Mapping EDEN: Revealing the territory created by society and software

Daniel "Drex" Drexler

Center for Science, Technology and Society at Drexel University

Mapping EDEN: Revealing the territory created by society and software

Overview

Description

Mapping EDEN is an experiment in wedding sociological critique with a craft-tradition of Computer Science seeking artifacts and ways of making and doing that suggest fresh approaches to multiple disciplines. It comes from a sense that we cannot sufficiently articulate why software has such an substantial impact on our world and why our responses to software are so insufficient in the face of the social footprint of the impact of software. Software is made by people who live and breath in the same world that we live in. We must struggle to find the aspect within software development that makes it so disruptive to our social worlds. Perhaps this project can help.

Background

This project is inspired by a personal feeling that there is a wide gap between the practical realities of creating software and the impact of software on our shared social world.

Merits

This project begins from a place inspired by feminist thinkings' Situated Knowledges and proceeds from there. It hopes to highlight the perspectives contained within a particular technical tool, the qualities of those perspectives, who those perspectives are helpful and hurtful to. It also hopes to closely examine the quality of the tool and what things emerge particularly easily from that tool as well as what is unusually well hidden.

It will show how the particular source code of the project co-creates its shadow in social space. How the concerns of the technical and the constraints of the tools that are used to create the tool drive the qualities that emerge from the techne ecosystem.

Conceptual Framework

Situated Knowledges

> A model is worked, and it does work
>
> ---
>
> *Donna Haraway*
>
> *Staying with the Trouble: Making Kin*
>
> *in the Chthulucene p. 63*

Our judgments are influenced by our experiences. This basic human quality is at the center how we understand others and how we learn, among other things. Within the social sciences, scholars have been working to demonstrate that this quality exists both within people and within processes and tools. This project is most directly inspired by the vein of work descended from critical feminism in the eighties around scientific epistemologies. Specifically Haraway (1988) introduces the term "Situated Knowledges" to talk about the power of speaking 'from' a location, which allows others to speak on the same topic from another location without needing to directly contradict the first speaker. It is a practice that strives to go beyond the flaws of scientific epistemologies and speak about a better system. Haraway was responding to work from Sandra Harding, who later returned to the topic to outline what Harding (1992) calls "Strong Objectivity." In conversation with the idea of situated knowledges, Harding critiques the traditional scientific practice of excising the background of the producers of scientific knowledge by pointing out that retaining how knowledge-finders are situated in the world adds to our understanding of their work. Out of the conversation between these two feminist thinkers has emerged a productive lineage of feminist thinkers who take the question of identifying how a 'knower' (either themselves or a third party) is situated in the world and how that situation has impacted the knowledge they produce. A modern vision of this analysis can be seen in Harraway (1997)'s Modest_Witness@Second_Millennium.FemaleMan©_Meets_OncoMouse^{TM}, which breaks down the rhetoric and framing around genetics research fetishizes the particular element of the gene because it is useful to the industry and particular narratives. Haraway advocates a view of the gene that centers it in a wide web of biological

mechanisms that interact and combine to create living creatures. Beyond individuals and rhetoric, the design and function of tools also encode a perspective and a view of the world. Some tools, like telescopes, do this in obvious ways. Others, like the tools used to analyze populations, contain intents that are less obvious. Subramaniam (2014) describes how the eugenic legacy of population level analysis makes some findings easier to see than others. How tools developed with the assumption that they would reveal differences between races are disappointingly easy to misuse, purposefully or accidentally, and create results that suggest racial difference. That much of the bias in a standpoint comes from what is hard to see or invisible with a particular tool and that this is only a flaw (for tools are meant to make select things more visible) when that tendency goes unacknowledged. That a tool contains a bias is desirable (what use is a telescope that does not magnify one area of the world?) and only becomes problematic when that bias is mistaken for the nature of the world.

Software Studies

> [S]haring produces its own kind of
> moral and technical order"
>
> *Christopher Kelty*
> *Two Bits: The Cultural Significance*
> *of Free Software p. 118*

Software is a tool that emerges out of a sea of tools. The methods of its construction, verification, distribution and maintenance are varied. This variance of practice arises at least in part from the lack of universally agreed upon standards for the goals of software or the appropriate metrics for success (Ensmenger, 2012). A diversity of practice does not create a homologous blend of cultures and practices. Instead, each software community has its own set of values and goals in addition to wider ethics that other projects and cultures share. Software Studies, broadly, attempts to connect the these hyper-local values that emerge from culture and circumstance to the technical and deterministic qualities of a particular piece of software or a particular piece of code. Works in the field often focus on a particular software community or a set

of communities with shared interests. Kelty (2008)'s work focuses on highlighting common social structures among particular software communities that both enable and emerge from the environment of open source software. There is also work that describes "code/space": places where the fixed rules of code and the fuzzy rules of social understanding combine to create hybrid spaces (Bridle, 2018; Kitchin & Dodge, 2011). Bivens (2017) represents our final influence as she compares how social qualities emerge with underlying technical qualities. Are social limitations matched by technical limitations? How do these two qualities co-create one another?

Design

This project will use sociological analysis to explore and highlight the standpoint of a piece of software. While I break down the particular qualities of the software package - its goals, blind spots and distortions, I will also be looking at its source code to compare the technical system to the external interface. What aspects seem constrained by the technical tools used to structure the system? What qualities are unrelated to the technical details?

Fieldsite

> That virtual worlds are places means they can be fieldsites;
>
> ---
> *Tom Boellstorff*
> *Two Bits: The Cultural Significance*
> *of Free Software p. 107*

The () software package, developed by the Sahana Foundation (a non-profit) and deployed to manage disaster response is the focus of this project. It was initially developed and deployed by an alliance of companies in the Sri Lankan Information and Communications Technology (ICT) sector in response to the Indian Ocean Earthquake & Tsunami. In 2009 development was transferred to the Sahana Foundation, an American non-profit. The Foundation lists its mission as "Saving lives through information management solutions." They describe their software as part of a wave of

change in how humanitarian responses can use ICT to improve the quality of their response. They situate themselves in the middle of other Free Open Source Software (FOSS) ICT projects such as OpenStreetMap which focus on community engagement and incorporating user data into digital platforms. has been used in response to a number of disasters in the developing world as well as ongoing supply management programs in the developed world (Sahana Foundation, n.d.).

The software's homepage can be found here: `https://sahanafoundation.org/`. A demo version of it running on the Sahana Foundation can be found here: `http://demo.sahanafoundation.org/\acrshort{EDEN}/`. 's source code can be found here: `https://github.com/sahana/\acrshort{EDEN}`

EDEN's identity as a piece of software is enmeshed in an FOSS ecosystem. It is written in Python, an open source language and uses the Python web framework web2py as the foundation its functionality is built on. Its functionality is broken up into many different web modules, which are organized around nodes of shared functionality that allow the individual modules to communicate and avoid duplicating functionality. generally uses other FOSS tools wherever possible: both of its database options (MySQL and PostgreSQL) are FOSS databases that are also available for free. This use of Free Open Source Software technologies helps ensure that users pay minimal fees to use , as it and all of its resources are available free of charge. However, because FOSS products are developed with many audiences in mind, their approach to problems is general and those projects will make governance decisions that may force you to leave them.

The Project

Phase One.   The first phase of this project will be a wide survey of literature. Focusing on Science, Technology and Society works that analyze the standpoint and biases of other technical tools and moving into analysis of disaster The aim of this project is to put academic work in conversation with practical tools as well as put academic criticism in conversation with work to transform the same tools. This will

build a set of analytical frames that will be essential for the second phase of the project.

Phase Two.   The second phase will be be a co-constitutive process between social critique and technical analysis. I'll begin by examining a standpoint that adopts, writing about what that standpoint intends to do and what side effects it might have. Those outcomes will be compared to and put in conversation with, the technical systems used to realize the standpoint. Over time, a paper that holds the technical choices, social goals and impacts, blind spots and clarifying qualities of will emerge from this work. Some standpoints will prove uninteresting and some will be difficult to match with sociological analysis.

As I find and characterize these points of social-technical tension, they will guide modifications to the software package itself. These modifications might attempt to address weaknesses in the particular standpoint the software addresses. They might be used to 'queer' the original intent of the module being modified. Such 'queered' modules might highlight how narrowly sees the world, how much about lived experience is irrelevant to , or highlight conditions that would make it impossible to continue to use and how unrelated they are to the circumstances that would lead people to use . Some number of these modifications will be made to the source code, but some more may highlight the technically unavailable resources that might enable software and society to interact differently.

Inspiration

I don't know of anything else quite like this project.

The praxis of this project is heavily inspired by Rena Bivens' stellar work that smashes together the technical realities of a product with the social understanding of it to problematize it. Its perspective is strongly influenced by Donna Haraway's attention to the specific qualities that structure spaces where sociological work happens. It is hoping to recall Banu Subramaniam's powerful tools for thinking through how and why a particular thing impacts the world in a particular way.

Research Questions

What ways of knowing and doing are assisted with the design of ?

What epistemological traditions are attended to by 's methods for storing data?

How do the social qualities of the social spaces within emerge from within its

data technologies?

What qualities of are co-creations between social space and the technical tools

used to constitute the software?

Schedule

Phase	Quarter	Quality	Time Allocation	Goal
One	Fall	Reading	80%	800 pages read with notes
		Writing	10%	Notes and planning documents
		Coding	10%	Notes on program structure
Two	Winter	Reading	40%	Final selection of bibliography with additions based on notes and readings.
		Writing	30%	Outline for final paper with 20% of content finished
		Coding	30%	Rough modifications completed, but in need of polish
	Spring	Reading	10%	Unexpected additions to literature and gathering specific quotes from previously completed elements.
		Writing	40%	Final paper
		Coding	40%	Full branch of project with documentation and demonstration server.

Outcomes

The project will produce three artifacts: a paper, a piece of software and a git source-tracking repository.

The reference Git repository for this project is available at `https://github.com/aeturnum/masters_project`.

The Git Repository. The two primary artifacts and all the other ephemera (including this proposal) involved in producing them will be archived in a git repository. Git will allow the process to be recorded and observed both in real time and in hindsight, giving the project a shape and an arc and a character outside of its final results.

A PDF of this document can be found at `https://github.com/aeturnum/masters_project/blob/master/full_proposal.pdf`.

The LaTeX source file can be found at `https://github.com/aeturnum/masters_project/blob/master/full_proposal.tex`.

The Paper. The paper will provide formal analysis of the original form of the software being considered. It will engage with social science literatures from traditions such as feminist critiques of technologies, critiques of algorithmic technologies, critiques of (often but not exclusively scientific) epistemological processes and software studies. Its discussion and conclusion will describe the particular view the software renders of the world and also argue how alterations to that piece of software might change its purpose, its views or its impact.

New EDEN. The software artifact (with the tongue-in-cheek name of New) will be a modified version of the selected software whose modifications will be driven by the suggested alterations in the paper. These two software products (modified and unmodified) can serve to contextualize the paper and bound its theoretical claims to what is pragmatically possible for a software developer.

Goals

This project will create a new space where the intersections between technology and society are hyper-visible. It will help to highlight the qualities of technical objects that give rise to their socially visible qualities. It will hopefully inspire others to further explore the power of modifying software as a way of finding and highlighting ways in which that software intrudes into our social lives. It may even provoke more engaged and practical work in software studies that struggle the daily challenges of writing software.

Glossary

EDEN The Emergency Development ENvironment, developed by the Sahana Foundation, is an Open Source Humanitarian Platform which can be used to provide solutions for Disaster Management, Development, and Environmental Management sectors.. 2, 5, 6, 7, 8, 9

MySQL MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress.. 6

OpenStreetMap An open community of volunteers that maintain data about roads, trails, cafés, railway stations, and much more, all over the world.. 6

PostgreSQL PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.. 6

python Python is an interpreted, high-level, general-purpose programming language.. 6

web framework A heterogeneous set of software that allow a programmer to efficiently write and manage providing a service over the Internet. This could be a website or a mobile app (often it is both) or a go-between for other services. Examples include web2py and Django in Python or Phoenix in Elixir.. 6

Acronyms

FOSS Free Open Source Software. 6

ICT Information and Communications Technology. 5, 6

References

Bivens, R. (2017, June). The gender binary will not be deprogrammed: Ten years of coding gender on facebook. *New Media & Society*, *19*(6), 880–898.

Bridle, J. (2018). *New dark age: Technology and the end of the future.* Verso Books.

Ensmenger, N. L. (2012). *The computer boys take over: Computers, programmers, and the politics of technical expertise.* MIT Press.

Haraway, D. (1988). Situated knowledges: The science question in feminism and the privilege of partial perspective. *Fem. Stud.*, *14*(3), 575–599.

Harding, S. (1992). Rethinking standpoint epistemology: What is" strong objectivity?". *Centen. Rev.*, *36*(3), 437–470.

Harraway, D. (1997). *Modest_Witness@Second_Millennium.FemaleMan©_Meets_OncoMouse^{TM}*. Routledge New York.

Kelty, C. M. (2008). *Two bits: The cultural significance of free software.* Duke University Press.

Kitchin, R., & Dodge, M. (2011). *Code/space: Software and everyday life.* MIT Press.

Sahana Foundation. (n.d.). *Making chaos managable.*

Subramaniam, B. (2014). *Ghost stories for darwin: The science of variation and the politics of diversity.* University of Illinois Press.