State transition modeling
Basic concepts (1)

• **state** – a situation during an object life time, when it satisfies some condition, performs some activity, or waits for some event

• **event** – occurring of some situation which is important in some context; time of an event is atomic
Basic concepts (2)

• **action** – an atomic operation execution; atomic in the current time scale

• **activity** – operation executed by an object staying in some state until it is interrupted by an event
Basic concepts (3)

- **state transition** – a relationship between two states of an object; the object in the first state will perform certain actions and enter the second state when a specified event occurs and specified conditions are satisfied.

- **state machine** – an abstract machine performing object state transitions on events; represented by state transition diagram.
State transition diagram

- Initial state: Off
- Final state: Shut down
- Event: Switch On
- Event: Switch off
State transition signature

event [guard condition] / action expression
Event kinds

- Call event
- Signal event
- Change event
- Time event
Call event

• An event occurring when an operation should be performed

operation name (parameter list)
Signal event

- «signal» Mouse
  - pos: Point
- «signal» MouseDown
  - btn: Button
- «signal» MouseUp
  - btn: Button
- «signal» MouseMove
- «signal» MouseRoll
  - dist: Point
• An event occurring when some condition is satisfied

  **when** condition expression
Time event

• An event occurring:
  – after some time interval
    **after** time expression
  – in some time moment
    **when** time expression
Guard condition

• Boolean expression containing:
  – triggered event parameters
  – object properties and states
  – when an object is in some state (in state name)

• Guard condition is checked after triggering an event, and before firing a state transition
Action expression

- A sequence of operation calls sequentially executed
- Operations are separated with semicolons
- Operations arguments can use triggered event parameters
Transition example

MouseDown (btn, loc) [loc in Window]
/ object := WindowFindObject (loc); object.Select ()
Entry & exit actions

<table>
<thead>
<tr>
<th>Data maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>entry / connect to data base</td>
</tr>
<tr>
<td>exit / disconnect data base</td>
</tr>
<tr>
<td>do/ check unused records</td>
</tr>
<tr>
<td>include / internal state machine</td>
</tr>
<tr>
<td>get progress / display progress</td>
</tr>
</tbody>
</table>
Compound states

• Sequential substates
• Concurrent substates
Sequential and concurrent substates

Diagnostics

Diagnosing Circuit 1
Pass

Diagnosing Circuit 2
Pass

Failed

Diagnosing Circuit 3
Pass

Diagnosing Circuit 4
Pass

Failed

Passed
Synchronizacja stanów równoległych

Inicjalizacja

Stan A1 — Stan A2
Stan B1 — Stan B2

Finalizacja

belka synchronizacji
Activity diagram

• A special state transition diagram when:
  – state nodes represent actions or sub-activities
  – state transitions are fired just after the preceding activities are finished
Activity diagram example

- Authorization
- Choose operation
  - Operation = get balance
  - Operation = withdraw
  - Operation = exit
- Display balance
- Entering amount
- Check balance
  - Balance positive
  - Balance negative
- Counting money
- Preparing check
- Taking money
Concurrent activities

Object 1

Activity A1

Activity A2

Object 2

Initialization

Activity B1

Finalization

Object 3

Activity C1

Activity C2

“swimlines"
Bibliography