Models of User Interfaces

- It is a very straightforward and natural way to model user interfaces using the concepts and notations of state machines.
  - Dialogs/Screens correspond to states,
  - Effects leading from one screen to another correspond to state transitions.

- The idea never quite made it, but has proven die hard.
  - The very first book on UIs ever (published 1971) already contained that idea.
  - The idea has been re-invented many times over, with different spins to it and for different types of (then modern) UIs: 3270 Dialogs, Web-pages, APPs,...
  - However, the idea also never quite became the dominant way to create UIs.

Why has this approach not prevailed? What are the problems – and can we fix them?
Landmarks of Using SMs for UI-Design

- **Paper Prototypes**
  - **Strengths:** very accessible & fast, cheap, inclusive/participatory
  - **Weaknesses:** not executable, not scalable, limited visual fidelity range

- **Silk/Denim**
  - **Strengths:** sketch input, generates HTML mock-ups
  - **Weaknesses:** not scalable, limited visual fidelity range, niche solution

- **Floella et al.**
  - **Strengths:** simple to use, very accessible & fast,
  - **Weaknesses:** not scalable, limited visual fidelity range, niche solution

- **Web Engineering**
  - **Strengths:** full-fledged code generation,
  - **Weaknesses:** expensive input, not scalable, limited visual fidelity range

---

Previous Approaches 1: Martin
Previous Approaches 2: Denert

Previous Approaches 3: Landay/Myers
WEDs /AIDE

- Our approach to address previous shortcomings consists of two elements:
  - Window/Event Diagrams (WEDs, since 1999 [1,2])
  - The Advanced Interaction Design Environment (AIDE, since 2008 [3,4])

- Together, they bundle contributions to achieve synergies.

1. **Input by Pen & Paper**
   - Best usability
   - great cost/benefit ratio
   - include diverse audiences

2. **UML State Machines**
   - concurrent states
   - history states
   - Rich events
   - extensions

3. **Navigation Support**
   - Locator, Toolbox, Zoom/Pan, ...
   - Accelerators
   - Diagram layers

4. **Executable Prototypes**
   - Simple HTML-mockups
   - executable XUL code
1. **Input by Pen & Paper**

   Issue New Reader Card
   
   1. Read
   2. Find
   3. Replacement Card
   4. Pick up at library
   5. Mail to home address

2. **UML State Machines**

2 UML State Machines

3 Navigation Support

- UI-Models become very large (and impossible to handle), quickly.
- We provide a multitude of tools for moving in large diagrams ("every known technique"):
  - Fully interactive locator view, Zoom/pan with multiple ways of interaction
  - Movable toolbox, keyboard accelerators, short-hands ("quick button", ...)
  - Jump-marks, color clouds, looking glass
- Syntactic Diagram Layers

<table>
<thead>
<tr>
<th>Orthogonal Models</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>- orthogonal angles or viewpoints</td>
<td>- n layers replace 2ⁿ separate views</td>
</tr>
<tr>
<td>- independent contributions (e.g., comments)</td>
<td>- merge n contributions while keeping traces</td>
</tr>
<tr>
<td>- optional aspects, features, or qualities</td>
<td>- easy combination of subsets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alternative Models</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>- competing variants</td>
<td>- avoid redundancy in overlapping submodel</td>
</tr>
<tr>
<td>- diverging versions</td>
<td>- easy switching between alternatives</td>
</tr>
<tr>
<td>- overlapping branches</td>
<td>- combine n branches in a single diagram</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequential Models</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>- consecutive steps or stages</td>
<td>- provide connected layouts</td>
</tr>
<tr>
<td>- increasing/decreasing levels of detail (zoom)</td>
<td>- avoid cascading changes</td>
</tr>
<tr>
<td>- temporal evolution of model</td>
<td>- allow also for branching evolution</td>
</tr>
</tbody>
</table>
Orthogonal Layers

- Options / Features
- Viewpoints
- Review Opinions

Alternative Layers

- Competing Variants
- Diverging Versions
- Overlapping Branches
Sequential Layers

- Consecutives Stages
- Zoom Levels
- Temporal Evolution

3 Navigation Support

Original Design of the implementation
4 Executable Prototypes

Results (1/3)
Toolchain from Sketches to Prototypes
Results (2/3)
Observational studies on WEDs

- Physical Interaction
- Embodied Cognition
- Maintain focus across zoom levels
- Integrate Overview and Details
- Allow concurrent work

Results (2/3)
Observational studies on WEDs

- Support Communication
- Deictic Interaction
- Facilitates Presentation
- Provides Overview
- Division of labor
- Sense of Ownership
Results (3/3)
Large Scale Experiment

Summary / Future Work

<table>
<thead>
<tr>
<th>Approach</th>
<th>Tool</th>
<th>Improvement</th>
<th>Evaluation</th>
<th>Dissemination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts and Notation for WEDs</td>
<td>Input by scanned sketches</td>
<td>Ongoing incremental usability improvements</td>
<td>Analyse observations from case study (grounded theory)</td>
<td></td>
</tr>
<tr>
<td>Incremental and Participatory UI Design</td>
<td>Extensive support for zooming and navigation</td>
<td>Android app for input/update, Automatic deployment</td>
<td>Define and conduct follow-up experiments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code Generation for HTML5 and XUL</td>
<td>XAML code generation, mixed code, frames/resolutions</td>
<td>Run and evaluate realistic field trial</td>
<td></td>
</tr>
</tbody>
</table>
Appendix:
A Simple Smartphone Library Client

PST-Hütte 2.7.2014
States
Initial/Final States
Transitions

Switch off

Switch on

TCL Online App

Sequential Substate
Explicit Actions
High level entry/exit

Switch off

TCL Online App

(c) 2014, H. Störrle