UML Diagram Types

Dynamic Models
- *activity diagrams*
- statechart diagrams
- interaction diagrams
  - sequence diagrams
  - collaboration diagrams
- use case diagrams

Structural Models
- class diagrams
- object diagrams
- packages

Architectural Models
- component diagrams
- deployment diagrams

Activity Diagram

def'n: dynamic model showing the flow from activity to activity (and action to action)

activity: ongoing non-atomic execution within a state machine

action: executable atomic computation that results in a change in state of a system or the return of a value
- call operation, send signal, create/destroy object, pure computation

Activity Diagram vs. Interaction Diagram

- Activity diagram
  - models flow of control from activity to activity (and action to action)
  - interaction diagram turned inside out
  - detailed view of an operation
- Interaction diagram
  - models flow of control from object to object
  - looks at objects that pass messages

Activity Diagram

- captures the critical path of the workflow
- kind of state machine

Convention

- modeled as vertices and arcs
- text may be structured text, or specific syntax of programming language
- may contain simple, composite states, forks, branches, joins

Contents of Activity Diagram

- Activity State

def'n: system state representing execution of (a series of) actions

- can be decomposed
- can be interrupted
- has duration
- may have entry and exit action
- Convention: lozenge shape

Contents of Activity Diagram

- Action State

def'n: system state representing execution of an action (atomic computation)

- cannot be decomposed
- cannot be interrupted
- execution time is insignificant
- special case of activity state (cannot be further decomposed)
- Convention: lozenge shape

Contents of Activity Diagram

- Transition
  - *def'n*: when action completes, flow of control passes immediately to next action or activity state
  - – triggerless: transition occurs once work in source state is complete, if guard is true
  - – Convention: simple directed line
  - – Types include: branching, fork/join, swimlanes

Transition Types

Branching

def'n: specification of alternate paths taken based upon some Boolean expression

- one incoming and two or more outgoing branches
- Boolean expression on each outgoing transition that is evaluated once (when entering branch)
- can use "else"
- can be used to indicate iteration
- structured text, or specific programming language

Transition Types

Fork/Join

def'n: specification of concurrent paths

- use synchronization bar to indicate
- convention: thick horizontal or vertical line
- fork: splitting of single flow of control into two or more concurrent flows of control
- join: synchronization of two or more concurrent flows of control
- joins and forks should balance
- parallel flows may communicate via signals (coroutines)

Transition Types

Swimlanes

def'n: specific, named locus of activities

- no deep semantics
- may represent some real-world entity
- every activity belongs to one swimlane, but transitions may cross lanes

Contents of Activity Diagram

- Object
  - can specify objects that are created, destroyed, or modified as part of the flow of control
  - can show how its role, state, and attribute values change

Common Uses of Activity Diagrams

- Model a workflow: focus on activities as viewed by the actors that collaborate with the system (business systems perspective)
- Model an operation: use as flowcharts to model the details of a computation
- Model order: sequence within state of statechart diagram

Model an Operation

- Collect abstractions involved in the operation (parameters, class)
- Identify preconditions and postconditions
- Model activities and actions over time
- Use branching for conditional paths
- Use forking and joining to indicate concurrent paths

Model an Operation

- UML is not a visual programming language
- Every operation can be flowcharted, but why would one want to?
- Model operation behavior that is complex and to difficult to understand by looking at code

Methods of Modeling an Operation

- Activity diagram
- Statechart diagram
- Flowchart
- Nassi-Schneiderman chart
- Pseudo-code
- Prose