Object Oriented Software Engineering (SE 313)
Lab Session # 04

Objective: Working with the Activity Diagrams with star UML

THEOR

Modeling with Activity Diagram

The following elements are available in an activity diagram.

- Action State
- Subactivity State
- Initial State
- Final State
- Synchronization
- Decision
- Flow Final
- Object Flow
- Signal Accept State
- Signal Send State
- Transition
- Self Transition
- Swimlane

Action State

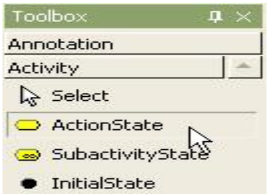
Semantics

An action state represents the execution of an atomic action, typically the invocation of an operation. An action state is a simple state with an entry action whose only exit transition is triggered by the implicit event of completing the execution of the entry action. The state therefore corresponds to the execution of the entry action itself and the outgoing transition is activated as soon as the action has completed its execution.

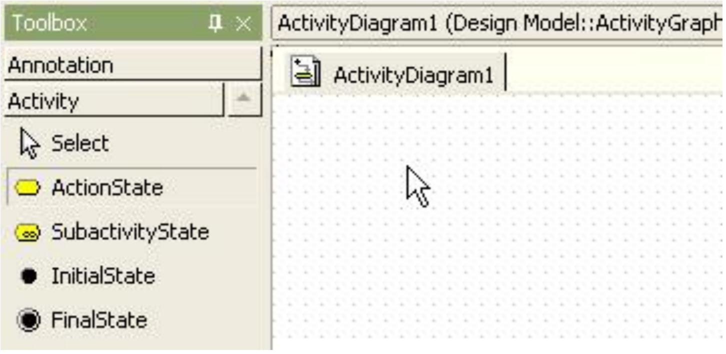
Procedure for creating action state

In order to create Action State,

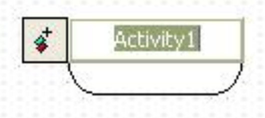
1. Click **[Toolbox] -> [Activity] -> [Action State]** button.



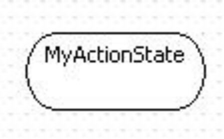
2. And click at the position where Action State will be placed in the **[main window]**.



3. A action state is created on the diagram and the quick dialog is shown.



4. Enter the action state name at the quick dialog and press **[Enter]** key. The result is as follows.



Subactivity State

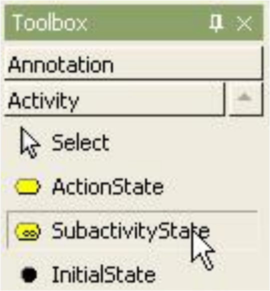
Semantics

A subactivity state represents the execution of a non-atomic sequence of steps that has some duration; that is, internally it consists of a set of actions and possibly waiting for events. That is, a subactivity state is a "hierarchical action," where an associated subactivity graph is executed.

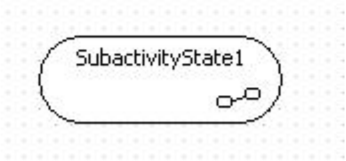
Procedure for creating subactivity state

In order to create SubactivityState,

1. Click **[Toolbox] -> [Activity] -> [Subactivity State]** button.



2. And click at the position where SubactivityState will be placed in the **[main window]**. A subactivity state is created and the quick dialog is shown. At the quick dialog, enter the subactivity state name and press**[Enter]** key. The result is as follows.

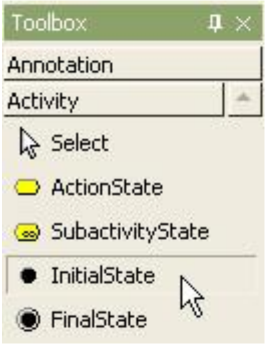


Initial State

Procedure for creating initial state

In order to create Initial State,

1. Click **[Toolbox] -> [Activity] -> [Initial State]** button.



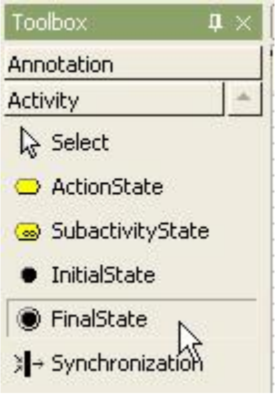
2. And click at the position where InitialState will be placed in the **[main window]**. Then a initial state is created.

Final State

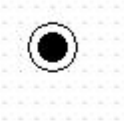
Procedure for creating final state

In order to create Final State,

1. Click **[Toolbox] -> [Activity] -> [Final State]** button.



2. And click at the position where Final State will be placed in the **[main window]**.



Decision

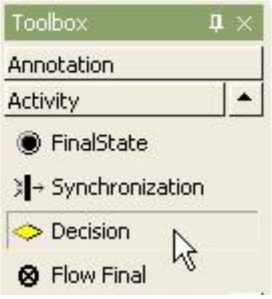
Semantics

A state diagram (and by derivation an activity diagram) expresses a decision when guard conditions are used to indicate different possible transitions that depend on Boolean conditions of the owning object.

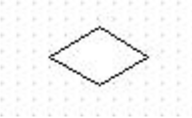
Procedure for creating decision

In order to create Decision,

1. Click **[Toolbox] -> [Activity] -> [Decision]** button.



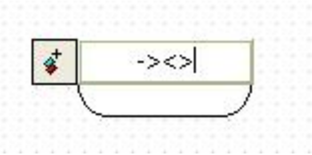
2. And click at the position where Decision will be placed in the **[main window]**. The decision is created on the diagram.



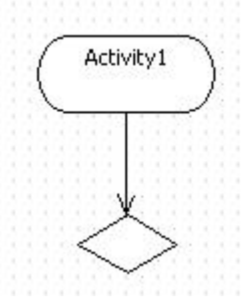
Procedure for creating decision from state

In order to create decision with incoming transition from selected object, use shortcut creation syntax.

1. Double-click state. At the quick dialog, enter "-><>"("<-<>" for incoming from decision) string.



2. Press **[Enter]** key and decision with outgoing transition from selected state is created.

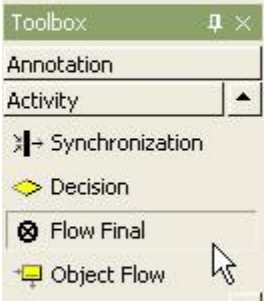


Flow Final

Procedure for creating flow final

In order to create Flow Final,

1. Click **[Toolbox] -> [Activity] -> [Flow Final]** button.



2. And click at the position where Flow Final will be placed in the **[main window]**.

Object Flow

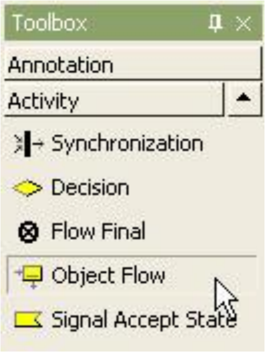
Semantics

An object flow is one of two types of activity edges, which are directed connection (flows) between activity nodes, the other being a control flow. As soon as the activity node at the source (tail) end of the flow is finished it presents tokens to the object flow at the target (arrowhead) end of the flow. An object flow can only carry object (data) tokens; it cannot carry control tokens. There are rules that specify whether tokens can flow along the object flow and these are determined by the type of activity node at the source and target of the flow. In the case of complete activities an object flow may define a weight, which specifies the minimum number of tokens that must flow along the object flow as a group.

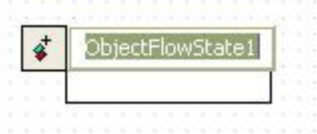
Procedure for creating object flow

In order to create Object Flow,

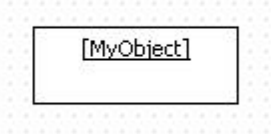
1. Click **[Toolbox] -> [Activity] -> [Object Flow]** button.



2. And click at the position where Object Flow will be placed in the **[main window]**. Then the quick dialog of object flow state is shown as follows.



3. At the quick dialog, enter the object flow state name and press **[Enter]** key.

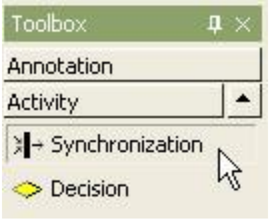


Synchronization

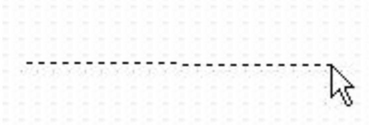
Procedure for creating synchronization bar

In order to create Synchronization,

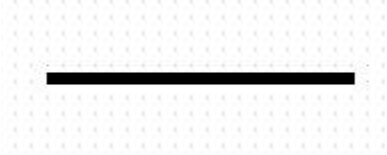
1. Click **[Toolbox] -> [Activity] -> [Synchronization]** button.



2. And click at the position where Synchronization will be placed in the **[main window]** and drag as size as you want.



3. The following figure shows the result of this procedure.



Signal Accept State

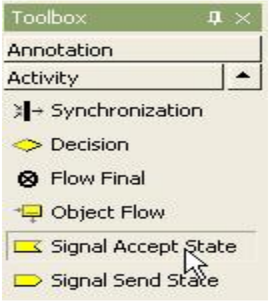
Semantics

The signal accept may be shown as a concave pentagon that looks like a rectangle with a triangular notch in its side (either side). The signature of the signal is shown inside the symbol. An unlabeled transition arrow is drawn from the previous action state to the pentagon and another unlabeled transition arrow is drawn from the pentagon to the next action state. A dashed arrow may be drawn from an object symbol to the notch on the pentagon to show the sender of the signal; this is optional.

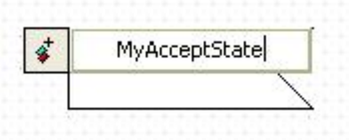
Procedure for creating signal accept state

In order to create Signal Accept State,

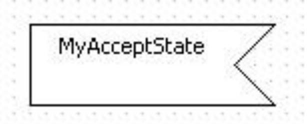
1. Click **[Toolbox] -> [Activity] -> [Signal Accept State]** button.



2. And click at the position where Signal Accept State will be placed in the **[main window]**.



3. At the quick dialog, enter signal accept state name and press **[Enter]** key.



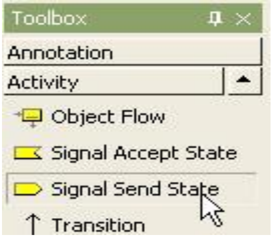
Signal Send State

The sending of a signal may be shown as a convex pentagon that looks like a rectangle with a triangular point on one side (either side). The signature of the signal is shown inside the symbol. An unlabeled transition arrow is drawn from the previous action state to the pentagon and another unlabeled transition arrow is drawn from the pentagon to the next action state. A dashed arrow may be drawn from the point on the pentagon to an object symbol to show the receiver of the signal, this is optional.

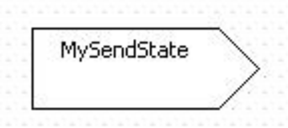
Procedure for creating signal send state

In order to create Signal Send State,

1. Click **[Toolbox] -> [Activity] -> [Signal Send State]** button.



2. And click at the position where Signal Send State will be placed in the [main window]. A signal send state is created and the quick dialog is shown. Enter signal send state name and press **[Enter]** key.

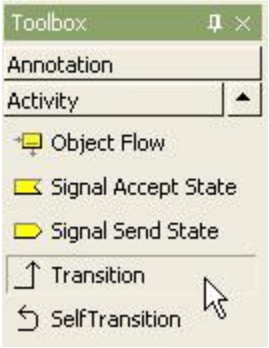


Transition

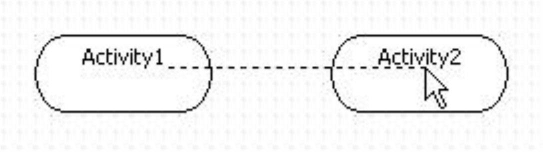
Procedure for creating transition

In order to create Transition,

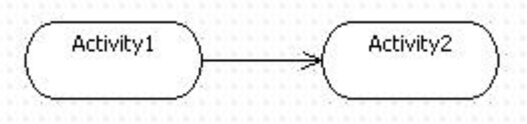
1. Click **[Toolbox] -> [Activity] -> [Transition]** button.



2. Drag and drop between states in transition direction in the **[main window]**.



3. Then the transition is created.

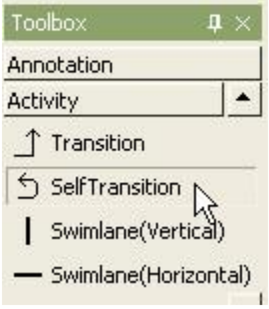


SelfTransition

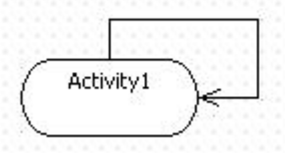
Procedure for creating self-transition

In order to create self-transition,

1. Click **[Toolbox] -> [Activity] -> [SelfTransition]** button.



2. Click state to have self-transition in the **[main window]**. Then a self-transition is created.



Swimlane

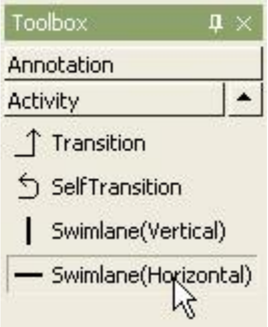
Semantics

Actions and subactivities may be organized into swimlanes. Swimlanes are used to organize responsibility for actions and subactivities. They often correspond to organizational units in a business model.

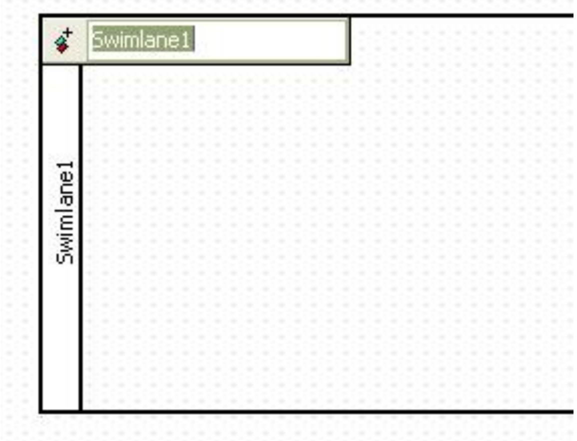
Procedure for creating horizontal swimlane

In order to create Horizontal Swimlane,

1. Click **[Toolbox] -> [Activity] -> [Horizontal Swimlane]** button.



2. And drag the boundary where Horizontal Swimlane will be placed in the **[main window]**.

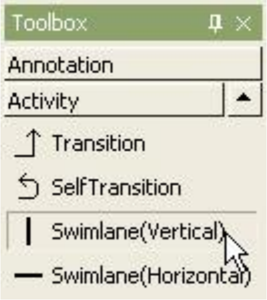


3. Then a horizontal swimlane is created on the diagram. And enter the swimlane name at the quick dialog and press **[Enter]** key.

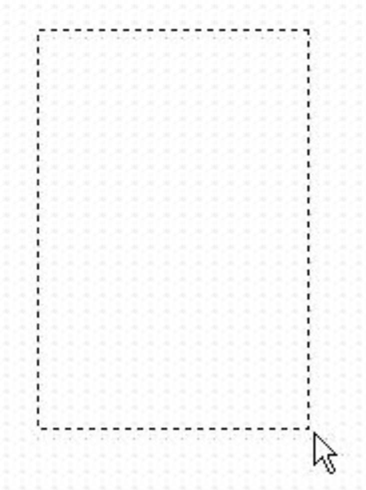
Procedure for creating vertical swimlane

In order to create Vertical Swimlane,

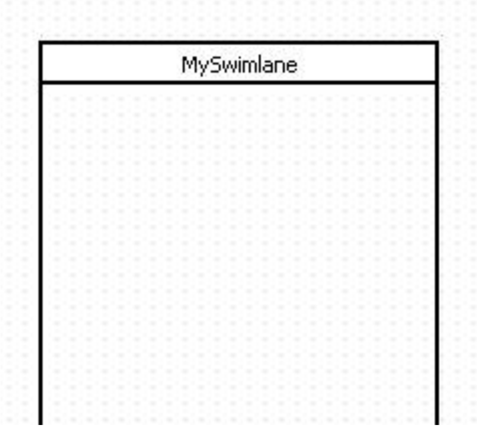
1. Click **[Toolbox] -> [Activity] -> [Vertical Swimlane]** button.



2. And drag the boundary where Vertical Swimlane will be placed in the **[main window]**.



3. A swimlane is created and quick dialog is shown. At the quick dialog, enter the swimlane name and press **[Enter]** to have done this procedure.



EXERCISES

1. Draw activity diagram for Online Shopping with standard steps.

   (activities would be Login, Search, Browse, Add to cart, edit cart, delete cart, Confirm purchase, make payments, receive item, rate website and logout.

2. Draw activity diagram for order management system. In the diagram four activities are identified which are associated with conditions. The following diagram is drawn with the four main activities:

   - Send order by the customer
   - Receipt of the order
   - Confirm order
   - Dispatch order

After receiving the order request condition checks are performed to check if it is normal or special order. After the type of order is identified dispatch activity is performed and that is marked as the termination of the process.

3. Re-draw order management system with the implementation of swimlane.

4. Draw the activity diagram of ATM system with the implementation of swimlane

Students Task

1. Draw activity diagram of online railway reservation

The two actors would be a customer and a ticket counter having following activities.

- Search train
- Check availability of train (if yes go to book ticket, if no go to quit)
- Quit
- Book ticket
- Fill details
- Submit details
- Make payment
- Print Ticket
- Cancel ticket
- Get refund

2. Draw activity diagram of patient management system having two actors Receptionist and Doctor. (e.g Appointment Management System)