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

> *[T]he "writing of technology" is by by no means universal; the opaque and stubborn places do not lie simply beneath technology, but are wrapped around and in it*
>
> *Adrian Mackenzie*
> *Cutting Code: Software and Sociality p. 181*

This project will examine Emergency Development ENvironment (EDEN), a Free Open Source Software (FOSS) project developed to manage disaster response. EDEN's makers, the Sahana Foundation, lists their mission as "Saving lives through information management solutions." They present EDEN as bringing the fruits of Information and Communications Technology (ICT) to humanitarian projects, part of a wider "Information and Communication Technologies for Development (ICT4D)" movement. Sahana lists "community engagement" as a goal and want to be seen as a 'user centered' project that gives users representation in digital spaces (Sahana Foundation, n.d.).

ICT4D and 'community driven' software attempts to focus on user agency, but like all software they emerge out of a social process in which the needs of the project's engineers and managers are most visible. Each EDEN release is a snapshot of the team's expectations about their users and the project's future (Mackenzie, 2006). Analyzing the material qualities of the tools used to create software can help us spread positive technological impacts more widely than they are currently felt.

I will look at both the interface and the internals of EDEN. What ways of being and doing are supported by the design of EDEN? What qualities might be impossible to represent? How are these standpoints impacted by the material qualities of the constitutive tools used to create EDEN?

Critical engagement with technoscience must have a second act. Does understanding standpoints within something help us change it? This project is a *situated intervention* into the code and design of EDEN (Zuiderent-Jerak, 2015). What can we learn by trying to change EDEN guided by the standpoints that we discover? What lessons can we draw about how to critique better by engaging with material realities?

This project hopes to realize the goals of Sismondo (2008)'s *engaged program* by bringing feminist epistemological critiques into conversation with the process-tool we call "software." I aim to produce new knowledge about *how software can change* and *how critique can productively contribute* to that change.

I focus on three questions:

1. What standpoints can be found within EDEN?

2. How are EDEN's standpoints driven by its constitutive technologies?

3. Through modifying EDEN, can we identify useful qualities of an applied standpoint critique?

Conceptual Framework

A model is worked, and it does work

Donna Haraway
Staying with the Trouble: Making Kin in the Chthulucene p. 63

Building software is as much about imagining the future and accounting for the past as it is about creating a particular product for a particular time. Though software projects produce executables (discrete tools that can be moved and sold), it is better understood as a process-tool which exists and develops within a social and commercial context whose concerns and constraints extend far beyond a given executable. Much of the work in software involves managing the relationships between the other process-tools on which it relies (databases, programming languages and so on), as well as the relationships the software has with its users, makers, and imagined future (Mackenzie, 2006). These relationships are necessarily managed from a perspective within the project. Software developers exist in a heterogeneous world of software creation and formation cultures which also drive decisions around the concerns and forms of the project (Ensmenger, 2012; Gabriella Coleman, 2012; Kelty, 2008). The experience from outside is difficult to understand from within, as each person encounters a given software in what Kitchin and Dodge (2011) call code/space: a place co-constituted by the qualities of the software and the context of the encounter. The ways in which software can shape how people see the world *through* it and how things are represented *within* it can depart dramatically from the designers' intent.

Software strives to show a "view from nowhere" when guiding its users. The reduction of the complexities of life into a manageable series of capta is why we use tools to make otherwise intractable situations tractable. These simplifications also present concerns: what information is being thrown away and how was it chosen? Each time a piece of software shows someone the world, we can understand that showing as being from a particular standpoint. That is, as having particular concerns and goals in showing *this* but *not that* (Harding, 1992). Such perspectives allow tools to fulfill their function while allowing us to avoid confronting power relations. To check the potential problematic outcomes we can use a *situated epistemological* approach to re-territorialize the hidden information. So re-territorialized, that information can be measured against the utility gained by hiding it (D. Haraway, 1988). To ignore this quality in tools can give rise to inadvertently viewing the world through constructs that render natural variation as racial difference or environmental factors as measurements of native purity (Subramaniam, 2014).

If standpoint analysis yielded simple, straightforward guidance then the task of improving problematic software would be simple. However, software has material qualities that aren't impacted by descriptions of their undesirable consequences. To change the impact of particular software on particular people in particular situations we must grapple with those material qualities (Bivens, 2017). Zuiderent-Jerak (2015)'s concept of *situated intervention* offers a powerful tool for learning how we can change real programs. Only by trying to apply the judgments from theory to artifacts that emerge out of the the process-tool of software can we learn to shape those judgments to be more useful to those who are building the original software.

The design of the project draws on Science, Technology & Society tradition of making & doing and specifically on Zuiderent-Jerak (2015)'s concept of *situated intervention*. It is deeply indebted to Bivens (2017) demonstration of the impact of the material nature of software and Mackenzie (2006)'s work on the Sociality of software.

Project Structure

> *That virtual worlds are places means they can be fieldsites;*
>
> *Tom Boellstorff*
> *Coming of Age in Second Life p. 107*

Outcomes

The project will produce three artifacts: a paper detailing some perspectives within EDEN and reflecting on the attempt to engage them, New EDEN (a modified version of EDEN) and a git repository that will have a history of the project.

The Git Repository. The two primary artifacts and all the other ephemera (including this proposal) involved in producing them will be publicly available in a git repository.

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

Phases

One. The first phase of this project will be a survey of literature. Focusing on Science, Technology and Society works that analyze the standpoint and biases of other technical tools and relevant work on digital humanitarianism and disaster recovery. I will also be familiarizing myself with the implementation and operation of EDEN itself. These theoretical tools are what I will engage the "text" of EDEN with in the second phase.

The git repository will begin having content added to it immediately.

Two. The second phase will be be a co-constitutive process between social critique and technical analysis. I'll search for perspectives within EDEN, analyzing how that perspective emerges from EDEN and impacts the world around EDEN. My focus will be perspectives that seem driven by the material technical realities of EDEN. These findings will be the center of the project. This will be a filtering process - some number of perspectives will need to be investigated and found uninteresting.

This phase will produce the first half of the paper.

Three. The perspectives found in phase two will guide modifications to the EDEN software package itself. These modifications will be made with the goal of engaging with EDEN on a material level and understanding the practicalities of trying to address encoded perspectives. The modified software, called "New EDEN", will be the second product of this project.

This phase will produce the second half of the paper and New EDEN.

Schedule

Phase	Quarter	Activity	Time Allocation	Goal
One	Fall	Reading	80%	800 pages read with notes
		Writing	10%	Notes and planning documents
		Coding	10%	Notes on EDEN 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
Three	Spring	Reading	10%	Unexpected additions to literature and gathering specific quotes from previously completed elements.
		Writing	40%	Final paper
		Coding	40%	Full branch of EDEN project with documentation and demonstration server.

Fieldsite Details

EDEN 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. EDEN 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.). It is written in the FOSS language Python and uses the FOSS web framework web2py to build its components. Its functionality is broken up into many different modules which group common functionality within a layer of access conventions. EDEN also uses FOSS tools wherever possible: its databases of choice (MySQL and PostgreSQL) are FOSS databases.

The software's homepage can be found here: `https://sahanafoundation.org/`.

EDEN's source code can be found here: `https://github.com/sahana/eden`

EDEN does not have a formal release system so this project will be done on the version represented at git commit 653f76aa574cb2dddb9053eef0f8d815156cb168.

Preparation of Researcher

As a Science, Technology & Society graduate student I've been engaging with analytical questions around the impact of tools and their power to structure space. A previous project of mine used a energy pricing tool as a jumping off point for a critique of approaching the question of energy supply financially and suggested an alternative or additive perspective and gave me experience engaging with the social implication of a technical system.

Previous to being a graduate student I obtained an undergraduate degree in computer science and worked professionally as a programmer for about eight years. I did most of my work writing and designing web services and generally used python for this purpose (like EDEN in both cases). I have experience using most of the tools and environments that EDEN utilizes and expect that this background will be an asset in implementing changes in phase three.

I enter this project with a practiced skepticism about the promises of technoscience from my STS education, which professional software experience supports. However, I believe that software is not magic and that, like Latour (1983)'s expanding laboratory, there is hope for us to reveal the man behind the technoscientific software curtain.

Goals

This project will try to be an example of Sismondo (2008)'s engaged program - connecting analysis to real world issues and expending our understanding of both. It also hopes to be an example of and a coherent argument for further interventions using STS theories in the real world. I hope to respond to D. J. Haraway (2016, p.55)'s statement that it "matters with which ways of living and dying we cast our lot" by pushing the discipline to become more materially engaged with the world we seek to understand.

Glossary

capta If data is every piece of information about an entity, capta are the semantically meaningfull elements within that data (Kitchin & Dodge, 2011). The data of DNA is the entire sequence and the capta would be (we think) the genes that encode for particular traits and proteins. 3

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. 5

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.. 5

python Python is an interpreted, high-level, general-purpose programming language.. 5

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.. 5

Acronyms

EDEN Emergency Development ENvironment. 2, 4, 5, 6

FOSS Free Open Source Software. 2, 5

ICT Information and Communications Technology. 2, 5

ICT4D Information and Communication Technologies for Development. 2

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.

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

Gabriella Coleman, E. (2012). *Coding freedom: The ethics and aesthetics of hacking.* Princeton University Press.

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

Haraway, D. J. (2016). *Staying with the trouble: Making kin in the chthulucene.* Duke University Press.

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

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.

Latour, B. (1983). Give me a laboratory and I will raise the world. en science observed: Perspectives on the social study of science (pp. 141-170). *Beverly Hills: Sage Publications.*

Mackenzie, A. (2006). *Cutting code: Software and sociality* (S. Jones, Ed.). Peter Lang Publishing.

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

Sismondo, S. (2008). Science and technology studies and an engaged program. *The handbook of science and technology studies*, *3*, 13–32.

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

Zuiderent-Jerak, T. (2015). *Situated intervention: Sociological experiments in health care.* MIT Press.