Abstractions! How do I even?

Veit Heller

EnthusiastiCon 2019

May 23, 2019

Agenda

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

Technophilosophy?

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

Abstractions

We build abstractions, always.

Abstractions

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.
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?

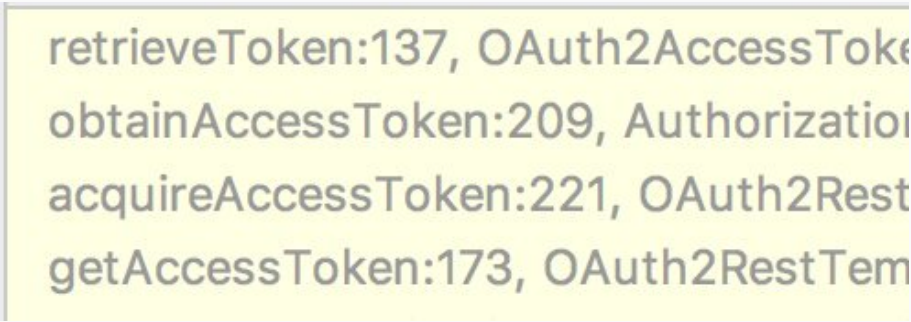
Because you work on them all day! And because it's fun!

Disclaimer

This book is not about abstractions, but about “episteme”, which are roughly the fundamental ideological axioms of all of knowledge—think paradigms, but without knowing that you adhere to them. This doesn’t make any sense without context, and the talk isn’t really about it anyway.
What it is about is how to impose order through abstraction.

Jorge Luis Borges

"[...] 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"


retrieveToken:137, OAuth2AccessToke
obtainAccessToken:209, Authorizatio
acquireAccessToken:221, OAuth2Rest
getAccessToken:173, OAuth2RestTem

Classifications

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

Why classifications?

If the classifications are intuitive, the abstraction will be intuitive as well.

Why classification?

Side note: it's impossible to abstract without classifying, because abstractions always add things or leave them out.

Why no classification?

Classification and order is hierarchy—don't let yourself be sucked into using hierarchies as a lens to see the world.

Disclaimer

This book is good, and reading it is very addicting!
It's about a father-son motorcycle trip, a life changed by mental illness, and the nature of "Quality".

The Three Things

There are three things in all dialogues: the subject, the object, and “Quality”, which happens between the two.

The Three Things

Quality is always between the observer and the observed.

The Three Things

Abstractions are always subjective. There is always a use case.

Objectivity in a subjective world

If everything is subjective, how do we empower opinionated users?

The curious case of Git

Git is the ultimate leaky abstraction.

The curious case of Git

It gives you control over the “porcelain”, or user-facing facilities, but also over the “plumbing”, and it does so gracefully.
I can `git commit`, but I can just as well `git commit-tree <tree> -p <parent-commit>`.

Do abstractions have to hide their details?

Only if you consider your abstractions to be good enough for everyone.

Why do I have to think about these things?

Thinking about how to write abstractions will make you reflect about your personal æsthetics and whether what you build actually holds up to your standards.

Why do I have to think about these things?

Reading philosophy will inspire you, or at least build a vocabulary that heps you articulate your æsthetics.

Techniques for writing better abstractions, summarized

A personal guidebook.

- Don't fear taking a hard stance if it leads to better classifications and the right abstractions. But be gentle to other people if they disagree.
- There should be no overlap between different function sets in your API. Remember: it's about clear boundaries.
- If the order is intuitive, the abstraction will be intuitive as well. But please write documentation anyway.
- If you're unsure about anything, talk to someone else. Bee tee dubbs: I'm always happy to chat about these things.

A Commentated Reading List I

- 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.

- Douglas R. Hofstaedter—Gödel, Escher, Bach: An Eternal Golden Braid: a weird and beautiful voyage into meaning, self-reference, and recursion.
- The Git Book, Chapter 10.1—Git Internals - Plumbing and Porcelain: is the start of a journey into weird Git internals that somehow ends up being extremely empowering and beautiful.

The End

Thank you!

Slides at `https://github.com/hellerve/talks`