

Chapter 4: Use Case Diagram

The Use Case Diagram

What is a Use Case Diagram

- Describes the behavior of a system from a user's standpoint
- Functional description of a system and its major processes
- Provides a graphic description of who will use a system and what kinds of interactions to expect within that system
- Processes that occur within the application area are called use cases
- Entities outside the area that are going to use the application are called actors

When to use a Use Case Diagram

When describing requirements for a system in the analysis, design, implementation and documentation stages.

When Constructing Use Cases Think About...

- What tasks the actor must perform
- What resources the actor requires in terms of data read/written
- Keeping the use cases simple

Creating the System's Use Case Diagram

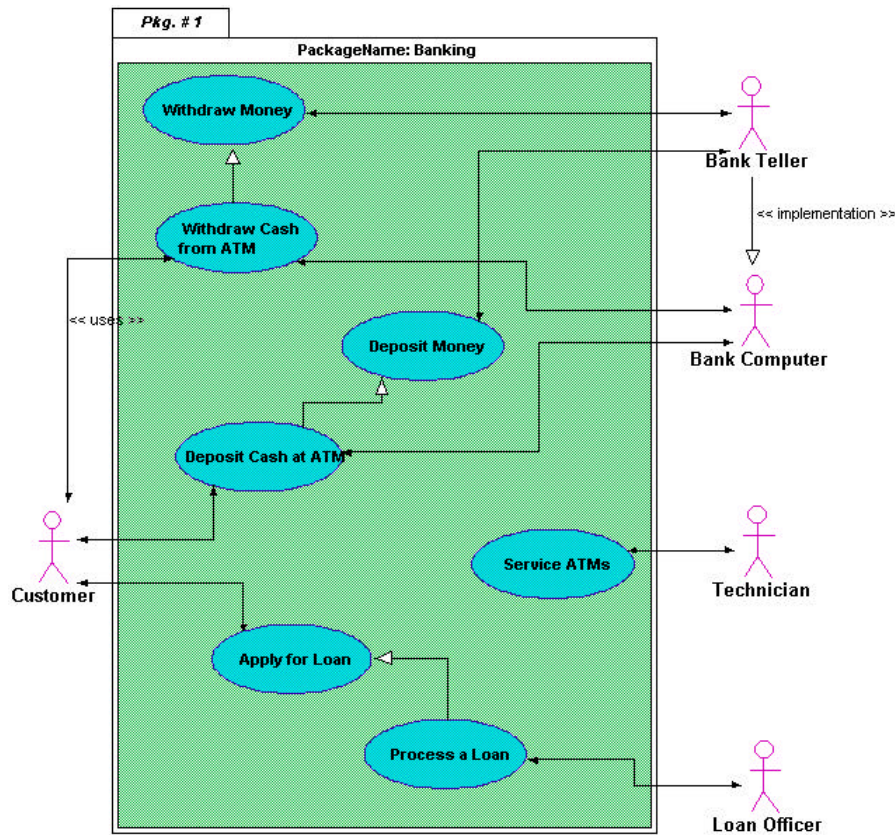
Using the Use Case Diagram model, you show the relationship among actors and use cases within a system. The Use Case diagram we will be creating tracks various functions and those who interact with the functions within a banking system.

Drawing Use Case symbols

Use Case symbols define instances. When representing an instance, you:

- ◆ Draw use case symbols
- ◆ Name the symbols
- ◆ Draw the actor symbols
- ◆ Label the actor symbols
- ◆ Link use case and actor symbols with communications and extends

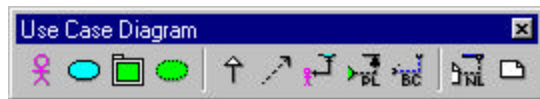
After you have completed the step-by-step procedure outlined in the tutorial, your use case diagram should look similar to the following example.



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

The Use Case Diagram Palette

Each icon on this palette represents a notation used to create a use case diagram.



Icon	Notation	Definition
	Actor	<ul style="list-style-type: none"> An actor represents a coherent set of roles that users of a system play when interacting with the use cases of the system. Actors can be anything - humans, devices, other systems One physical object can play several roles and so can be modeled by several actors.



Use Case

Use cases describe what a system does, not how it does it. A use case contains multiple scenarios, each of which describes a specific flow of events through the use case. Use case behavior is specified by describing the scenarios clearly enough for outsiders to understand.



Package

The Package symbol is used in the Use Case, Class, and Component diagrams. A package is a grouping of model elements. Packages themselves may be nested within other packages. A package may contain both subordinate packages and ordinary model elements. Some packages may be Subsystems of Models. The entire system description can be thought of as a single high-level subsystem package with everything else in it.



Design Pattern

Patterns describe small, recurring entities, reusable on a day to day basis in order to resolve specific problems. These patterns do not express the general form of an application.

This symbol is available in the following diagrams: Use Case, Class, Collaboration, Deployment, and Component.



Generalization Link

Generalization is the taxonomic relationship between a more general element and a more specific element that is fully consistent with the first element and that adds additional information. This link is also known as a specialization or inheritance link.

Note: In previous version there was a Use Case Link and an Actor Link. The Generalization link takes the place of both of this icons.



Dependency Link

The dependency link is a semantic relation between the source and target elements. It indicates that when there is a change to the target element there may be a change necessary to the source element. You can label the dependency link and set the stereotype.



Comm Link

The link is used to associate an actor to a use case. It shows participation of an actor in a use case and is the only relationship between actors and use cases.



Pattern Member Link

Used to show the member classes or objects participating in a pattern.



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.


Defining the Use Cases

The first step is to define the use cases within the system.


1. Click  on the Use Case Diagram palette to select it.
2. Place the cursor in the upper middle portion of the design area and click once. A Use Case symbol is placed in the design area.
3. To change the name of a symbol or link, all you have to do is select the symbol or link and start typing. A text box automatically opens. Enter the label "Withdraw Money" and click anywhere outside the text box. The unnamed label is replaced with the new text.

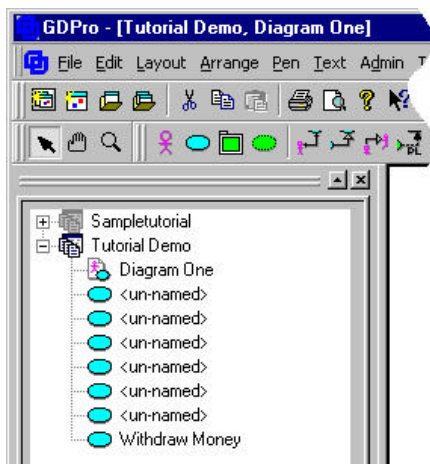



4. Double-click the Use Case icon in the Use Case Diagram Palette . The cursor changes  indicating it is in the multiple placement mode. By double-clicking on the Use Case icon, you can draw multiple Use Case symbols in the design area.

 **Note:** You can also click the icon once and then while holding down the space bar, place multiple symbols in the diagram. Release the space bar before placing the last symbol.

5. Place six more use cases in the design area as illustrated.

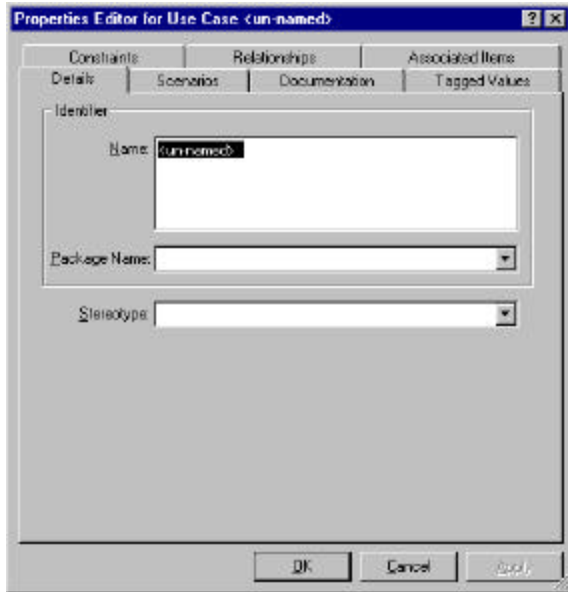
 **Note:** As you place the symbols in the design area, notice that a representative symbol is placed in the System Hierarchy Window under the system name. The list of symbols is organized alphabetically according to the symbol label.




6. Deselect the Use Case symbol icon by clicking the cursor icon  located above the Use Case Diagram palette or press the ESC key.

Labeling the Use Case Symbol

1. Click once to select the use case that is to become the Withdraw Cash from ATM. Handles appear around the class symbol.
2. Double-click the use case and the Properties Editor dialog box opens. The cursor is active in the Name text box and the label <unnamed> is highlighted.

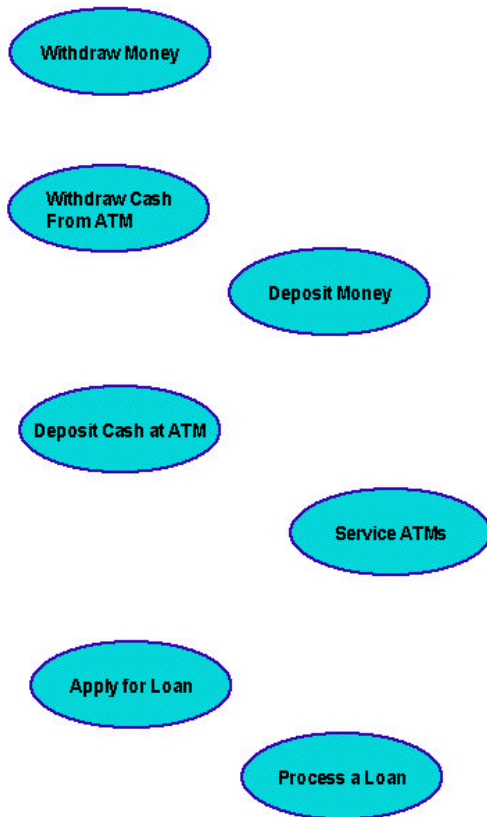


3. Enter the text "Withdraw Cash" and press Enter. The cursor advances to the next line.
4. Enter the text "from ATM" and click . The text "Withdraw Cash from ATM" is entered in the use case symbol and the dialog box closes.

Label Remaining Use Cases


We will use the direct entry method to label the remaining class symbols.

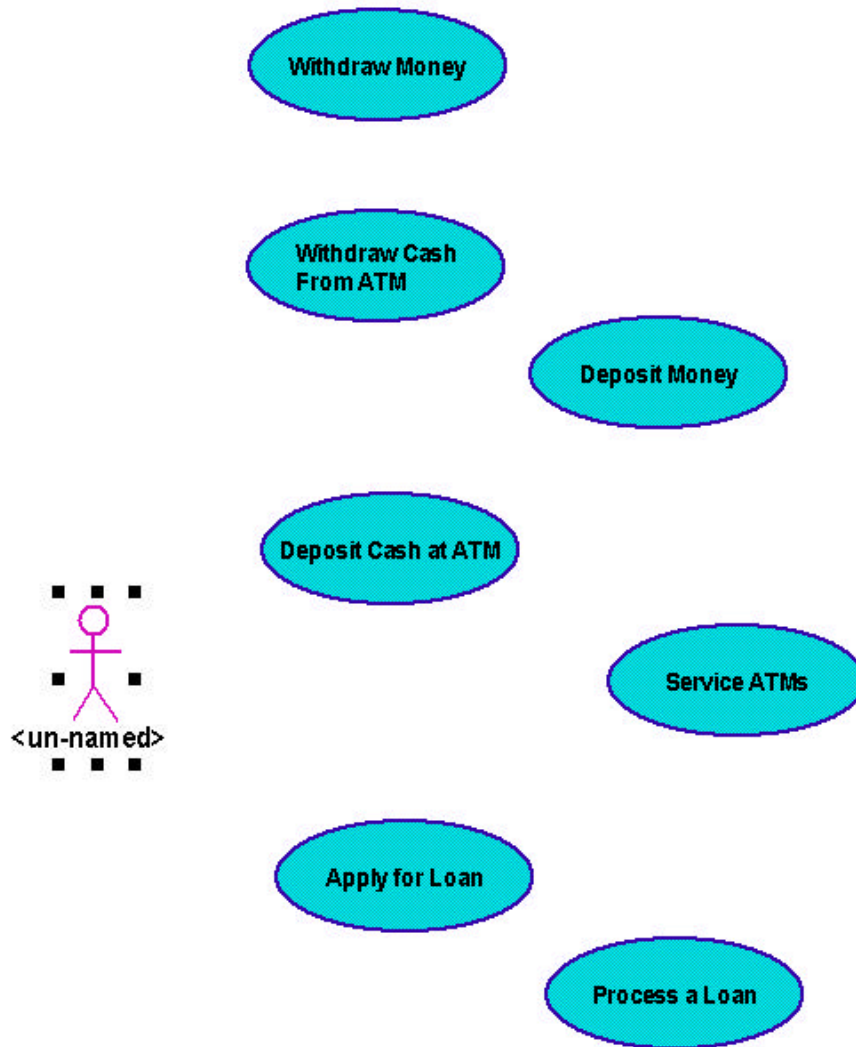
1. Click once to select the symbol to be labeled "Deposit Money".
2. Enter the text "Deposit Money" and click anywhere outside the text box. The symbol is labeled.
3. Repeat steps 1 and 2 to label the remaining use case symbols.



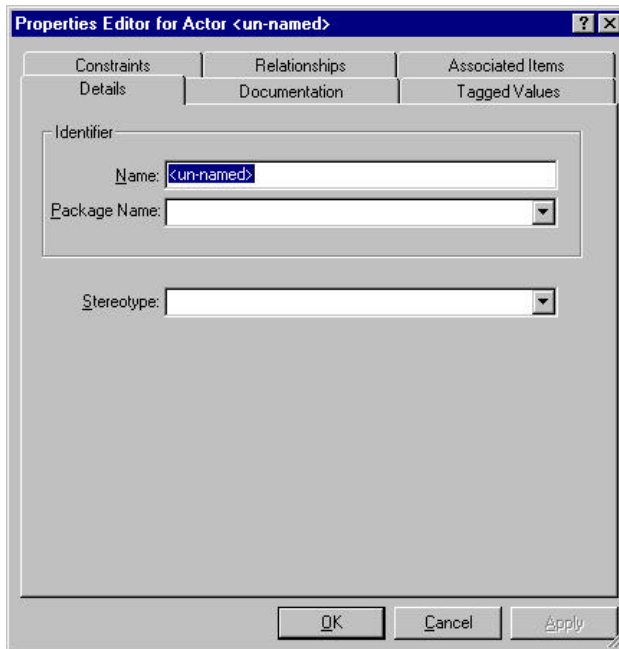
Add Actors


The next step in creating a Use Case diagram is to place and define the actors. An actor is a predefined stereotype of type showing an entity outside the package that interacts with use case symbols.

1. Select the Actor icon by clicking once on the  in the Use Case palette.
2. Put the actor in the diagram by placing the cursor to the left of the Use Case symbols and clicking once. An unnamed actor is placed in the design area.





3. Double-click the actor graphic and the Properties Editor for Actor dialog box opens.

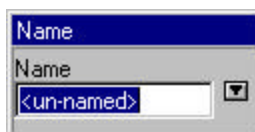


4. Enter the text "Customer" in the Name text box and click . The dialog box closes and the actor is labeled.



Continue Adding Actors


1. Double-click the Actor icon in the Use Case Diagram Palette  to place multiple symbols. The icon is grayed out.
2. Place four more actors in the design area as illustrated.
3. Deselect the Actor icon by clicking the cursor icon  located by the Use Case Diagram palette or press the ESC key.
4. Click once on the actor symbol that will be labeled Bank Teller. The actor is selected.
5. Click the <unnamed> label under the selected actor and the Name pop-up editor opens.




6. Enter the text "Bank Teller" in the text box and press Enter. The pop-up editor closes and the actor symbol is labeled.
7. Repeat steps 4 through 6 to label the remaining actors Bank Computer, Technician and Loan Officer.

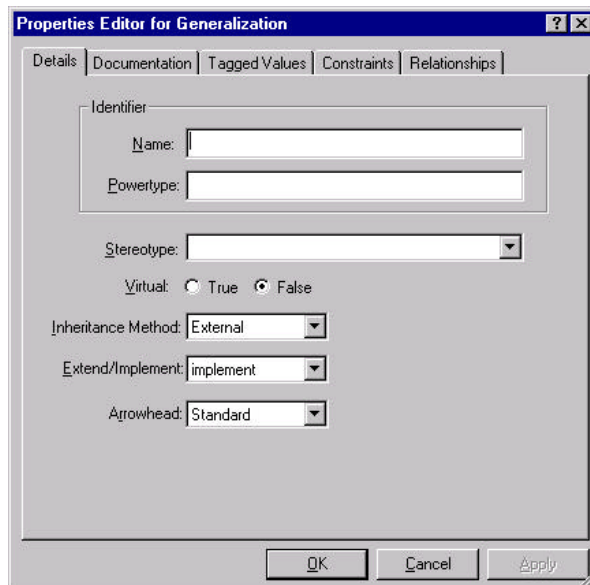
Linking Actors to Actors


The actor link shows the communication or connection between an actor and another actor. Because ORTHOGONAL LINKS is the default setting, all links drawn are automatically squared.

1. Click the Generalization Link icon in the Use Case Diagram palette .
2. Click inside the actor symbol labeled Bank Teller, drag the cursor down to the actor labeled Bank Computer and click again. A valid link snaps in place between the two actors.

 **Key:** You can define the proximity snap sensitivity of establishing link relationships. When you draw a link to a symbol, the link is automatically connected when it is dropped within a user-defined distance from the target symbol.


3. Double-click the generalization link and the Properties Editor for Generalization Link dialog box opens.




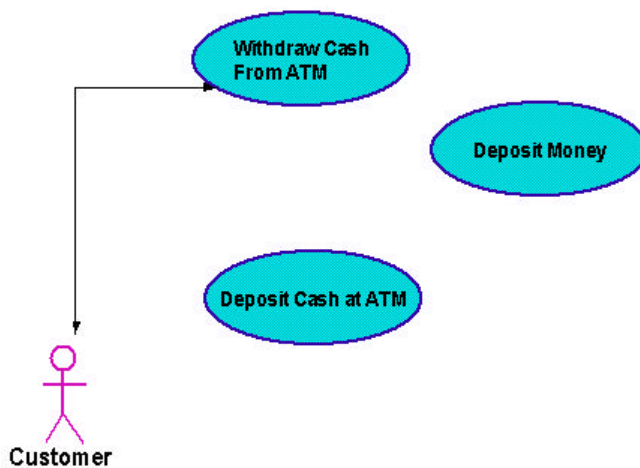
4. Select "implementation" from the Stereotype drop down list and click . The dialog box closes and the link is labeled "<<implementation>>".

Linking Actors to Use Cases

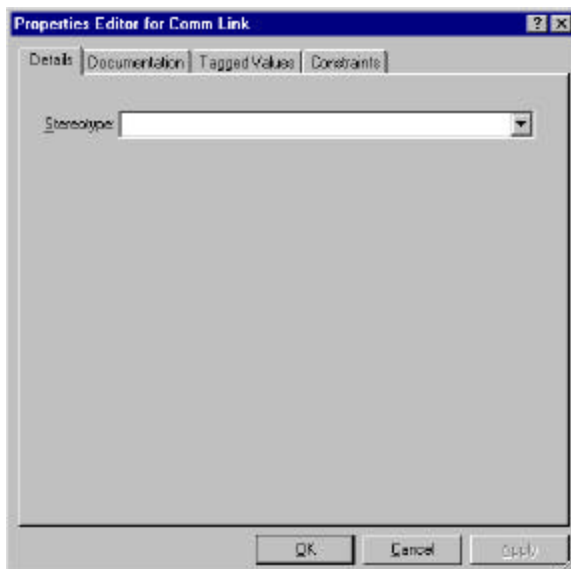
A link shows the communication or connection between an actor and a use case class.

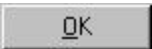
1. Click the CommLink icon in the Use Case Diagram palette .
2. Click inside the actor symbol labeled Customer, drag the cursor inside the "Withdraw Cash From ATM" use case symbol and click again. A link snaps in place from the actor to the use case.

 **Note:** You can draw the link directly from the "Customer" actor to the "Withdraw Cash From ATM" use case. The link is automatically squared as shown in the following graphic.




3. Label the link by double-clicking the commlink symbol between Customer and Withdraw Cash from ATM. The Properties Editor for Comm Link dialog box opens.



4. Enter the text "uses" and click . The dialog box closes and the commlink is labeled.

Draw Remaining Links

1. Click the CommLink icon in the Use Case palette .
2. While holding down the space bar, connect the following actors and use cases by clicking first in the actor symbol, dragging the cursor to the use case symbol and then clicking once again.


ACTOR	USE CASE
Customer	Deposit Cash at ATM
Customer	Apply for Loan
Bank Teller	Withdraw Money
Bank Teller	Deposit Money
Bank Computer	Withdraw Cash From ATM
Bank Computer	Deposit Cash at ATM
Technician	Service ATM's

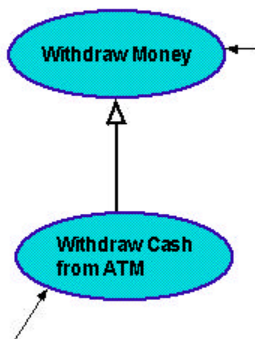
3. Release the space bar and draw the last link.

Loan Officer	Process a Loan
--------------	----------------

Using the Extended Links

An extended link shows a relationship from one use case to another, specifying how the behavior defined for the first use case can be inserted into the behavior defined for the second use case.

1. Click the Generalization Link icon in the Use Case diagram palette .
2. Click once in the "Withdraw Cash From ATM" symbol, drag the cursor to the "Withdraw Money" symbol and click again. A Use Case link is drawn with the arrow pointing towards the Withdraw Money symbol.





3. Repeat steps 1 and 2 to draw extend links between the following use cases:

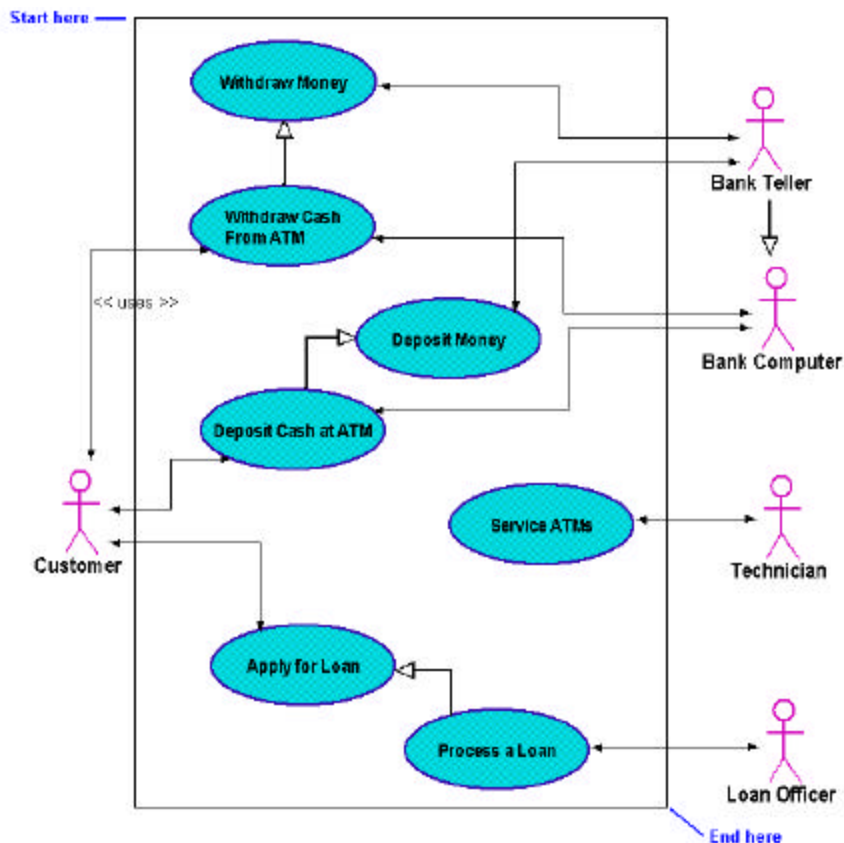
- ◆ Deposit Cash at ATM --> Deposit Money
- ◆ Process a Loan --> Apply for Loan


Adding the Package Symbol

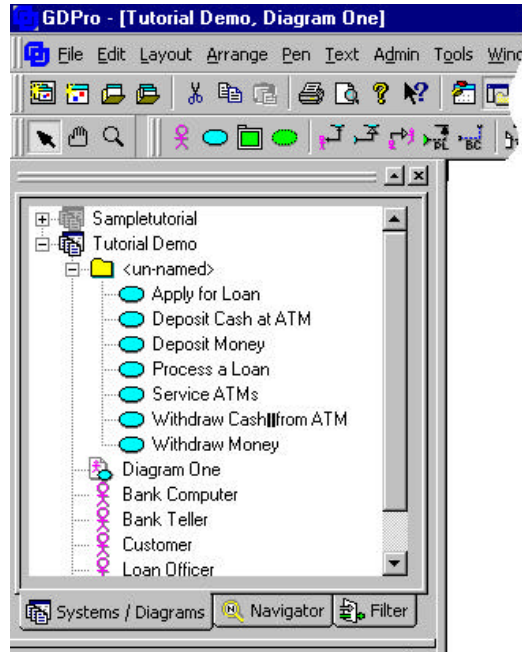
A package is a collection of connected units that are organized to accomplish a specific purpose. For this tutorial the package encompasses all the use cases needed for the specified "banking" purpose.

1. Click the Package icon  in the Use Case Diagram palette.
2. Starting in the upper left corner of the diagram, click and drag the cursor to the lower right corner and release the mouse. A package labeled <unnamed> is drawn around the use case symbols. Do not include the actors in the package.

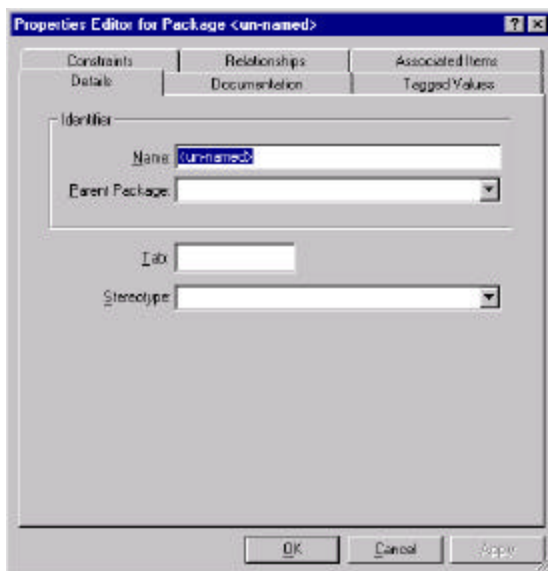
 **Key:** Once you place a package around the use case symbols, these symbols are now in a relationship with the package. When you select a use case symbol in the package, the symbol now has blue "handles" instead of black.




 **Note:** Notice that once the package is drawn, a package symbol icon appears in the System Hierarchy Window and all the uses cases are placed below that package icon.



- Double-click in the background of the package. A Properties Editor for Package dialog box opens.



- Enter the text "Banking" in the Name text box.
- Tab to advance to the Tab text box and enter the number "1". Click  in the properties dialog box and the package tab and label appear in the Package box.

The Finished Use Case Diagram

Once you have added the package and labeled it, your finished diagram should resemble the following:

