

Chapter 10: Activity Diagram

The Activity Diagram

What is an Activity Diagram

- Describes activities and flows of data or decisions between activities
- Provides a very broad view of business processes
- Can be used to break out the activities that occur within a use case
- Commonly shows many different activities that will be handled by lots of different symbols
- Good for showing parallel threads

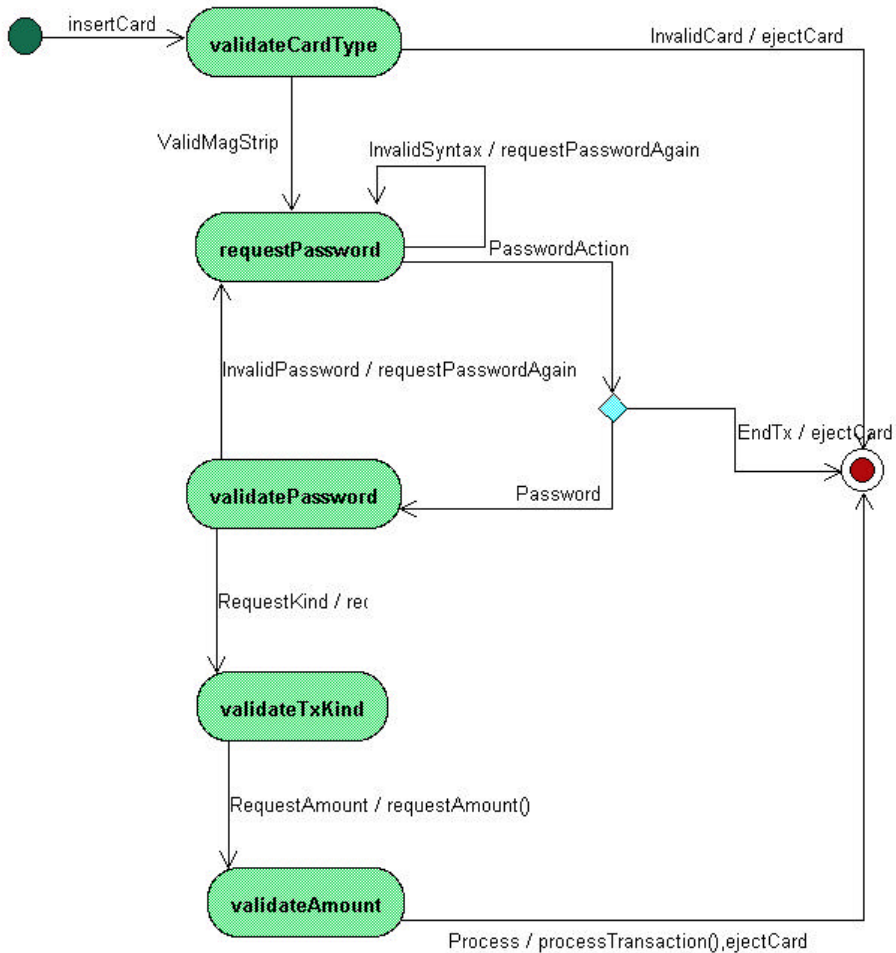
When to use an Activity Diagram


- When describing work flow across many use cases
- When analyzing a use case, and before methods are assigned to symbols
- When dealing with multi-threaded applications

Creating an Activity Diagram

This diagram is useful in showing work flow connections and describing behavior that has a lot of parallel processing. When you use an activity diagram you can choose the order in which to do things. It states the essential sequencing rules to follow. It is different from a flow chart in that it shows parallel processes, not just sequential processes.

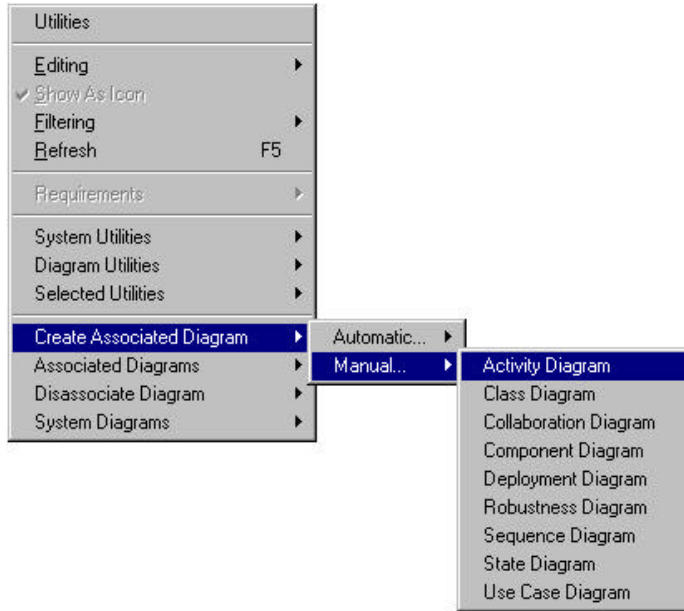
When you have completed your Activity Diagram portion of the tutorial your model should be similar to the following example.



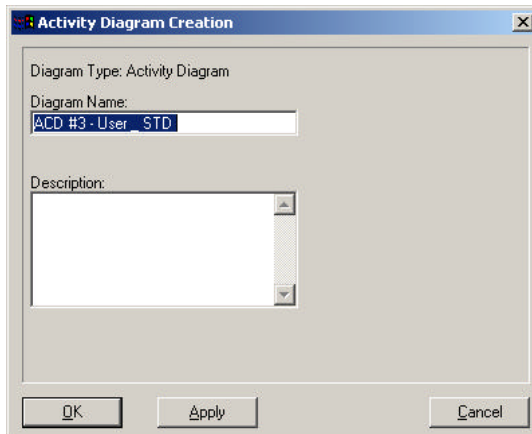
 **Note:** The diagram shown above is for reference only. Use the instructions beginning on the next page to draw your Activity diagram.


Drawing the Activity Diagram


1. Right-click in the background of the state diagram. The Utilities background menu opens.




2. Choose CREATE ASSOCIATED DIAGRAM->MANUAL->ACTIVITY DIAGRAM. The diagram type dialog box opens. The default name includes the type of diagram (Activity), and the diagram name (ACD #1 - User_ STD). You can edit this name but for purposes of the tutorial we will leave the default. If you want, you can also edit the Description of the diagram.



3. Click . The Diagram type dialog box closes and the Diagram Window opens with an Activity Diagram labeled "ACD_1_STD".

 **Note:** The diagram you just created is now the active design model. When the Activity model is created, the Diagram Window displays a palette with icon symbols used to create activity diagrams.







 **Note:** When the Activity Diagram is created, the previously created diagrams also remain open.

The Activity Diagram Palette

Each icon on this palette represents a notation used to create an activity diagram.



<u>Icon</u>	<u>Notation</u>	<u>Definition</u>
	Action State	<ul style="list-style-type: none"> ◆ In a conceptual diagram an activity is a task that needs to be done - either by a human or a computer. ◆ In a specification-perspective diagram or an implementation-perspective diagram, an activity is a method on a class
	State Transition	<p>In an activity diagram the states are activities representing the performance of operations. The transitions are triggered by the completion of the operations.</p> <p>A transition is a relationship between two states. This transition indicates that control is passed from the first state to the second state once the work of the source state is complete.</p> <p>To emphasize functional flow of control, transitions can be labeled, and contain parameters, guard conditions and action expressions.</p>
	Initial State	A pseudo state to establish the start of the event into an actual state.
	Final State	Signifies when a state transition ends.



Decision As in a flowchart, a conditional flow of control is modeled using the decision element. A decision is shown by labeling each output transition of a decision with a different guard condition.



SwimLane It is a "package" for organizing activities within a class. Each zone or lane represents the responsibilities of a particular class or department. Swimlanes depict both the activity diagram's logic along with the depiction of responsibility. To use swimlanes, you must arrange your activity diagrams into vertical zones separated by dashed lines. Transitions may cross swimlane boundaries.



Merge Branch Bar The merge branch bar symbol is also known as a "Synchronization Bar". It merges concurrent transitions to a single target. It splits a single transition into parallel transitions.



Binary Constraint The binary constraint notation is available on all diagram palettes. A constraint is a semantic relationship among model elements that specifies conditions and propositions that must be maintained as true. Otherwise the system described by the model is invalid. Certain kinds of constraints (such as association "or" constraint) are predefined in UML, others can be user defined. A constraint represents semantic information attached to a model element, not just a view of it.

A binary constraint allows a constraint to be defined between any symbols on the diagram. The binary constraint allows the constraint to be defined on the link rather than in a note symbol. If there is a need for a single constraint or three or more way constraint, then a note symbol is used to explain the constraint and the note symbol is linked to the constrained symbols using a note link.




Note Link The note link notation is available on all diagram palettes. The note pad can be used to record information for a object or link in a diagram. This information is not included in generated code but is for information only. Each note pad can contain unlimited text, can be numbered, a stereotype defined, and a noted element entered.

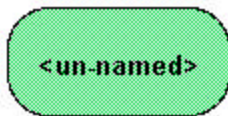


Note Pad The note pad notation is available on all diagram palettes. The note pad can be used to record information for an symbol or link in a diagram. This information is not included in generated code but is for information only. Each note pad can contain unlimited text, and be numbered. You can also define a stereotype, and enter a noted element.

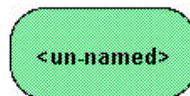
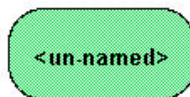
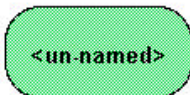
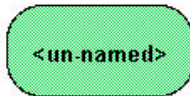
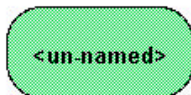
Placing Action States


The first part of creating the Activity diagram consists of placing the action states.

1. From the Activity Diagram palette double-click the ACD ActionState symbol icon .
2. Position the cursor in the top center portion of the drawing area and click. An action state symbol is placed on the design area.



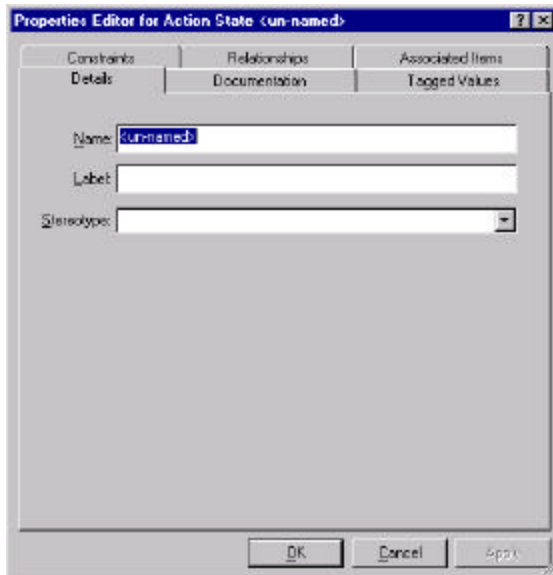
3. Place four more action state symbols in the diagram. When all symbols are in place the diagram should resemble the following graphic.




4. Deselect the action state icon by pressing the ESC key or by clicking the cursor  located below the Activity Diagram palette.

Identifying the Action States

1. Double-click the top action state symbol. The Properties Editor Action State dialog box opens.



2. Enter the text "validateCardType" in the Name text box and click . The Attribute Editor closes and the text appears in the action state symbol.



3. Click once on the second action state symbol to select it. Click once again and the State Identifier pop-up editor opens.



4. Enter the text "requestPassword" and press Enter. The window closes and the text is entered in the action state symbol.
5. Repeat steps 4-6 to label the remaining action state symbols as follows:
 - validatePassword
 - validateTxKind
 - validateAmount

validateCardType

requestPassword


validatePassword

validateTxKind


validateAmount

Using the Initial State in an Activity Diagram

The initial state is a pseudo state to establish the start of the event into an actual state.

1. Click the Initial State symbol in the Activity Diagram palette .
2. Place the symbol by clicking approximately one inch (1") to the left of the "validate card type" state symbol.



3. To connect the initial state symbol to the action state symbol, click the Transition symbol in the Activity Diagram palette .
4. Click once on the right side of the initial state symbol, drag the cursor to the left side of the action state symbol and click again. A transition link now connects the two symbols.




5. To label the transition click once on the link to select it. Click again and the Transition Identifier pop-up editor opens.

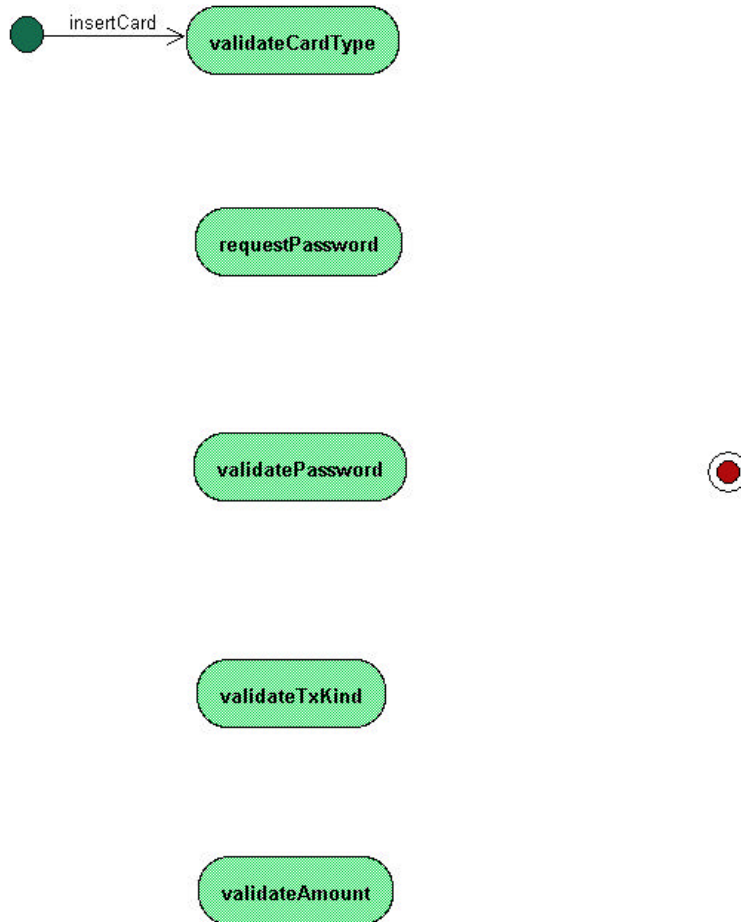
Transition Identifier	
EventName	GuardCondition
<input type="text"/>	<input type="text"/>

6. Enter the text "insertCard" in the EventName text box and press Enter. The pop-up editor closes and the link is labeled.

Using the Final State in an Activity Diagram


The final state symbol represents the completion of the activity.

1. Click the Final State symbol in the Activity Diagram palette .
2. The final state will complete activities from multiple actions so we will put the symbol to the right of all the action states. Place the symbol by clicking approximately two inches (2") to the right of the "validatePassword" action state symbol.



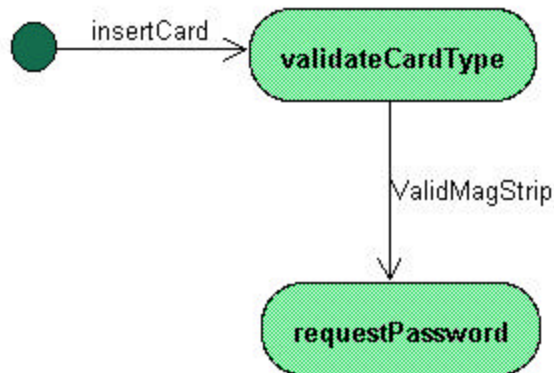
Transition Links in an Activity Diagram

Each element in the activity diagram is connected by a transition link.

1. To draw a transition link, click the Transition symbol in the Activity Diagram palette .
2. Click once on the bottom center of the action state symbol labeled "validateCardType". Drag the cursor down to the top center of the action state labeled "requestPassword" and click again. A transition link now connects the two symbols.
3. To label the transition click once on the link to select it. Click again and the Transition Identifier pop-up editor opens.

Transition Identifier	
EventName	GuardCondition
<input type="text"/>	<input type="text"/>


4. Enter the text "ValidMagStrip" in the EventName text box and press Enter. The pop-up editor closes and the link is labeled.

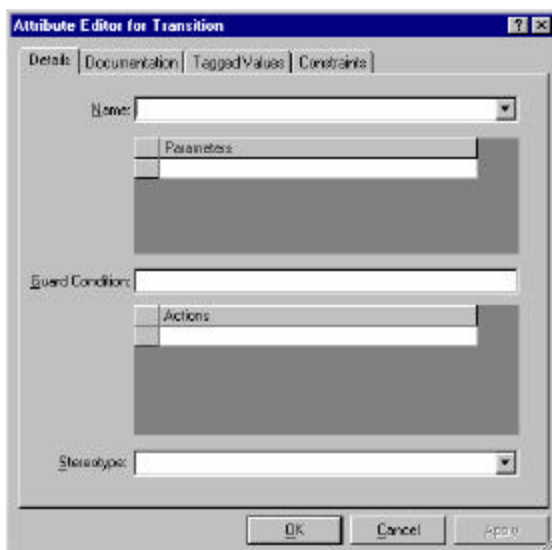


5. Since we are going to put another link on the right side of the diagram we need to reposition the label. Place the cursor on the label "ValidMagStrip" and while holding down the mouse button, move the label to the left side of the link. Release the mouse button when the label is in place.

Additional Transition Links


The next link we will draw is between the "validateCardType" symbol and the Final state symbol.

1. Select the Transition Link icon from the palette .
2. Click the right side of the "validateCardType" symbol. Drag the cursor down to the Final State symbol. The link is squared automatically.
3. Following the steps 1 and 2, draw the next link from the "validateAmount" symbol to the Final State symbol.
4. To label the transition between "validateCardType" and the final state, double-click the link. The Properties Editor for Transition dialog box opens.




5. Enter the text "InvalidCard" in the Name text box.

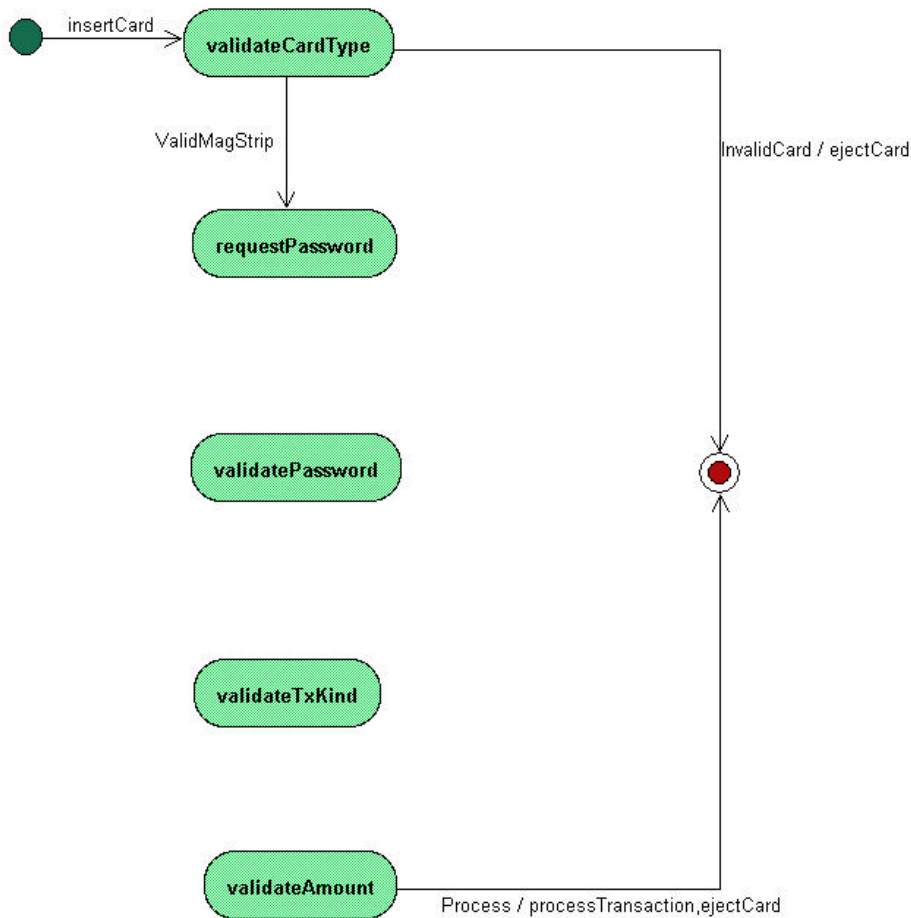
6. Double-click the list box below Action. A text box is activated.

7. Enter the text "ejectCard" and click . The Properties Editor dialog box closes and the transition link is now labeled.

8. Following steps 5-7, label the second link with the following text:


Text Box Label	Text
Event Name	= Process
Action	= processTransaction()
Action	= ejectCard

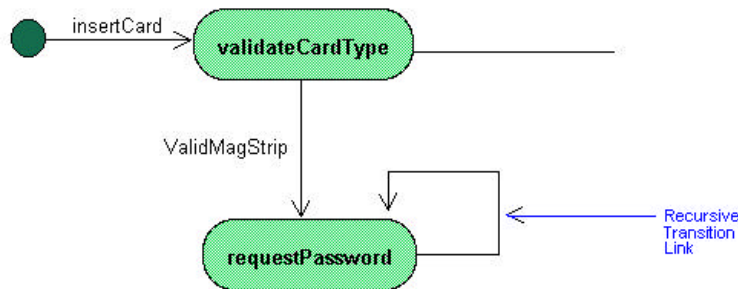
 **Note:** After entering the first action, double-click in the next text box and enter the second action in the activated text box.



The Recursive Transition Link

One of the transitions is a repeat activity so it transitions back to the same action state it started from.


1. Select the Transition Link icon from the palette .
2. While holding down the Ctrl key, click the right side of the "request password" symbol. Drag the cursor to the right in a straight line approximately one-half inch (1/2").
3. Click and drag the cursor up approximately one-half inch (1/2").
4. Click and drag the cursor to the left approximately one-half inch (1/2").
5. Click, release the Ctrl key and drag the cursor down to the top of the Action State.
6. Click once again to end the link.

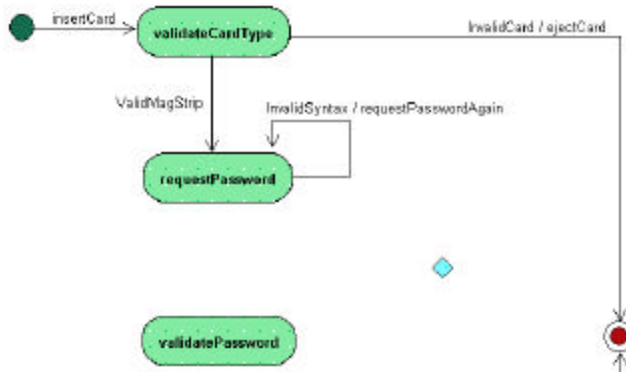


7. Double-click the link to open the Properties Editor for Transition dialog box. Following the procedure used to label previous links, enter the text "InvalidSyntax" for the Event Name and "requestPasswordAgain" for the Action.

Using the Decision Symbol

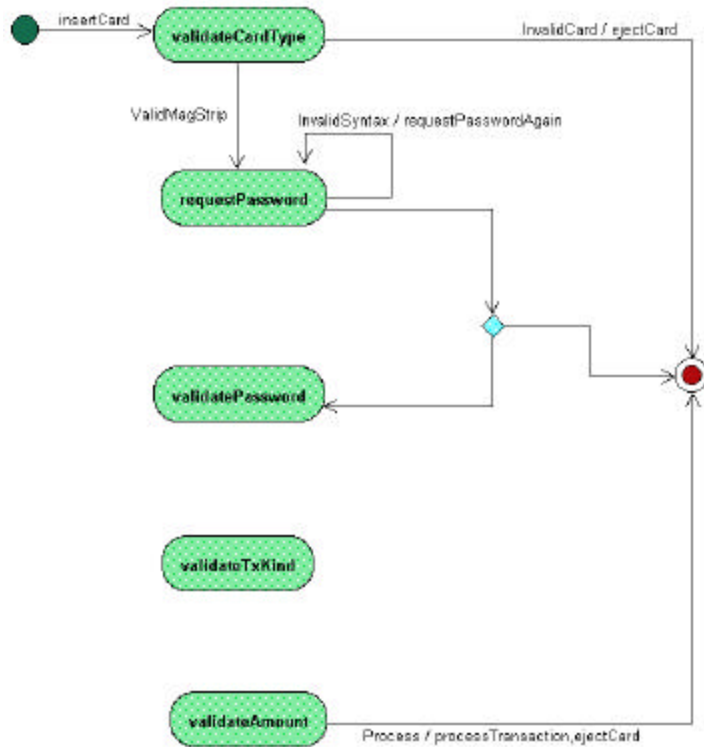
An activity diagram shows a decision to indicate different possible transitions depending on the conditions of the owning symbol. This diagram uses a decision to show two possible resolutions of the request password action.

1. Click the Decision symbol on the Activity Diagram palette .
2. Place the symbol to the right and between the "request password" and "validate password" action state symbol.



3. Draw the following transition links:

- request password to decision symbol (◆)
- decision symbol (◆) to validate password
- decision symbol (◆) to final state symbol (●)



4. Label the links as follows:

<u>Link</u>	<u>Event Name</u>	<u>Action</u>
request password to decision	PasswordAction	
decision to validate password	Password	
decision to final state	EndTx	ejectCard

The Remaining Transition Links

To complete the activity diagram we need to draw three more transition links.

1. Select the Transition Link icon from the palette .
2. Click on the upper left corner of the "validatePassword" action state. Drag to the lower left corner of the "requestPassword" action state and click again. A transition link is drawn between the two states.
3. Repeat steps 1 and 2 to draw the following links:

validatePassword	to	validateTxKind
validateTxKind	to	validateAmount

4. Label the links as follows:

<u>Action State</u>	<u>Event Name</u>	<u>Action</u>
validatePassword to requestPassword	InvalidPassword	requestPasswordAg ain
validatePassword to validateTxKind	RequestKind	requestKind()
validateTxKind to validateAmount	RequestAmount	requestAmount()

Your completed Activity diagram should resemble the following illustration:

