State Transitions in Complex, Scriptable Systems



Marijn Haverbeke

Code editors

- Lots of extensions
- Interdependent state
- Many types of updates



Modules managing imperative state



Modules managing imperative state



Modules managing imperative state

Atomic Updates



State as a value



State as a value



First-class transactions





$F_{n+1} = f(F_n, Tr, S_{n+1})$









State Configuration

- Determines set of fields
- Inherited by updated state
- > Update through transactions

Imperative Shell Functional Core



State/view data-flow

Incremental Updates

Principles: State

- Single immutable value
- Holds a set of fields
- > Atomically updated

Principles: Transaction

- Holds a set of effects
- Describes the entire change
- Only way to update state

Benefits

Single update codepath

- Composable
- Consistent by construction

codemirror.net marijnhaverbeke.nl

