STATECHARTS: A Visual Formalism for Complex Systems

Alessandra Nardi
Fan Mo
EE249 Discussion Section
September 14, 1999

STATECHARTS - 1

- Visual formalism for the specification of reactive systems
  - not a support, but the way itself

- Extension of State Transition Diagram of Finite State Machines
  - State/Event Description
STATECHARTS - 2

- Flat FSM - too complex

- Mechanisms to enhance the descriptive power:
  - Hierarchy
  - Orthogonality

HIERARCHY - 1

- Boxes denote states
- Encapsulation expresses hierarchy
  - different state levels
- Arrows can originate and terminate at any level
- Arrows labeled with events
  - optionally: parenthesized conditions
HIERARCHY - 2

- XOR Decomposition
- Way to economize arrows

HIERARCHY - 3

- Default states
HIERARCHY - 4

- Entering a group of states - History
  - State most recently visited

HIERARCHY - 5

- Entering a group of states - History
  - Only at one level
**HIERARCHY - 6**

- Entering a group of states - History*
  - through the hierarchy

**HIERARCHY - 7**

- Entering a group of states - Condition
HIERARCHY - 8

- Entering a group of states - Selection

HIERARCHY - Summary

- XOR Decomposition
- Default states
- Entering a group of states
  - History
  - History*
  - Condition
  - Selection
ORTHOGONALITY - 1

- Boxes splitted by a dashed line express orthogonality
- Independency and/or Concurrency

ORTHOGONALITY - 2

- AND Decomposition
- Way to economize states

\[ N = N_1 \times N_2 \]
MORE FEATURES....

- Time Constraints
- Unclustering
- Actions and Activities

TIME CONSTRAINTS

- Delays
  - timeout(event, number)
- Time Bounds

\[ \Delta t_1 < \Delta t_2 \]
UNCLUSTERING

- Laying out parts outside the natural neighborhood

ACTIONS AND ACTIVITIES

- Connection with the “real world”
  - ACTIONS (zero time)
  - ACTIVITIES (non-zero time)

- Actions to control activity X
  - start(X)
  - stop(X)
ACTIONS AND ACTIVITIES

- Extension of State Transition Diagram

SUMMARY

- Extended State Transition Diagram
  - Hierarchy
  - Orthogonality
- Default states
  - enter by (history, condition, selection)
- Time constraints
- Actions and Activities
WATCH

WATCH INTERFACE

■ OUTPUT
- Main Display Area, 4 Smaller Display Areas
- Two-Tone Beeper

■ INPUT
- 4 Control Buttons: a, b, c, and d
- Battery
WATCH FUNCTION

- Display Time (am/pm or 24 hour)
  or Display Date (day, month, date of week)
- Two Independent Alarms
- A Stopwatch (lap and regular modes, and a 1/100 s display)
- A Light for Illumination
- A Weak Battery Blinking Indication
- Beeper Test

WATCH USER’S GUIDE

<table>
<thead>
<tr>
<th>Button</th>
<th>Current State</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Normal</td>
<td>Select displays (time/date, alarm setting, chime setting, stopwatch)</td>
</tr>
<tr>
<td>b</td>
<td>Normal</td>
<td>Press with button d enter Beep-test</td>
</tr>
<tr>
<td></td>
<td>Update or Setting</td>
<td>Exit current update or setting</td>
</tr>
<tr>
<td></td>
<td>Stopwatch</td>
<td>Run/Stop</td>
</tr>
<tr>
<td>c</td>
<td>Update or Setting</td>
<td>Select Update Content</td>
</tr>
<tr>
<td>d</td>
<td>Time</td>
<td>Time/Date switch</td>
</tr>
<tr>
<td></td>
<td>Alarm Setting</td>
<td>Turn on/off</td>
</tr>
<tr>
<td></td>
<td>Chime Setting</td>
<td>Turn on/off</td>
</tr>
<tr>
<td>Normal</td>
<td>Press with button b to enter Beep-test</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stopwatch Run</td>
<td>Regular/Lap display</td>
</tr>
<tr>
<td></td>
<td>Stopwatch Stop</td>
<td>Clear stopwatch to zero</td>
</tr>
<tr>
<td></td>
<td>Update or Setting</td>
<td>Resume</td>
</tr>
</tbody>
</table>
STATECHART: watch

STATECHART: main//alarm1-status//alarm2-status//chime-status//light//power

We are here
STATECHART: update

We are here

STATECHART: alarm1, update1

We are here
STATECHART: chime

- chime
  - off
  - on

We are here →

STATECHART: stopwatch

- stopwatch
  - zero
  - display
    - reg
    - lap
  - run
    - on
    - off

We are here →
STATECHART: alarms-beep

alarms-beep

- alarm1 beeps
- alarm2 beeps
- both beep

- T is T1
- any button pressed
- 30 sec in alarms-beep

T is T2 (P2)

T is T1 (P)

T1: time setting of alarm1
T2: time setting of alarm2
P1: alarm1_enabled ^ (alarm2_disabled V T1!=T2)
P2: alarm2_enabled ^ (alarm1_disabled V T1!=T2)
P: alarm1_enabled ^ alarm2_enabled ^ T1=T2

STATECHART: alarm1-status

alarm1-status

- disabled
- enabled

- d (in alarm 1.off)
- d (in alarm 1.on)

We are here →
STATECHART: alarm2-status

alarm2-status

disabled

enabled

d
(in alarm 2.off)

d
(in alarm 2.on)

STATECHART: chime-status

chime-status

disabled

enabled

quiet

beep

T is whole hour

2 sec in beep

We are here →

We are here →
STATECHART: light

We are here

STATECHART: power

We are here