

Guide Description:

As part of the Techno-Galactic Software Observatory, we will collectively produce a Software Observation Guide [[so, not survival ... Software REVIVAL Guide - think important to insist on observation]]. The guide will lay out ways to achieve critical distances from the seemingly endless software systems that surround us and will offer practical tools for the tactical (mis)use of software, empowering/enabling users to resist embedded paradigms and assumptions. The guide will include a description of the myths and realities of software, a list of currently available methods for approaching software including their risks and benefits, and a glossary of terms. Participants will contribute to the guide by naming and describing methods of observing software, by nominating and defining terms for the glossary, and by sharing personal stories of close encounters with software. Friday afternoon proposal for Guide structure and contributions

Table of Contents

Introduction

Reflecting on/performing observation and frameworks

Related Readings

Bare Necessities

Related Readings

Becoming Software, different ways of doing software history

Origin stories, multiple births of software (when, where)

Definitions of software

* Jean Heuns, Thomas C, ...

* <http://etherbox.local:9001/p/multiple-software-axes>

* <http://etherbox.local/home/pi/etherdump/axis-matrix.diff.html>

* http://etherbox.local/home/pi/etherdump/problematisation_of_definitions.md.diff.html

Software as environment

* How to transform a sip-well into a fountain

Techniques of Observation

Methodologies

* <http://etherbox.local/home/pi/etherdump/side-channel-analysis.diff.html>

Tools

* wireshark

* oscilloscopes

* em probes

*

Close Encounters with Software

personal stories, anecdotes, journals, written reports from clinic

* http://etherbox.local:9001/p/jans_heuns

Test Environments

* <http://etherbox.local/home/pi/etherdump/templatestyleguide.md>

* <http://etherbox.local:9001/p/etherboxmanual.md>

Vocabulary

Discussion on Thursday afternoon

turning observation sideways invasive observation one minute

it is about self-tracking, another on knowledge production

attention to the unequal power relation observing software

would set up

situated observation

why this focus observation? (so what is our agenda / what

are our agendas) - linked to politics of technology and

power struggle - interested in methods for feeling and grant-

ing agency and authorization to talk about and engage

with software and technology - stop using software as a

user, to be able to annotate, to have a different relation.

breaking routine - interest in the act of observing itself,

to debug and its politics - stopping and observing, time

to watch the software unfold - the no distance: observa-

tion without distancing. [[two-slit experiment model of ob-

servation impacting system being observed ???] - ways

to understand how the world works and be involved or

extend the ways of being involved (not just the guys do-

ing the brilliant work) but all of us - the guide is important

for having a wider range of tools that are available to be

able to work with different software with different agen-

cies (not just CS, hackers, programmers) - different defi-

nitions of software providing possibilities for methodolo-

gies/techniques for observing - our methods for observa-

tion, like mapping, come with their luggage. output from

observation seems to require some relationship to repre-

sentation... - tool clinics: using observation as a tool for re-

flexion and transformation. difference to design-crit. Care

for not losing playfulness The idea that this person will not

defend is that the propose intimacy and trust. The act of

observation has a coincidence.

observation and speculation becoming ? how do you bring

speculation to the present? thomas and his references to

the real world: claustrophobic, here is where we are, the

good guys and bad guys, let's expose these categories

speculation in the sense of confusing and mixing who speaks

who ask and about what SOFTWARE IS NOT A READY MADE

THING becoming, and how, who does it help to bring fic-

tion? Karen Barad, Meeting the Universe Halfway: Quan-

tum Physics and the Entanglement of Matter and Meaning

https://library.memoryoftheworld.org/b/wZAfhaCuiCv_eMO_m8kIKws

... and what about critique? - how to make sure it is affirma-

tive critique - different spectra of critique (so task is to open

up the spectrum of critique) - every community has different norms of critique which come with a set of ground truth assumptions they will not question) - reparative vs paranoid forms of critique ((Eve Sedgewick - https://www.dropbox.com/personal?preview=Teaching%2FAMRS%2FReadings%2FSedgwick_paranoid+and+reparative+reading)

paranoid: You're fighting with ideologies, looking at gaps that can be attended to. empowering and reclaim (Isabelle Stengers) with critique we think we are empowering people but we may be reclaiming positions, find another position reparative: trying to understand, you are not distanced from the system, but you are with it and trying to be generous many channels to give input and feedback in free software but if it doesn't fit into a certain range / template there's no place to send it to

observation and critique: affirmative, reparative and not paranoid observation becomes a way to formulate the critique from understanding what is put in front of you you engage through observation, and not as an alien coming in

is it the difference between passive and active observation which one is the reparative?

how about kinds of action: actively not doing, passive, is also doing

who is the audience for this work of observation? for us, and who are we non experts and experts keeping the conversation flowing instead of a one-way channel of communication patience and generosity respectful and disrespectful accessibility

(depends on what you are looking at) confusion

what are our core values and ground truths? bringing our concerns and issues in the debate patience and generosity theory of power attention and care agency, choices take the time, time to observe confusion play situated observation and knowledge <http://www.staff.amu.edu.pl/~ewa/teaching/2013/what%20we%20know%20we%20do>

What to do with observations? what do observations do with us? immediate/intended/unintended consequences meta-observation/long duree of observation/impact observation relational observations

nice to have: what are the tools that we currently use -> see below techniques of observation

Mapping (?) observations atlas - collection of representations constellation is map/representation compass (ref. speculative design) - tool of navigation

Collection

Observation can be captured by devices

- A collection of software definitions Is it the scale of the observation?

"to make a system that allows observation only, it seems rather impossible"

Techniques of the observer - note taking -command line -debugging -browser dev tools - poking around -sniffing -conversation

distancing no distancing critique

Enactments/Forms of Observations Representations Simulations (Models?) Reparative Critique (Eve Sedgewick - https://www.dropbox.com/personal?preview=Teaching%2FAMRS%2FReadings%2FSedgwick_paranoid+and+reparative+reading)

Table of Contents:

Inspiring section from the SAS survival handbook; strategy (strategic software teaching) FAFMRS%2FReadings%2FSedgwick_paranoid+and+reparative+reading

observatory, finding our research site... reading the signs (signs of problems, traces of solutions, hidden paths of inquiry...) disasters (software breakdowns, reading the ruins, tactical practices of provoking a breakdown)

Myths and Realities

Paradigms and Assumptions (ex: user vs programmer. hardware vs software, . . .)

What are the paradigms of software? Kittler basically said: "the Software is a construction in this sense. The paradigm is: software Delegating blame and responsibility to software (how it migrates - Are we blaming the software because we cannot blame somebody else? What is logged by a system? How to read log files?

How to recognise a software system and what are its boundaries?

Manifestos of blaming/ Normativity of software (the many "right ways" * Manifestos as programmers' tools

agile: process as an outcome

implementing agile involves one guy reading the book

building software under the contingencies of a moving target

Task analysis: starting point software development

what is a task, how can this become entangled with more playful

Origin Stories - The Multiple Births of Software (and the consequences of each of these definitions, for how they can be observed, the perspectives they produce, with what agendas they are proposed)

Materiality, abstraction and the different axis - legal when are the first software patents? responsibility The break-up of AT&T and the end of its monopoly allowed to sell unix licenses <https://www.freebsd.org/doc/en/articles/bsd-gpl/unix-license.html> - professional jobs division of labor user/producer - as language, different from data - in hardware - etymology - personal histories - as individual experience with and feelings about software - as data to be sold as product - as data to be used in engineering of software - as a configuration - as a signal - through tools

Tools and concepts

* toolchain: not just crafting software from scratch but setting

* repositories:

* dependencies

package managers take care of dependencies

- reverse engineering
- clean room implementation
- follow the money (political economic analysis) ## Translocation (and or as conversion)
 - translators

A list of tools - wireshark - oscilloscopes (very cool ones) - https://www.dropbox.com/personal?preview=Teaching%2FAMRS%2FReadings%2FSedgwick_paranoid+and+reparative+reading

Recognizing software

How does software emerge and spread?

Detecting traces of software (on other surfaces, materials, physical spaces?)

Exercises for experiencing software (performing it, re-enacting, sensing, making it tangible)

Close encounters with software (journals for/from walk-in clinic . . . or before? after?)

Methods for interfering with software (recipes ?), 'tactical (mis)use' (draw it boundaries, count the layers inside)

Detournement of security tools for observation - conventional tools for observation used/applied/appropriated in a different way

On Contexts and Movements Between

Appendix

Affordances and limits of different distinctions/drawing lines, making axes for overlap ...

- ie. there is/what is the difference between data and execution of data, executability of data

Reader (= edited/updated/completed reader?)

<http://etherbox.local:9001/p/reader.md>

Douglas Adams

Margaret Atwood

Eve Sedgewick

I proposed another text from wendy chun in the reader pad

a map somehow, in time and space, of software observations

Topics PING : Network command to check the identity of a server pinging as a way to do observation (and also pong) ping as a command, to connect to a server

AND

pong (& mechanical pong - niklas roy ?)

ANIMALS (as logos, as observers, as ??)

* bugreporting / between observation and action

A list of existing tools for observing software The menagerie of tools including the Zebra and the Giraffe, wearing magnify glasses, software called Zookeeper. Why a giraffe? Because of the big neck? Why not birds?

* Apache Zookeeper

<<https://zookeeper.apache.org>>

* O'Reilly media book covers also are animals, one animal = one methods or programming language

Reporting strategy

Delegating blame and responsibilities to software "Computer said no" How to blame software (?) in different situations?

ways to critique systemic issues in software forms of critique

Some way to discover/see what are the boundaries of the software system?

A list of programming manifestos - ways developers reinvent the writing of software (Luis cringing when agile manifesto comes up. It is too hard to get out of because it never ends!) what is made agile ... moving targets

Testing and reporting are now included in the software development process, and more difficult to take separately. task analysis how to (re)define tasks and other software design process stages: flowcharts, wireframes, scenarios - this may be only applicable on conventional software, wonder how this way of observation may be applied in experimental and conceptual software (there were some interesting flowchart-"sjabloon" in the museum)

"agile in a narrow space"

Observation methods/tools related to toolchains and pipelines "the house of cards of software" and "when you update your system, one of the cards might fall" repositories and where things come from.

Observation methods/tools related to package management automated process that takes care of everything you need dependencies, checking dependencies (so also the opposite boxing/containerization) multi-platform, a way of

putting the program next to your

reverse engineering: you have insight into the methods

clean-room implementations: if a company wants to reproduce a product, but the people involved cannot be tainted by having seen the original one they are put into a "clean room" with the specification to avoid being legally tainted [so: methods for knowing it was really "from scratch". how do you know the room is clean]

Lets do an exercise using axis metaphors instead of line metaphors and see when and where things start blurring

MANUAL TO ETHERBOX MOVED HERE

<http://etherbox.local:9001/p/etherboxmanual.md>

NOTES ON WEDNESDAY AFTERNOON SESSION

<http://etherbox.local:9001/p/files.md> !!!!

NOTES ON THURSDAY SESSION <http://etherbox.local:9001/p/thursday> and on SIDE CHANNEL ANALYSIS <http://etherbox.local:9001/p/side-channel-analysis>