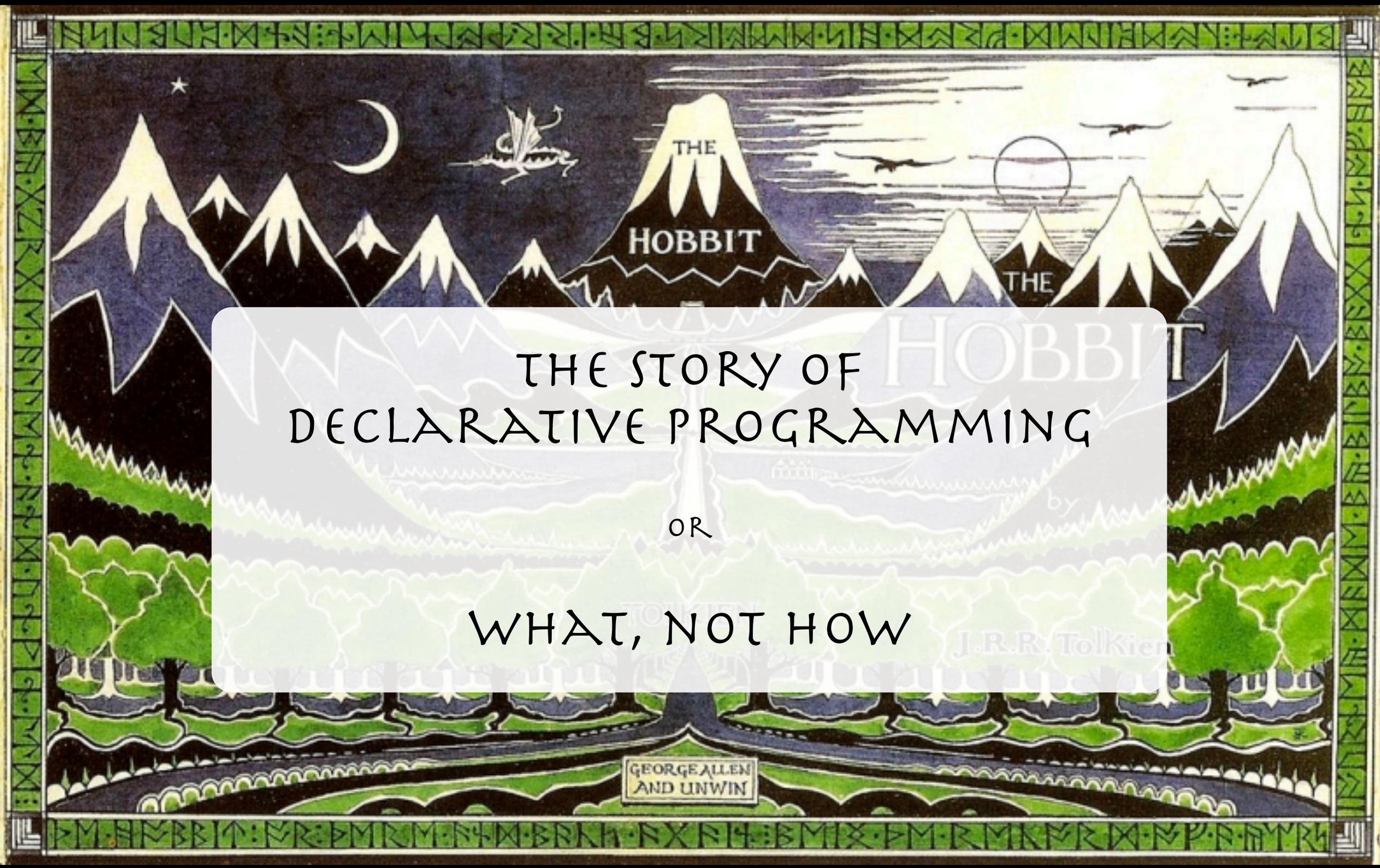
Propaganda!

A declarative programming model in Clojure

Thomas G. Kristensen

Agenda

- Declarative programming
- Current limitations
- The propagator model
- Business example
- In summary



THE STORY OF DECLARATIVE PROGRAMMING

OR

WHAT, NOT HOW


Don't Panic

In production?

Over 50% of business decision at uSwitch are made based on declarative programming models

In production?

Over 50% of business decision at uSwitch are made based on declarative programming models

not implemented by developers!

Limitations

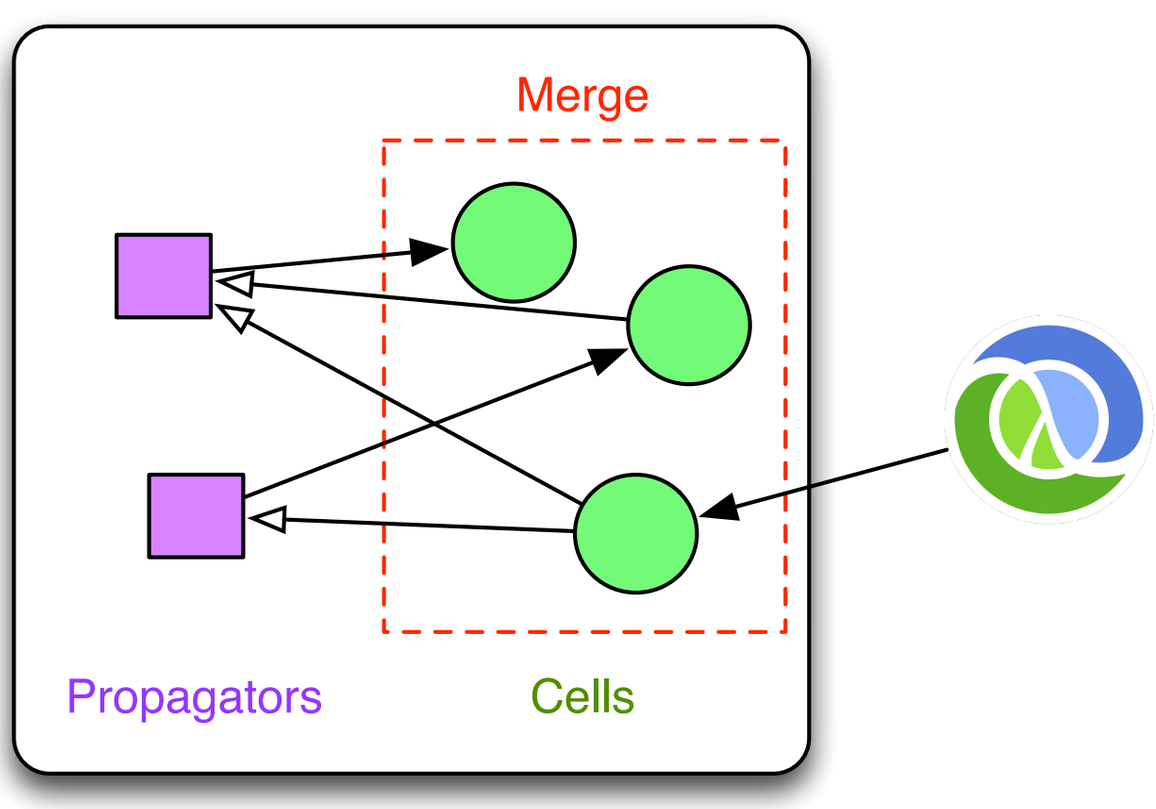
- We are locked into a grid
- We can't follow order of execution
- We can't extend cells with new datatypes
- We can't redefine how to update values in cells

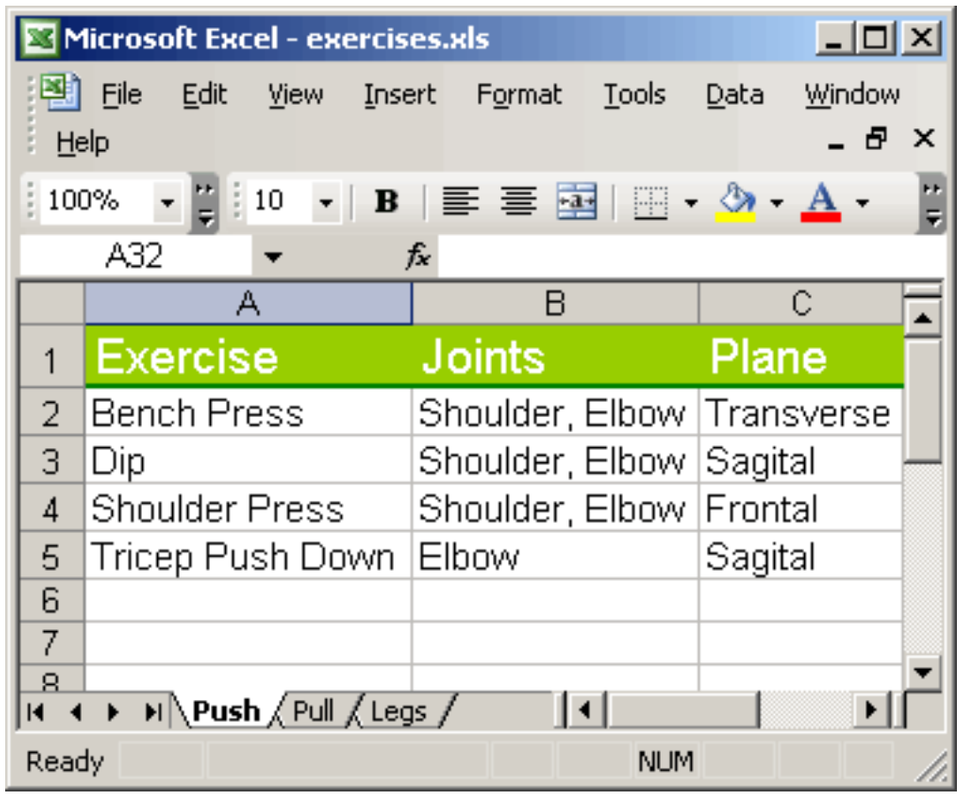
Propagators

- We can define our own cells
- We can track execution
- We can extend cells with new types
- We can observe cells and do arbitrarily complex propagation

For a completely different (more academic) motivation, see Clojure/conj

Analogy





screenshot (c) 1998

demo

Personal projection

Coming to a bill near you!

Model

```
personal-projection = standing-charge
                    + unit-rate x usage
                    - discounts
```

Disclaimer: This is extremely simplified. Attempting to implement the correct personal projection may result in permanent injury and enlightenment into the British energy market and its regulations.

demo

In summary

- Propagators are like spreadsheets - except when they are not
- We can define new cell value types and we can create complex propagators
- Runs on Clojure and ClojureScript

Thanks

```
[propaganda "0.2.0"]
```

More information

- github.com/tgk/propaganda
- tgk.github.io
- youtube: “clojure conj propagators”

We’re hiring!