Declarative Interaction

- Event-driven Functional Reactive Programming (E-FRP)
- mutable values as time-varying data streams
- event triggers propagation through dataflow graph
  but only for scalar values
- Streaming Database

Why Declarative

- Less code + faster iteration
- Performance + scalability
- Reuse + portability (flexibility)
- Programmatic generation

Imperative Interaction

1. Manually maintain state and dependencies
2. Side-effects
3. "Callback hell"

Demo: SPLOM of Iris

Based on Arvind's slides presented in InfoVis'15.

Talk is cheap. Show me the code

-Linus Torvalds

Future Work

Eco-system

Server-side computation

"Data tiles"?

workload partition?

--

Comments

• Declarative specification rocks
  • reusable, shareable (e.g., WebDesigner...)
  • elegant! (once learning the hotspot)

• E-FRP could be the next hotspot
  • similar to ReactJS
  • FP also

• Eco-system that speaks Vega
  • but Vega is not enough

• Open source

--

Comments

• Requires clear and well-ordered data
  • same as Tableau

• No way to debug

• Language level optimisation & runtime evaluation

• Tradeoff: Cognitive Dimensions of Notation

• Learning curve is quite steep

• Lack of community

• Foreign to FRP

---

Architecture:

Dataflow Graph

optional

~2x faster than D3

Full benchmark studies in the paper and online:

http://github.com/vega/vega-benchmarks