State Machines

Declarative (Deterministic)

- No control flow
- What you want to accomplish, not how you do it
- Looks like RFC / description
- Much easier to reason about
- Abstraction, somewhere lies the procedural code

Imperative (Procedural)

- Explicit steps
- Conditionals

Imperative Example

Go out of the north exit of the parking lot and take a left. Get on I-15 North until you get to the 12th street exit. Take a right off the exit like you're going to Ikea. Go straight and take a right at the first light. Continue through the next light then take your next left. My house is #298.

Declarative Example

My address is 298 West Immutable Alley, Eden, Utah 84310

Finite-state Machines

It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some external inputs; the change from one state to another is called a transition. An FSM is defined by a list of its states, its initial state, and the conditions for each transition.

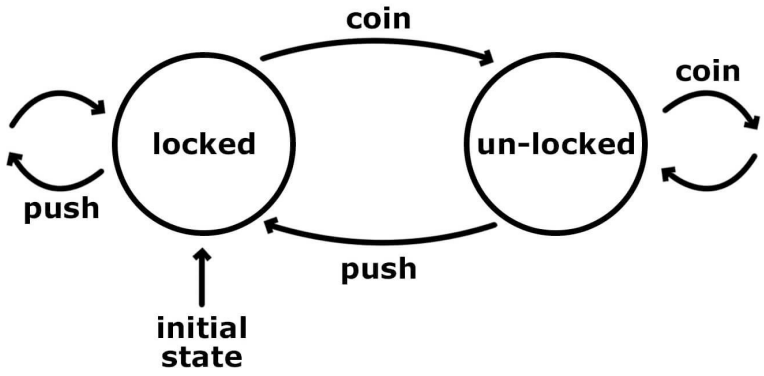
States

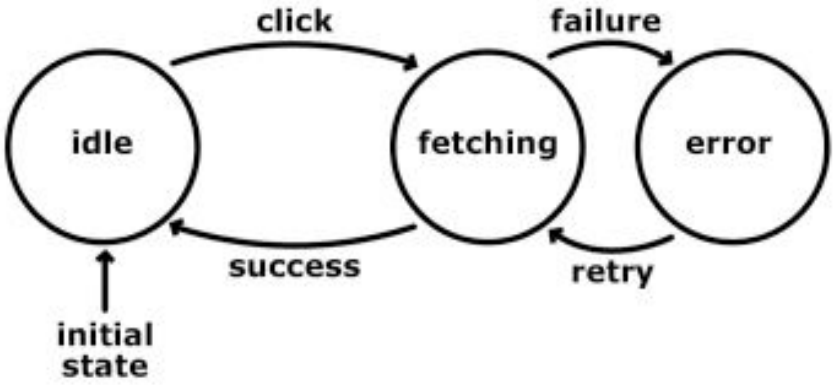
- Present-tense adjectives (locked, open, enteringPassword, flowCompleted)

Transitions

- Past tense event descriptions (COIN_INSERTED, EXIT_INITIATED, PASSWORD_ENTERED)
- Decoupled from state

Harel State Tables





State Chart extensible Markup Language (SCXML)

```
<state id="fetchMachine" initial="idle">
   <state id="idle">
      <transition event="FETCH_INITIATED" target="fetching"/>
   </state>
   <state id="fetching">
      <transition event="FETCH_FULFILLED" target="idle"/>
      <transition event="FETCH_REJECTED" target="error"/>
      <onentry>
         <action="fetchContent" />
       </onentry>
   </state>
   <state id="error">
      <transition event="RETRY_INITIATED" target="fetching"/>
   </state>
</state>


<state id=”conditionalExample">
  <transition event="e" cond="x==1" target="s1"/>
  <transition event="e" target="s2"/>
  <transition event="*" target="s3"/>
</state>
```

X State

View code examples