all potentially valid, depending on a range of contextual factors not exhausted by obviously “technical” concerns. Barthes suggested that composing a structuralist analysis was like composing a poem, and engineering is likewise expressive. Netflix’s altgenres are in no way the final statement on the movies. They are, rather, one statement among many—a cultural production in their own right, influenced by local assumptions about meaning, relevance, and taste. “Reverse engineering” seems a poor name for this creative practice, because it implies a singular right answer—a fact of the matter that merely needs to be retrieved from the insides of the movies. We might instead, more accurately, call this work “interpretation.”

. . .

So, where does this leave us with reverse engineering? There are two questions at issue here:

1. Does “reverse engineering” as a term adequately describe the work that engineers like those employed at Netflix do when they interact with cultural objects?
2. Is reverse engineering a useful strategy for figuring out what engineers do?

The answer to both of these questions, I think, is a measured “no,” and for the same reason: reverse engineering, as both a descriptor and a research strategy, misses the things engineers do that do not fit into conventional ideas about engineering. In the ongoing mixture of culture and technology, reverse engineering sticks too closely to the idealized vision of technical work. Because it assumes engineers care strictly about functionality and efficiency, it is not very good at telling stories about accidents, interpretations, and arbitrary choices. It assumes that cultural objects or practices (like movies or engineering) can be reduced to singular, universally-intelligible logics. It takes corporate spokespeople at their word when they claim that there was a straight line from conception to execution.

As Nicholas Diakopoulos has written, reverse engineering can be a useful way to figure out what obscured technologies do, but it cannot get us answers to “the question of why.” As these obscured technologies—search engines, recommender systems, and other algorithmic filters—are constantly refined,

we need better ways to talk about the whys and hows of engineering as a practice, not only the what of engineered objects that immediately change.

The risk of reverse engineering is that we come to imagine that the only things worth knowing about companies like Netflix are the technical details hidden behind the curtain. In my own research, I argue that the cultural lives and imaginations of the people behind the curtain are as important, if not more, for understanding how these systems come to exist and function as they do. Moreover, these details are not generally considered corporate secrets, so they are accessible if we look for them. Not everything worth knowing has been actively hidden, and transparency can conceal as much as it reveals.

All engineering mixes culture and technology. Even Madrigal's "reverse engineering" does not stay put in technical bounds: he supplements the work of his bot by talking with people, drawing on their interpretations and offering his own, reading the altgenres, populated with serendipitous algorithmic accidents, as "a window unto the American soul." Engineers, reverse and otherwise, have cultural lives, and these lives inform their technical work. To see these effects, we need to get beyond the idea that the technical and the cultural are necessarily distinct. But if we want to understand the work of companies like Netflix, it is not enough to simply conclude that culture and technology—humans and computers—are mixed. The question we need to answer is how.

. . .

Thanks to Taylor Nelms and Christina Agapakis for help with this.

