

Abstractions! How do I even?

Veit Heller

April 18, 2019

EnthusiastiCon 2019

Agenda

- Motivation!
- Michel Foucault—The Order of Things
- Robert M. Pirsig—Zen and the Art of Motorcycle Maintenance
- Summary!

Motivation!

Technophilosophy?

We're going to talk a lot about philosophy in this talk. None of it will be about the branches of philosophy that seem most notably intertwined with our profession: technophilosophy.

We build abstractions, always.

So do philosophers; they have a headstart of just a few millenia.

Technophilosophy!

In a sense, all philosophy is philosophy about technology, if the goal of technology is abstracting things.

But I don't have a PhD!

Me neither, and English isn't my first language either.

The beauty about a lot of philosophy is that it is very simple, because it deals with very simple things—but not easy things!

Why do I have to think about these things?

Because you work on them all day! And because it's fun!

Michel Foucault—The Order of Things

“[...] animals are divided into: (a) belonging to the Emperor, (b) embalmed, (c) tame, (d) sucking pigs, (e) sirens, (f) fabulous, (g) stray dogs, (h) included in the present classification, (i) frenzied, (j) innumerable, (k) drawn with a very fine camelhair brush, (l) *et cetera*, (m) having just broken the water pitcher, (n) that from a long way off look like flies [...]”

— Michel Foucault, citing Jorge Luis Borges, in “The Order of Things”

It is often better to have no system to have the wrong system.

Classifications are defined by the space between them, not by their label.

Order is statist: you have to impose rules, and all of them are arbitrary.

What makes order good? What makes it bad?

Good abstractions impose a sensible order. The space between concepts is well defined.

Why order?

If the order is intuitive, the abstraction will be intuitive as well.

Why abstractions?

A good abstraction is good because it imposes order. Good order is good because it provides abstraction.

This is a tautology, but it makes sense because this is how the mind operates: always classifying, putting in boxes, hiding details.

Robert M. Pirsig—Zen and the Art of Motorcycle Maintenance

Splitting and reassembling things

Lateral and literal thinking

Do abstractions have to hide their details?

The curious case of Git

Summary!

Why should I care, revisited

Techniques for writing better abstractions, summarized

A Commentated Reading List

- Michel Foucault—The Order of Things: not about abstractions per se, but nonetheless exciting!
- Robert M. Pirsig—Zen and the Art of Motorcycle Maintenance: one of the best books I've ever read, stock-full of marvelous writing, storytelling, and compelling philosophical ideas.
- Christopher Alexander—Notes on the Synthesis of Form: provides a beautiful system of judging tradeoffs and implementations of ideas.
- Zachary Tellman—Elements of Clojure: talks about names and idioms rather than how to build anything in particular, and is thus more useful than any other programming book I've read; not just for Clojure programmers.
- Guy Steele—Growing a Language: both a paper and a talk. It explores building technical abstractions from first principles, and is removed enough from technology to be generally useful.

Thank you!

Questions?

Slides at <https://github.com/hellerve/talks>