## UML Activities & Actions

Charles ANDRE - UNSA

## Action & Object Nodes

#### Accept inputs, start behaviors, provide outputs



### Overview

- Activity modeling emphasizes the sequence and conditions for coordinating lower-level behaviors, rather than which classifiers own those behaviors.
- These are commonly called control flow and object flow models.
- The actions coordinated by activity models can be initiated
  - because other actions finish executing,
  - because objects and data become available,
  - or because events occur external to the flow.

# Activity diagrams

- Possibly more than any other UML diagram, activity diagrams apply to much more than just software modeling. They are applicable to just about any type of behavioral modeling; for example, business processes, software processes, or workflows.
- Software modeling, activities typically represent a behavior invoked as a result of a method call.
- Business modeling, activities may be triggered by external events, such as an order being placed, or internal events, such as a timer to trigger the payroll process on Friday afternoons.
- Activity diagrams have undergone significant changes with UML 2.0; they have been promoted to first-class elements and no longer borrow elements from state diagrams.
- Activity diagrams capture *activities that are made up of smaller actions .*

# Actions & Activities

- An action execution corresponds to the execution of a particular action.
- Similarly, an activity execution is the execution of an activity, ultimately including the executions of actions within it.
- Each action in an activity may execute zero, one, or more times for each activity execution.
- At the minimum, actions need access to data, they need to transform and test data, and actions may require sequencing.
- The activities specification (at the higher compliance levels) allows for several (logical) threads of control executing at once and synchronization mechanisms to ensure that activities execute in a specified order.

# Activity - Meta model



## Flows - Meta model



### Actions

- An action is a named element that is the fundamental unit of executable functionality. The execution of an action represents some transformation or processing in the modeled system, be it a computer system or otherwise.
- The action begins execution by taking tokens from its incoming control edges and input pins. When the execution of an action is complete, it offers tokens in its outgoing control edges and output pins, where they are accessible to other actions.

## Actions (2)



# Activity Modeling (1)

- Activity modeling emphasizes the input/output dependencies, sequencing, and conditions for coordinating other behaviors.
- Uses secondary constructs to show which classifiers are responsible for those behaviors.
- Focus is on what tasks need to be done, with what inputs, in what order, rather than who/which performs each task.

# Activity Modeling (2)

Tasks and ordering ...



# Activity Modeling (3)

... plus resource assignments



### "Flow" semantics



#### Activity execution defined in terms of flow of control and objects/data.

## Actions & Object Nodes (1)

Accept inputs, start behaviors, provide outputs



# Actions & Object Nodes (2)

Alternate object node notation.

Must use this notation if the output type is different than the input type.





- Tokens can
  - Stack up in "in/out" boxes
  - Prevent upstream behaviors from taking new inputs
- Applicable to systems with significant resource constraints, such as physical or manual processes



- Tokens can be
  - Stored temporarily
  - Divided between flows
- Tokens cannot
  - Flow in more than one direction, unless copied

## Activity Parameter Nodes



Parameter nodes accept and provide values to/from whatever behavior uses this activity

## Streaming Parameters

#### Streaming: values accepted and provided while action is executing



In this activity diagram, the original analog audio is streamed through a digital-toanalog converter, and the resulting digital data is sent to the encoding algorithm for processing.



**Charles ANDRE - UNSA** 

## Control Nodes (1)

#### Route objects/data

#### At beginning and end of activity:



Gets control when containing activity starts. Flow out immediately.

Accepts inputs, aborts containing activity

Accepts inputs, does nothing

### Control Nodes (2)

#### Route objects/data

In middle of activity:



Interruptible Region

#### Flows abort region



## Extensions for Systems Engineering

- Available in SysML (a UML profile for SE)
- Control as data
  - Enabling and disabling control values
  - Output from activities to turn other behaviors "on" and "off".
- Rate of flow, on edges and streaming parameters
- Reduce buffering
  - Overwrite values already in buffer
  - Turn off buffering
- Probability on decisions, parameter sets, competing outflows from object nodes
- Behavior decomposition

## Rate and buffer reduction



## Activity decomposition



Charles ANDRE - UNSA

### References

- Conrad Bock, UML 2 Activity articles: <u>http://www.conradbock.org/#UML2.0</u>
- UML 2 specification: <u>http://www.omg.org/docs/formal/07-02-</u> 03.pdf
- SysML specification: <u>http://www.omg.org/docs/ad/05-11-01.pdf</u>