**Goal:** Raise productivity of non-programmer experts with visual programming languages that give constant (live) feedback about code changes.

**Problem:** Live languages require migrating run-time state with code changes.

**Approach:** Provide a reusable framework utilizing semantic deltas for designing and constructing stateful visual programming languages.

**References**

**State Machine Visual & Textual Definition**
```
Machine

State

: Trans
event: “close”

: Trans
event: “open”

State

machine doors
state closed
open => opened
state opened
close => closed
end
```

**State Machine Run-Time**

```plaintext
<table>
<thead>
<tr>
<th>State</th>
<th>#</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>closed</td>
<td>4</td>
<td>[open]</td>
</tr>
<tr>
<td>opened</td>
<td>3</td>
<td>[close]</td>
</tr>
</tbody>
</table>
```