State modeling

Marlon Dumas
Institute of Computer Science
<table>
<thead>
<tr>
<th>Notation</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class diagrams</td>
<td>Domain &amp; application modeling</td>
</tr>
<tr>
<td>Use case diagrams/use cases</td>
<td>User-system interaction modeling</td>
</tr>
<tr>
<td>Sequence diagrams</td>
<td>User-system &amp; system-system interaction modeling</td>
</tr>
<tr>
<td>Statecharts</td>
<td>Object behavior modeling</td>
</tr>
<tr>
<td>Petri nets</td>
<td>Concurrent systems &amp; process modeling</td>
</tr>
<tr>
<td>Decision trees &amp; DMN</td>
<td>Decision modeling</td>
</tr>
</tbody>
</table>
Capturing Object Behavior

- Typically the state of an object changes in response to stimuli (“events”)

- Scholarship application:
  - A student creates the application (draft)
  - The student can withdraw it (withdrawn)
  - Or the student can submit it (submitted)
    - It can still be withdrawn!
  - A secretary can cancel it because of eligibility rules (cancelled)
  - A committee accepts or rejects it (accepted or rejected)
    - If accepted the student can still withdraw it?!
  - If accepted, the accounting department disburses it (disbursed)
State Machine

- A machine whose output behavior is not only a direct consequence of the current input, but of some past history of its inputs.
- Characterized by an internal **state** which represents this past experience.

If the phone is ON, then pushing this button will turn it off.

If the phone is OFF, then pushing this button will turn it on.
State Machine (Automaton)

Initial state (entry)
Phone is on
State
Transition (event “off”)
Phone is off

on
off
Another example
Exercise 1

- Model the scholarship application as a state machine
  - The student creates the application (draft)
  - The student can withdraw it (withdrawn)
  - Or the student can submit it (submitted)
    - It can still be withdrawn!
  - A secretary can cancel it because of eligibility rules (cancelled)
  - A committee accepts or rejects it (accepted or rejected)
    - If accepted the student can still withdraw it
  - If accepted, the accounting department disburses it (disbursed)
Actions on the State Machine

- State changes can induce side-effect actions

**Mealy automaton**

**Moore automaton**
Extended State Machines

- State machine with variables

![State machine diagram]

- Light is off
- Light is on

\[ c := c + 1 \]

- \( c : \text{Integer} \)
Consider the state machine of an answering machine

Add the following events and actions: Call detected, Answer call, Play announcement, Record message, Caller hangs up, Announcement complete

Revise the state machine so that the machine answers after five rings.
Statecharts

- The statecharts notation extend state machines with:
  - Various types of events and conditions
  - State hierarchy (statecharts inside statecharts)
  - Concurrency
  - Other “cool” features we’ll see…

- Part of UML
- Heavily used in SysML
A state model consists of one or more state diagrams.

State

Transition
States

- A state is an abstraction of attribute values and links of a particular object
  - An object has a finite number of possible states
  - It can only be in one state at a time

- Solvent
- Minimized
- Maximized
- Insolvent
- Waiting for authorization
- Decision notified
Events

- An event is a “stimulus” that can trigger a state change of an object

- Kinds of events
  - Call event
  - Signal event
  - Change event
  - Time event
1. Call events

- A call event represents the reception of a request to invoke a specific operation.

```
ShoppingCart
<table>
<thead>
<tr>
<th>subtotal</th>
<th>vat</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>placeOrder</td>
<td>cancelOrder</td>
<td>addProduct</td>
</tr>
</tbody>
</table>

Product
| name | description | photo |

Purchase
| quantity | pricePerUnit |
```

```
ShoppingCart

CartIsEmpty
| removeProduct |

Shopping
| removeProduct |
| placeOrder |
| cancelOrder |
| addProduct |
```
2. Signal Event

- A signal is an explicit one-way transmission of information from one object to another
  - A signal event is asynchronous
  - A call event is a two-way synchronous communication
- Signal events can be specified as UML classes

<table>
<thead>
<tr>
<th>«signal» FlightDeparture</th>
</tr>
</thead>
<tbody>
<tr>
<td>airline</td>
</tr>
<tr>
<td>flightNumber</td>
</tr>
<tr>
<td>city</td>
</tr>
<tr>
<td>date</td>
</tr>
<tr>
<td>time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>«signal» MouseButtonPushed</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
</tr>
<tr>
<td>location</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>«signal» SelectionChanged</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetControl</td>
</tr>
<tr>
<td>selectionIndex</td>
</tr>
</tbody>
</table>
3. Change events

- A change event is an event that is caused by the satisfaction of a boolean expression
  - UML specifies that the expression is continually tested
  - An implementation would not continuously check the expression, but at least often

Examples:
- when (room temperature < heating set point)
- when (room temperature > cooling set point)
- when (battery power < lower limit)
- when (tire pressure < minimum pressure)
4. Time event

A time event is an event that is caused by the occurrence of an absolute time or the elapse of a time interval

Examples absolute time:
- at (January 1, 2010)
- at (20:00)

Examples time interval:
- after (10 seconds)
- after (10 days)
Guard conditions

- A transition is the change from one state to another
  - E.g. A phone line transitions from “Ringing” state to “Connected” when somebody picks the phone up
- A boolean expression can be used to add constraints in the firing of a transition
  - Interesting when more than one transition can be selected at a given time
Transition effects and do-activities

- Mealy automata actions correspond to UML statecharts’ transition effects
  - A transition effect can be an assignment or the call to an operation

![Diagram showing transition effects]

- UML statecharts can also specify actions attached to state nodes (as for Moore automata)
  - A “do-activity” is an activity that should execute continuously for an extended time
Exercise 3: complete this statechart

An order cannot be placed unless the cart is not empty

ShoppingCart

CartIsEmpty

Shopping

placeOrder

removeProduct

cancelOrder

addProduct

An entry activity is performed whenever the state is entered

Whenever the state is exited, by any outgoing transition, the exit activity is performed first.
Example: Flashing Light Bulb (to be developed in the practice session)
Order of activities

After first “off” event
- print(“exiting”)
- print(“to off”)
- turnLightOff()

After second “off” event
- print(“exiting”)
- print(“needless”)
- turnLightOff()
Event handling and self-loops

In this case “off” event is handled bypassing both the entry and exit activities

- print(“skipped”)

LightIsOff

entry/turnLightOff()
exit/print(“exiting”)
off/print(“needless”)

LightIsOff

entry/turnLightOff()
off/print(“skipped”)
exit/print(“exiting”)

In this case “off” event is handled bypassing both the entry and exit activities

- print(“skipped”)

LightIsOff

entry/turnLightOff()
exit/print(“exiting”)
off/print(“needless”)
Readings & Resources

Material covered in lectures:

- This week: Blaha & Rumbaugh, Chapter 5
- Next week: Blaha & Rumbaugh, Chapter 6

For the practice sessions:

- Yakindu StateCharts Tool (SCT)
  - [http://statecharts.org/download.html](http://statecharts.org/download.html)
Exercise 4 (next week)

- Prepare the state diagram for a washing machine
  - On / Off button
  - Start button (but no stop button)
  - Feedback is given on the current stage (soaking, rinsing, draining, drying)
- Three washing programs
  - Regular
  - Delicate (no soaking)
  - Super delicate (no soaking, no drying)
- Off can be clicked only before starting, or after finishing
A simple digital watch has a display and two buttons to set it, the A button and the B button. The watch has two modes of operation, display time and set time. In the display time mode, the watch displays hours and minutes, separated by a flashing colon.

The set time mode has two submodes, set hours and set minutes. The A button selects modes. Each time it is pressed, the mode advances in the sequences: display, set hours, set minutes, display, etc. Within the submodes, the B button advances the hours or minutes once each time it is pressed. Buttons must be released before they can generate another event. Prepare a state diagram of the watch.