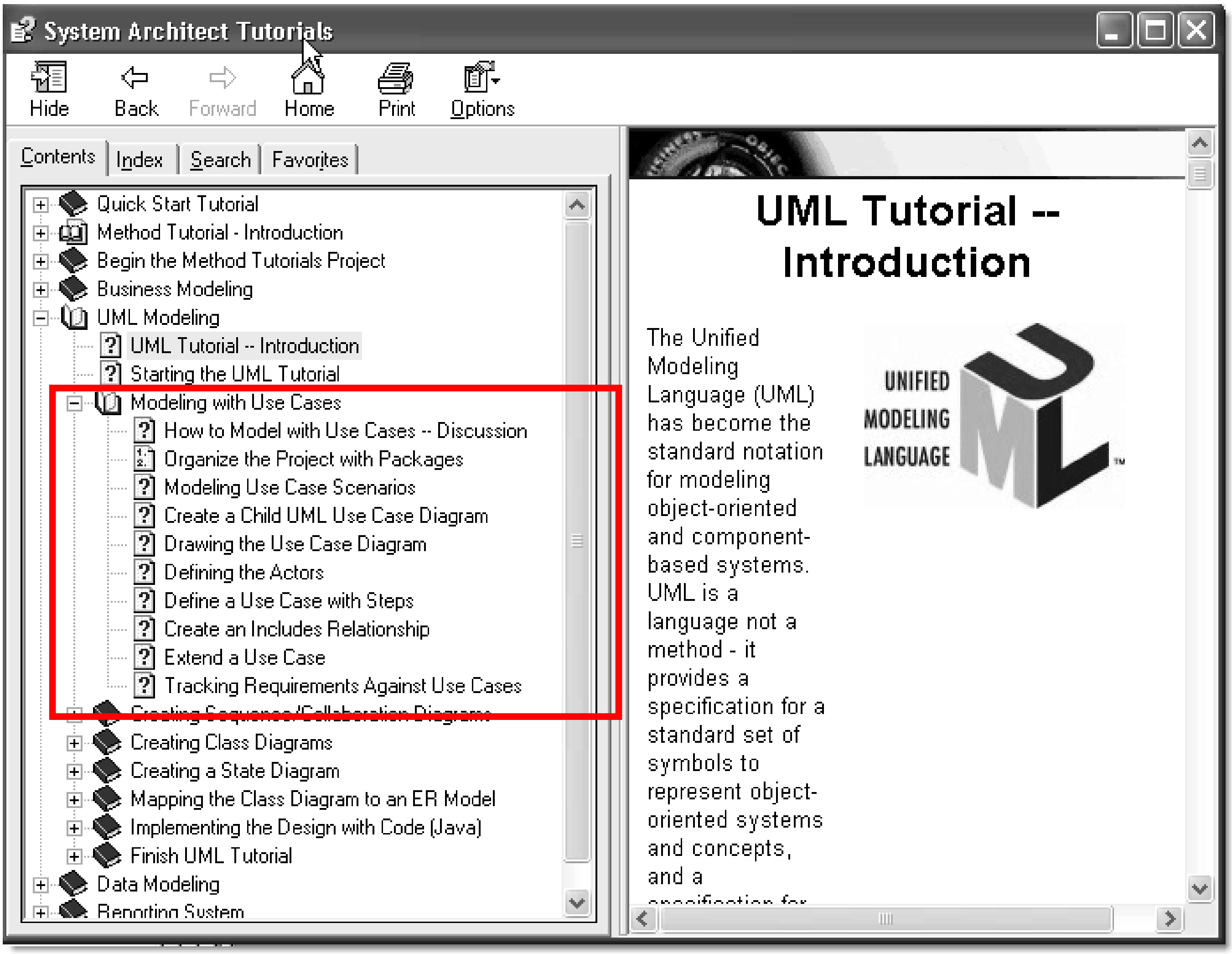
System architect Tutorial 2

UML Use Cases

This material has been abstracted and slightly modified from the System Architect tutorial File

Robin Beaumont robin@organplayers.co.uk 03/01/2004 13:02



- Quick Start Tutorial
- Method Tutorial - Introduction
- Begin the Method Tutorials Project
- Business Modeling
- UML Modeling
  - UML Tutorial -- Introduction
  - Starting the UML Tutorial
  - Modeling with Use Cases
    - How to Model with Use Cases -- Discussion
    - Organize the Project with Packages
    - Modeling Use Case Scenarios
    - Create a Child UML Use Case Diagram
    - Drawing the Use Case Diagram
    - Defining the Actors
    - Define a Use Case with Steps
    - Create an Includes Relationship
    - Extend a Use Case
    - Tracking Requirements Against Use Cases
  - Creating Sequence/Collaboration Diagrams
  - Creating Class Diagrams
  - Creating a State Diagram
  - Mapping the Class Diagram to an ER Model
  - Implementing the Design with Code (Java)
  - Finish UML Tutorial
- Data Modeling
- Reporting System

UML Tutorial -- Introduction

The Unified Modeling Language (UML) has become the standard notation for modeling object-oriented and component-based systems. UML is a language not a method - it provides a specification for a standard set of symbols to represent object-oriented systems and concepts, and a specification for

Contents

Comments I have added are in red (RB)

Starting the UML Tutorial

1. Starting System Architect

To start System Architect:

1. Run System Architect from the **Start**, **Programs** menu or double-click on the **System Architect** program icon.

1.1. Creating/Opening an Encyclopedia

We will open an encyclopedia that has already been populated with some of the systems we will build. The project encyclopedia we will open is called **Tutorial.**

To begin any project in System Architect, you must open the 'encyclopedia' the project is contained in, or create a new 'encyclopedia' for a new project. Normally, when creating a new encyclopedia, you are free to name it whatever you'd like, and create as many encyclopedias as you like (although you should have a rational for managing encyclopedias first).

For the student version of System Architect, however, you are limited to six hard-coded encyclopedia. Their names are **PROJECT1** through **PROJECT4**, and two encyclopedias that are pre-loaded with information -- **SAMPLES** and **TUTORIAL.** These are specially encrypted encyclopedias only for use with the Student version of System Architect. Encyclopedias built with the standard version of System Architect cannot be opened with the Student version. The first four encyclopedias, **PROJECT1** through **PROJECT4**, are provided for you to do your own coursework in. The **SAMPLES** encyclopedia is an encyclopedia that provides sample diagrams for all major methods supported by the tool.

This Tutorial

In this tutorial, you will begin by opening the **TUTORIAL** encyclopedia. It is a project encyclopedia that already has some information and diagrams modeled in it. You will build upon this information as you take the tutorial.

Important Note: In this tutorial, we will model in the encyclopedia provided called Tutorial, using information already provided in it as a starting point. If other team members have already taken this tutorial and modified the Tutorial encyclopedia, you may want to start with a fresh version of this encyclopedia as a starting point. A zip file, **Tutorial.zip**, is provided in the Tutorial directory. It provides a fresh version of the Tutorial encyclopedia. You should unzip this file to a new path before beginning these exercises.

What Is an Encyclopedia?

The System Architect encyclopedia is a relational database in a single sub-directory on your computer. Whether you're working on a stand-alone PC, or as part of a network in a client/server system, the relationship between the encyclopedia and a sub-directory is 1-to-1. That is, the encyclopedia is in a sub-directory; one sub-directory contains just one encyclopedia.

All the diagrams and all the definitions, both those associated with diagrams and those that aren't, are in the encyclopedia.

If you have several projects, you may have several encyclopedias, but that's entirely an individual decision. For example, you can build an enterprise encyclopedia in which you keep all the information relating to your company's standards as well as information that has been used in old projects. When the company decides to start a new project, such as an Accounts Payable (AP) system, those parts of the enterprise encyclopedia that should be used as a "leaping off point" for the AP system are exported to a new encyclopedia. And after the AP system is completed, some of the information discovered in the process of designing that system might be merged into the enterprise encyclopedia.

1.2. Opening the Tutorial Encyclopedia

To open the **TUTORIAL** encyclopedia, select **File, Tutorial**.

1.3. Setting Preferences

Now you are almost ready to begin the tutorial project. However, first you need to set your preferences for how System Architect will react to your drawing commands.

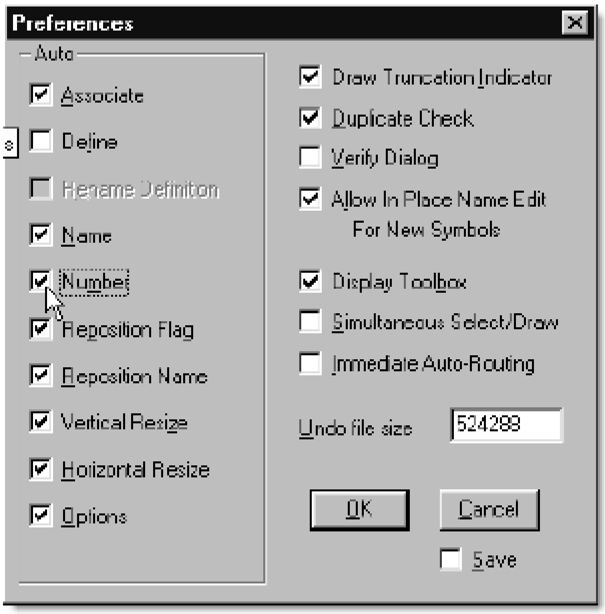
System Architect offers many optional drawing and tool behavior features. They can be turned on or off at any time with the Preferences dialog.

To set Preferences:

From the Tools menu, select Preferences. The Preferences dialog box appears.

Set the selections to match those shown in the diagram below.

Click OK to close the Preferences dialog



Preferences

Auto

☑ Associate

□ Define

□ Rename Definiton

☑ Name

☑ Number

☑ Reposition Flag

☑ Reposition Name

☑ Vertical Resize

☑ Horizontal Resize

☑ Options

☑ Draw Truncation Indicator

☑ Duplicate Check

□ Verify Dialog

☑ Allow In Place Name Edit For New Symbols

☑ Display Toolbox

□ Simultaneous Select/Draw

□ Immediate Auto-Routing

Undo file size 524288

□ Save

Explanation of Some Preferences

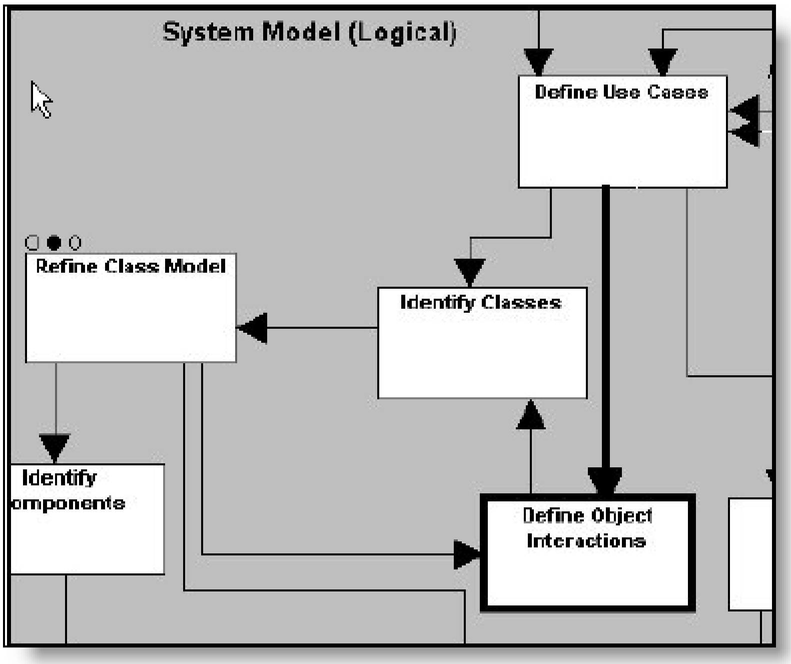
Preferences (Auto,...)	**Setting**	**Explanation**
Associate	On	Brings up the Associative dialog whenever you draw an association line.
Name	On	Brings up the **Add Name** dialog whenever you draw a symbol on a diagram.
Define	Off	Enables you to brainstorm; in other words, place symbols on the diagram and define them later.

For a fuller explanation of the Preferences dialog, go to the Help menu and select Help for the On-line Help System. You can also find this information in your System Architect User Guide or System Architect Tutorial manual, as well. These manuals are provided in .pdf format in the **System Architect Student Version\Manuals** directory.

The completion time of this section is approximately 50 minutes.

2. How to Model with Use Cases -- Discussion

In this section we will begin using Use Cases to model the business scenarios. Within the scenario map for this tutorial, we will perform the process highlighted in bold in the picture below.



On a Use Case diagram, you model Use Cases, representing the scenarios in the business or system, and the external actors which interact with the scenarios, helping you capture the way a system or business works or should work. The scenarios are described with text or with a sequence of steps. Use Case modeling is considered a problem-driven approach to object-oriented analysis because the designer gives primary consideration to the problem at hand and not to the relationship between objects (as in the data-driven approach).

Modeling with Use Cases can illuminate the need for objects in your system. Once the system has been described in terms of its scenarios, the modeler can examine the text or the steps defining each scenario and determine necessary objects.

When building a new system, designers model scenarios which describe the way the system or business should work. When redesigning an existing system, many modelers choose to model first the scenarios of the current system and then the scenarios for how the system should work.

Use Cases are also used in testing a design. Once the design is complete, the modeler can walk through the steps of the scenarios to determine if the design enables the scenarios to occur as planned.

Do not worry too much if you do not understand much on this page – once you have worked through the material it will become clearer.

The SA tutorial assumes that you have learnt about this method, do not worry too much about it. RB

Use Cases vs Process Models

Where do Use Cases fit in with the process and functional modeling that we have done so far in this tutorial? They differ in approach: If a user is strictly using UML to model their business and systems, they may use Use Cases at the very start of contextual modeling, creating a Context Use Case diagram that divides the segments of the business with packages, and create Use Cases for each business segment package. UML Activity diagrams might then be used to model the process flow of each business Use Case.

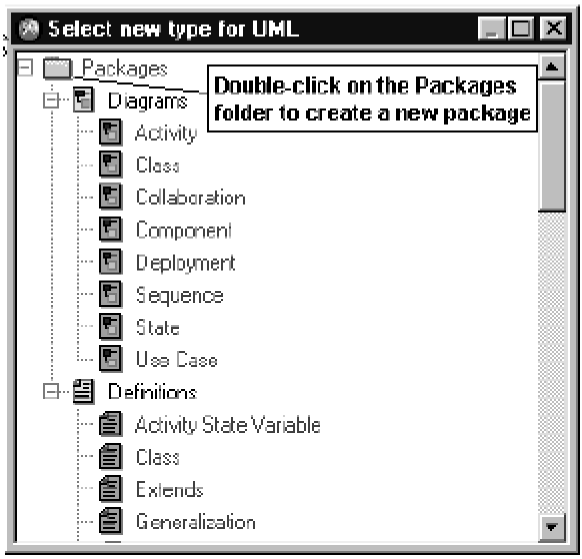
Other users prefer to use the techniques presented in this tutorial so far - using process chart diagrams to get a handle on the business events, process flows, and results, and the functions that these processes create for contextual and conceptual business modeling. Once these diagrams have been created, the user has the option of using Use Cases as a bridge into object-oriented analysis and design of the business systems. You might create a Use Case corresponding to every low-level process (Elementary Business Process) on the Process Chart diagram, then use the techniques of UML to examine the Use Cases to find objects. Although this introduces a second set of diagrams representing similar ideas (and thus added modeling maintenance, since a change to a business process would have to be represented in both places), it can provide a clear bridge between conceptual business modeling and system modeling with UML.

A third alternative would be to skip UML Use Cases entirely, and use the Process Charts to drive the next levels of application design (i.e., creating Sequence and Class diagrams).

1.4. Organize the Project with Packages

As we start our business modeling with Use Cases, the first thing we want do is define the scope of our project. We can categorize the parts of the business we will model using packages. For example, in a hotel business, the Use Cases involved with making a reservation should not be mixed with those involved with paying the employees, which should not be mixed with those involved in maintaining the hotel. Instead, you might categorize them as separate packages: Reservations, Human Resources, and Building Maintenance.

In addition, we may want to categorize our project in terms of the stages of development. For example, we may want to create packages for the Business Use Case View, the Logical View, the Physical Component View, and the Physical Deployment View. In modeling this way, we can create business elements that are separate than logical elements that are separate from design elements -- for example, a Customer class in the Business Use Case View will be a different class than a Customer class in the Logical View.



Double-click on the Packages folder to create a new package

Create a Business Use Case View Package

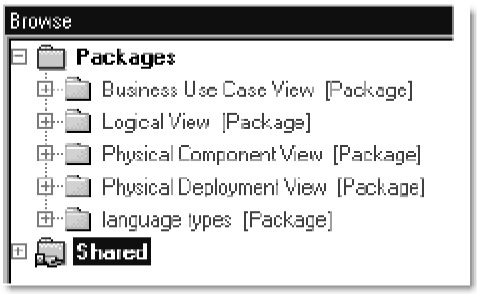
For this Tutorial we'll subdivide our project into packages for various stages of development -- we'll create a Business Use Case View, a Logical View, a Physical Component View, and a Physical Deployment View. Within the Business Use Case View, we'll build a Reservations package to hold the Use Cases of that system.

Let's create a Business Use Case View package:

1. Select the **UML** tab in the browser. Right-mouse click on the **Packages** folder and select New. The **Select new type for UML** dialog will open.
2. Double click on the **Packages** folder to create a new package. The **Dictionary Object - Package** dialog will open.

3. Type in the name of the new package, **Business Use Case View**. Click **OK**. Another **Dictionary Object - Package** dialog will open, this time enabling you to specify the parent package.

4. Since we want to create this package at the highest level, we'll leave the **Parent Package - (Package)** field empty (the Packages folder that it will fall under doesn't count). Click **OK** to close the second dialog. The new package will be created, and will be placed in alphabetical order under the **Packages** folder.



Browse
Packages
Business Use Case View [Package]
Logical View [Package]
Physical Component View [Package]
Physical Deployment View [Package]
language types [Package]
Shared

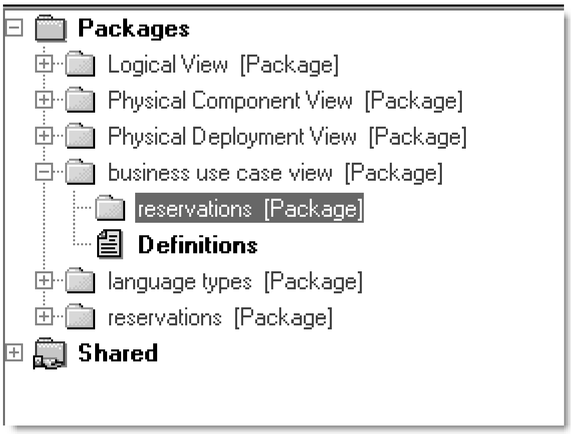
Let's now further refine what part of the business we will model with Use Cases.

5. Right-mouse click on the **Business Use Case View** folder and select **New**. The **Select new type for UML** dialog will open.

6. Double click on the **Packages** folder to create a new package. The **Dictionary Object - Package** dialog will open.

7. Type in the name of the new package, **Reservations**. Click **OK**. Another **Dictionary Object - Package** dialog will open, this time enabling you to specify the parent package.

8. Click on the **Choices** button next to the **Parent Package - (Package)**. From the list of packages provided, select **Business Use Case View**, and holding down your left mouse button, drag-and-drop it into the **Parent Package** field. Click **OK** to close the dialog.



Packages
Logical View [Package]
Physical Component View [Package]
Physical Deployment View [Package]
business use case view [Package]
reservations [Package]
Definitions
language types [Package]
reservations [Package]
Shared

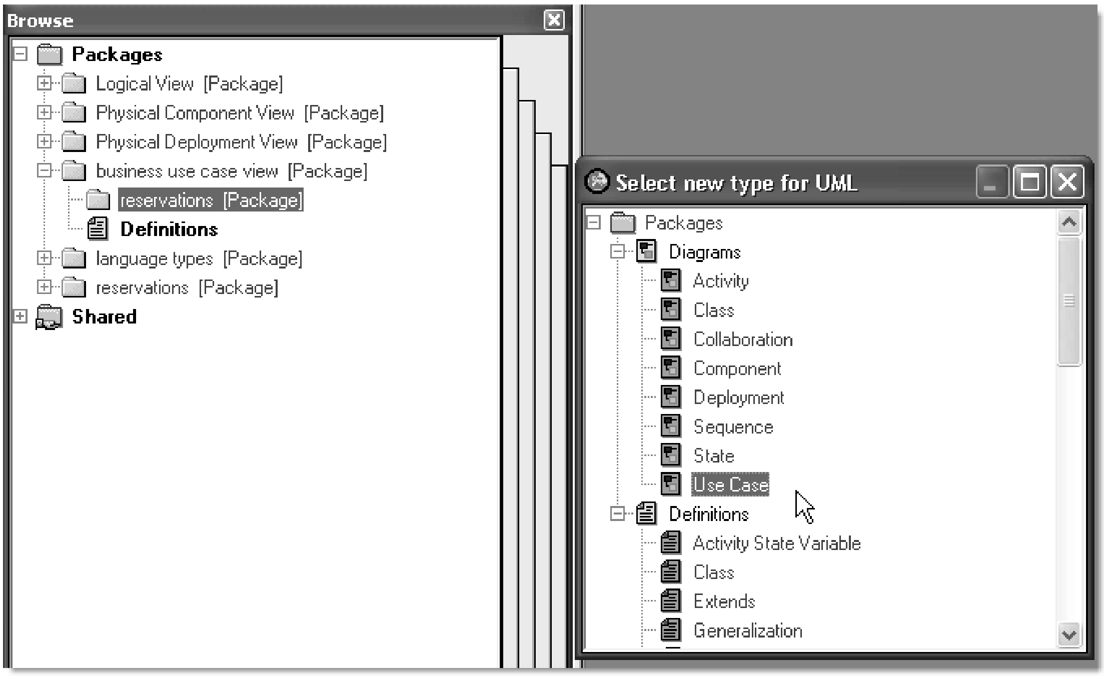
1.5. Modeling Use Case Scenarios

Creating the Context Use Case Diagram

During business analysis of the system, you can develop one Use Case diagram that acts as the context Use Case diagram. On this diagram, you can draw the overall Use Cases of the domain, with actor interactions. One of the goals of drawing a Context Use Case diagram is to identify all of the actors of the system. You may optionally use the System symbol to further refine the categorization of the Use Cases.

Create a Use Case Diagram

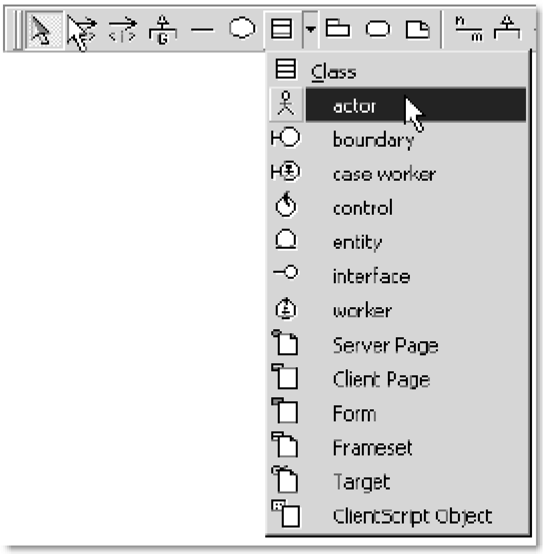
1. In the **UML** tab of the browser, right-mouse click on the package **Reservations** and select **New**. The **Select new type for UML** dialog opens.



2. Double-click on **Use Case** (under the list of **Diagrams**). The **New Diagram** dialog opens.
3. Type in the name of the new Use Case diagram, **Reservations Context Diagram**. Click **OK**. In the next dialog that opens (titled **Diagram - Use Case - Reservations Context Diagram**), you may specify the package that this diagram will be contained in. It is automatically filled in with the name of the package you right-mouse clicked on (**Reservations)**. You can specify a different package at this point, but we'll leave it **Reservations**.
4. Click **OK** to close the dialog.

Modeling an Actor

The first goal of modeling scenarios is to determine how an external actor uses the system to gain something from it and what scenarios the actor can go through. When making a reservation, it is an external actor called Customer who can make a reservation.



5. From the **Draw** toolbar, select the **Actor** symbol. (The Actor symbol is a stereotype of a class -- it is located under the Class symbol. Click on the down button next to the Class symbol and select the Actor symbol. It resembles a stick figure.) Your cursor turns into a drawing pen.
6. Left-mouse click once on the diagram workspace to draw the actor. Name the actor Customer, and hit Enter.

Hint: Remember you can select and move the object around the screen by either hitting the **escape key** or clicking on the Selection Mode icon on the tool bar (far left)

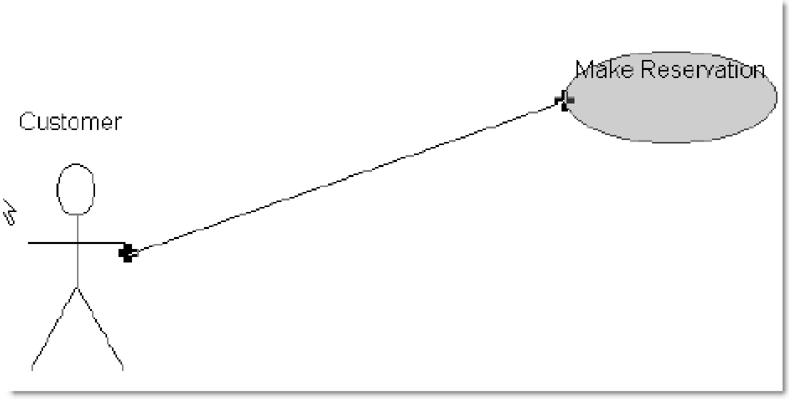
Modeling a Use Case

The Customer can make a reservation. Let's draw a Use Case called **Make Reservation**.

7. Select the Use Case symbol on the toolbar (looks like an oval). Your cursor will turn into a drawing pencil. Draw with it on the diagram by clicking it. Name the Use Case **Make Reservation**.

During Use Case analysis, the **Use Case Association** line is used to show simple communication between an actor and a Use Case.

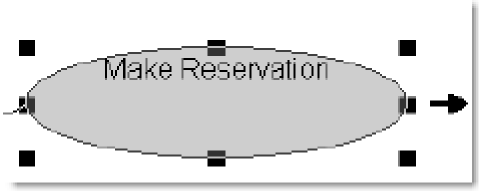
8. From the toolbar, select the **Use Case Association** line drawing too. Your cursor will turn into a drawing pencil.



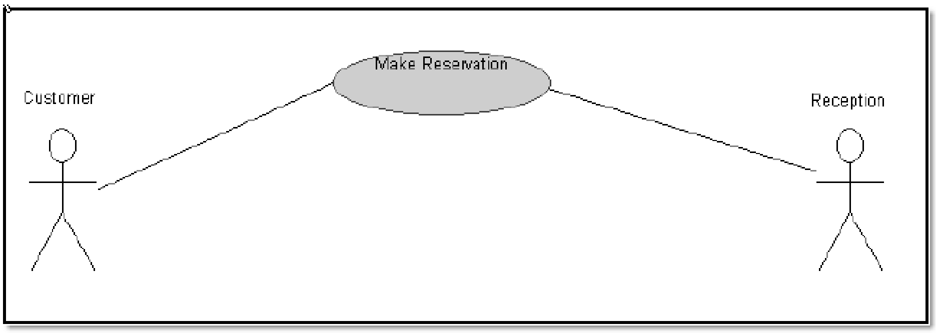
9. Draw an association between the **Customer** actor and the **Make Reservation** Use Case as follows: Place the drawing cursor inside the **Customer** actor and click once (and release) so that you get a **+** connector indicator. Move your line over to the **Make Reservation** Use Case to get a **+** connector indicator again. Once you get a **+** indicator at both ends of the Use Case Association line, left click your mouse again to make the connection.

You may want to widen the Use Case at this point to fit the name better.

10. Select the Use Case and drag on its right center handlebar to widen it. See the hint above if you have forgotten how to select objects.



To set this width as the default for all future Use Cases drawn, select the Use Case and choose the menu option **Format, Symbol Format, Symbol Style**. In the **Symbol Style** dialog that opens, toggle on **Set Size** and click **OK**.



One other actor communicates with this Use Case - **Reception.** Now reception may be a person, or it may be the company's website. If there is a difference between the two, we might model them as two different actors, but for now we will model it as one actor - Reception.

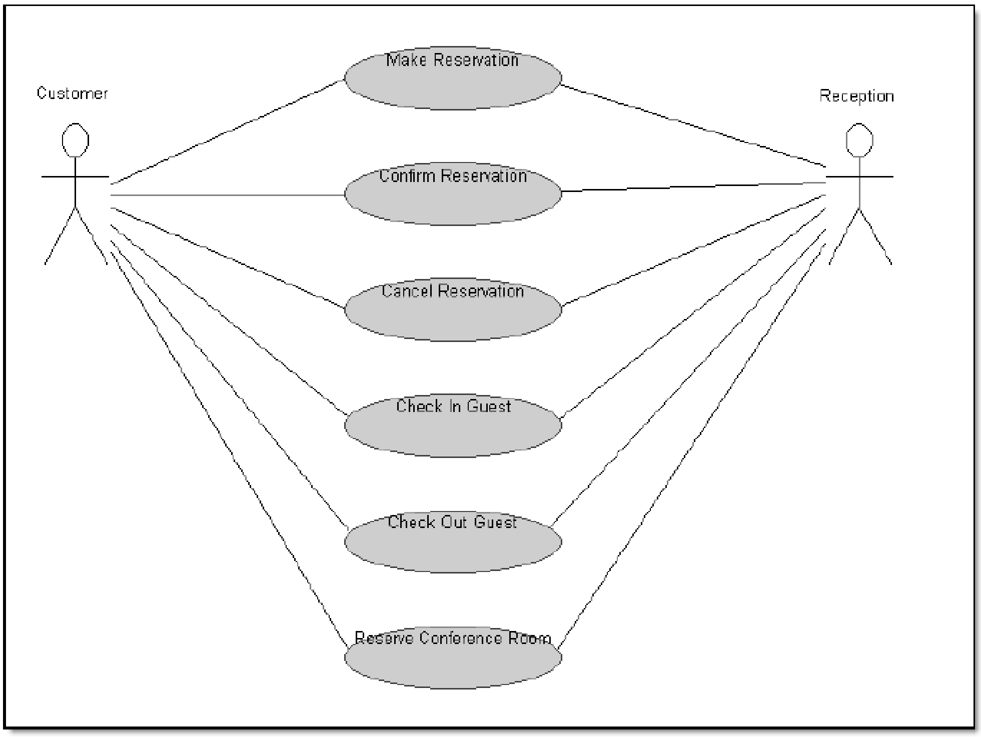
11. From the **Draw** toolbar, select the **Actor** symbol. Place the actor symbol on the diagram to the right of the **Make Reservation** Use Case, name it **Reception**, and hit **Enter**.
12. From the toolbar, select the **Use Case Association** line drawing tool and draw a **Use Case Association** line from the actor **Reception** to the Use Case **Make Reservation**.

A customer also interacts with our reservation by confirming a reservation, cancelling a reservation, and reserving a conference room. In addition, reception may check-in a guest or check-out a guest. Let's model these scenarios as Use Cases:

13. Draw the following Use Cases, under Make Reservation:

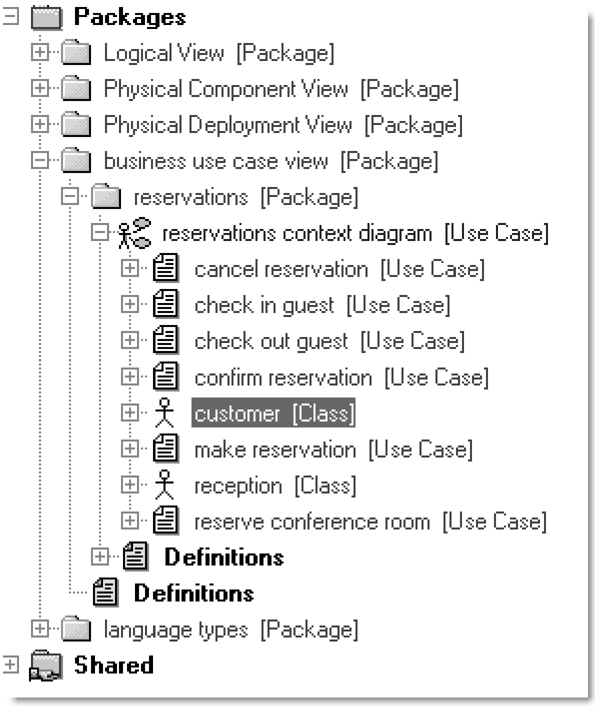
- **Confirm Reservation**
- **Cancel Reservation**
- **Check In Guest**
- **Check Out Guest**
- **Reserve Conference Room**

14. Connect the **Customer** actor to each Use Case with a **Use Case Association** line, and connect the **Reception** actor to each Use Case with a **Use Case Association** line to create the diagram opposite:



This is a first level of Use Cases for our Hotel Reservation System. We are not trying to get into too much detail here, just establish the main scenarios involved with reserving rooms. The Use Cases on this diagram are just defining the main processes.

If you expand the reservations context diagram title now in the browse window you will see all the new use cases listed.

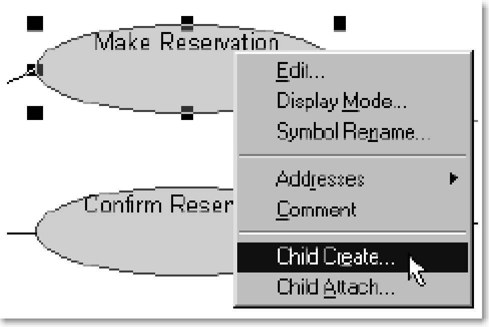


```
Packages
  Logical View  [Package]
  Physical Component View  [Package]
  Physical Deployment View  [Package]
  business use case view  [Package]
    reservations  [Package]
      reservations context diagram  [Use Case]
        cancel reservation  [Use Case]
        check in guest  [Use Case]
        check out guest  [Use Case]
        confirm reservation  [Use Case]
        customer  [Class]
        make reservation  [Use Case]
        reception  [Class]
        reserve conference room  [Use Case]
      Definitions
    Definitions
  language types  [Package]
Shared
```

1.6. Create a Child UML Use Case Diagram

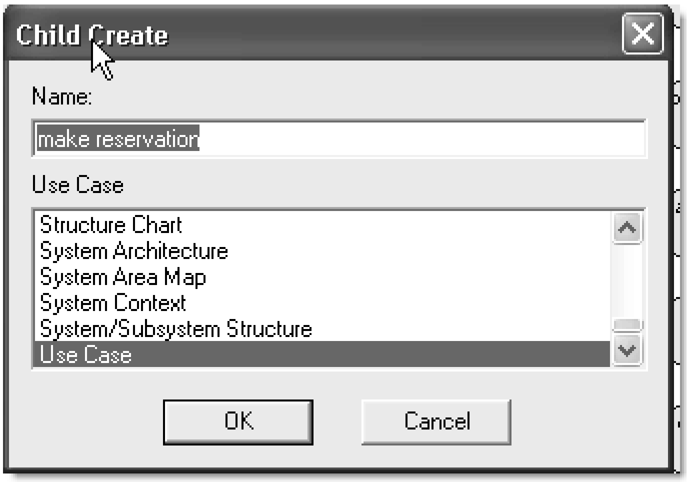
Now let's model the **Make Reservation** Use Case in greater detail. We'll create a child Use Case diagram:

1. Select the **Make Reservation** Use Case then right-mouse click on it and select **Child Create** from the drop-down menu.



2. The **Child Create** dialog opens. Leave the default selections to create a Use Case diagram named **Make Reservation**, and click **OK**

3. You will then receive a message asking you if you wish to save the changes to the **Reservations Context Diagram**. Click **Yes**

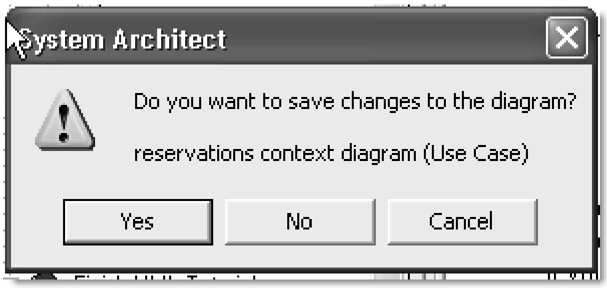


Child Create

Name: make reservation

Use Case

Structure Chart
System Architecture
System Area Map
System Context
System/Subsystem Structure
Use Case



System Architect

Do you want to save changes to the diagram?

reservations context diagram (Use Case)

4. Finally you will be asked what package the new Use case diagram belongs to – accept the default and click OK to close the dialog box.

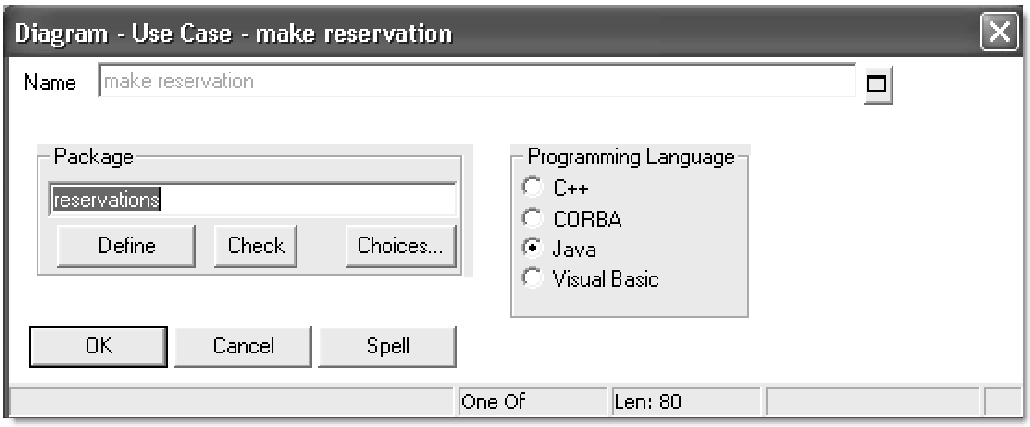


Diagram - Use Case - make reservation

Name: make reservation

Package: reservations

Programming Language
○ C++
○ CORBA
● Java
○ Visual Basic

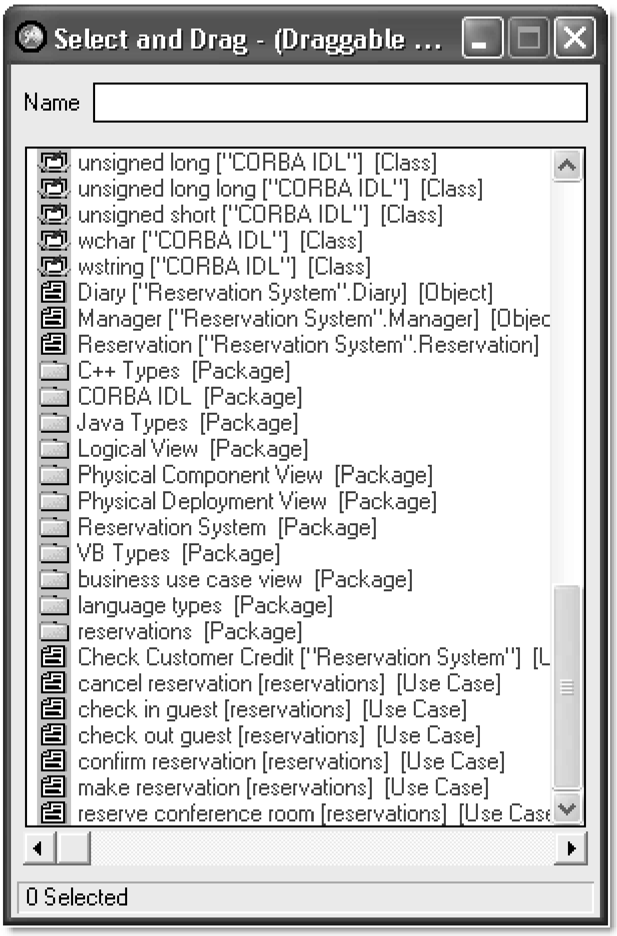
One Of | Len: 80

You now have a new blank canvas to start to draw the **Make Reservation** use case diagram.

1.7. Drawing the Child Use Case Diagram

We have already modeled part of the Make Reservation Use Case scenario on the Use Case Context diagram. Let's reuse that information.

First let's redraw the **Make Reservation** Use Case that was already included on the Use Case Context diagram.



Select and Drag - (Draggable ...

Name

- unsigned long ["CORBA IDL"] [Class]
- unsigned long long ["CORBA IDL"] [Class]
- unsigned short ["CORBA IDL"] [Class]
- wchar ["CORBA IDL"] [Class]
- wstring ["CORBA IDL"] [Class]
- Diary ["Reservation System".Diary] [Object]
- Manager ["Reservation System".Manager] [Objec
- Reservation ["Reservation System".Reservation]
- C++ Types [Package]
- CORBA IDL [Package]
- Java Types [Package]
- Logical View [Package]
- Physical Component View [Package]
- Physical Deployment View [Package]
- Reservation System [Package]
- VB Types [Package]
- business use case view [Package]
- language types [Package]
- reservations [Package]
- Check Customer Credit ["Reservation System"] [L
- cancel reservation [reservations] [Use Case]
- check in guest [reservations] [Use Case]
- check out guest [reservations] [Use Case]
- confirm reservation [reservations] [Use Case]
- make reservation [reservations] [Use Case]
- reserve conference room [reservations] [Use Case

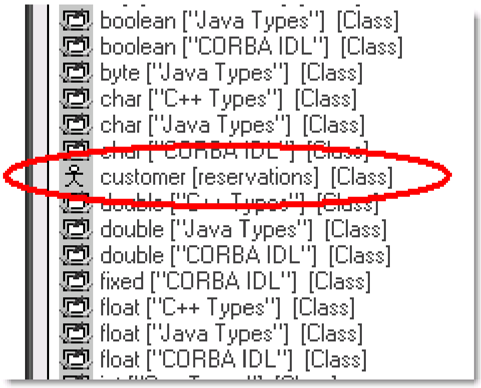
0 Selected

1. Select the **Select Mode** pointer tool from the toolbar, then right-mouse click anywhere on the white area on the Use Case diagram workspace and select **Choices** from the drop-down list. The **Select and Drag** dialog gives you a list of all modeling elements that already exist in this encyclopedia that can be reused on this diagram.

2. At the bottom of the **Select and Drag** dialog, find the Use Case **Make Reservation**. Select and drag it onto the diagram workspace.

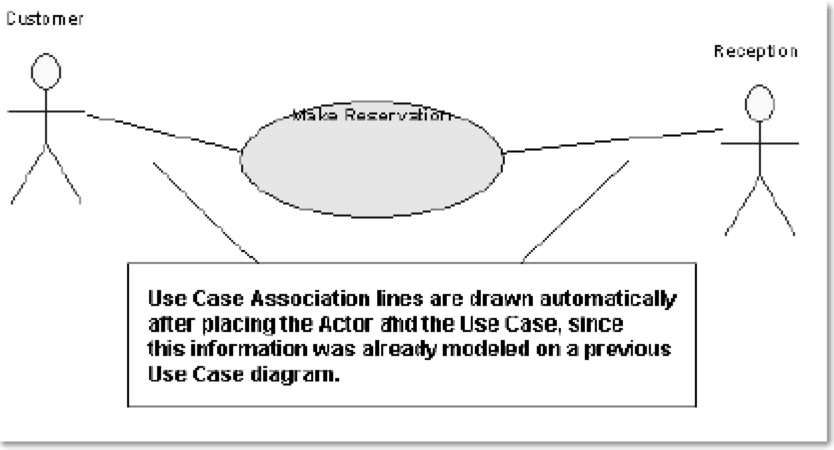
3. At the top of the **Select and Drag** dialog, find the class labeled **Customer [Reservations]** (this is the class name and the package name). Select and drag it onto the diagram workspace to the left of the **Make Reservations** Use Case. Notice that the Use Case Association is automatically drawn between them.

4. In the **Select and Drag** dialog, find the class **Reception[Reservations]**, and select and drag it onto the diagram workspace to the right of the **Make Reservations** Use Case. Once again the Use Case Association line is drawn automatically between them.



- boolean ["Java Types"] [Class]
- boolean ["CORBA IDL"] [Class]
- byte ["Java Types"] [Class]
- char ["C++ Types"] [Class]
- char ["Java Types"] [Class]
- char ["CORBA IDL"] [Class]
- customer [reservations] [Class]
- double ["C++ Types"] [Class]
- double ["Java Types"] [Class]
- double ["CORBA IDL"] [Class]
- fixed ["CORBA IDL"] [Class]
- float ["C++ Types"] [Class]
- float ["Java Types"] [Class]
- float ["CORBA IDL"] [Class]

5. Click the **x** in the upper-right-hand corner of the **Select and Drag** dialog to close it.



Use Case Association lines are drawn automatically after placing the Actor and the Use Case, since this information was already modeled on a previous Use Case diagram.

1.8. Defining the Actors

The Customer and Reception actors are considered to be classes of objects with a stereotype of actor. If you open up an actor's definition dialog (by double clicking on the object), you will see a definition dialog for a class, with Stereotype property filled in for Actor.

Although we could add details to the definitions of the actors we have drawn at this point, such as attributes, we will not. We will add information about classes when we take a closer look at the behavior of these classes (and their instances, or objects) with Sequence diagrams in the next section.

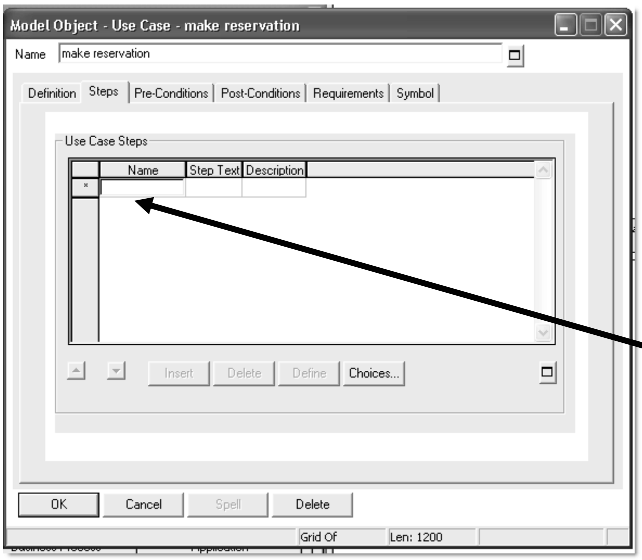
Defining a Class: You do not normally need to detail any information about the class at this stage of development. In fact, during early, business-level analysis of the system you may want to purposely avoid spending any time adding details so that you do not get sidetracked from the important thing at hand - modeling the dynamics of the scenario. Later, you may go back to the class definition and add details. In this tutorial in fact, we will create a new logical, Customer class in a later section.

1.9. Define a Use Case with Steps

The simplest way to define a Use Case is by writing a textual description of the scenario. There are other ways to define the Use Case, for example, listing the steps that form the scenario. In addition, one may want to list the pre-conditions that must exist before the scenario can take place and the post-conditions that must exist after the scenario has run its course.

Let's start with some basic steps for Making a Reservation, assuming the room the customer requests is available and the customer's credit is good:

1. Customer Queries for Available Rooms
2. Store Customer Details
3. Check Diary for Room Availability
4. Room is Available
5. Advise Customer of Availability
6. Customer Requests Reservation
7. Provisionally Book Room
8. Figure Out Price; Advise Customer
9. Customer Accepts Terms
10. Check Customer Credit
11. Customer Credit OK
12. Reserve Room



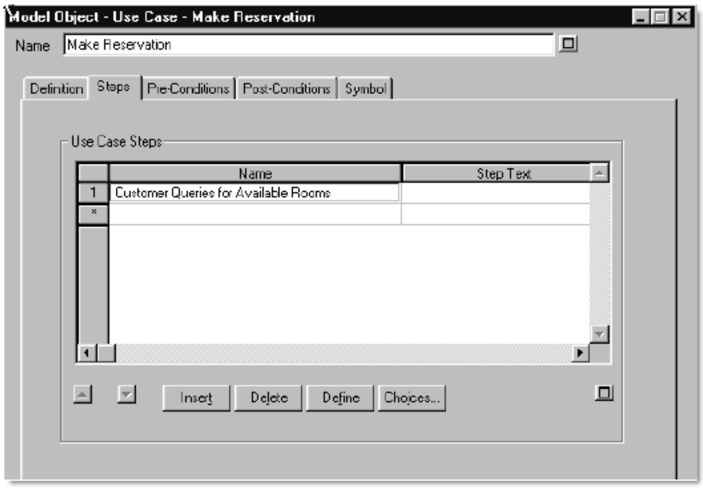
Model Object - Use Case - make reservation

Name: make reservation

To add these steps to the definition of the Make Reservation Use Case:

1. Open the definition dialog of the **Make Reservation** Use Case by double-clicking on it or right-mouse-clicking on it and selecting **Edit**.
2. Select the **Steps** tab within the Use Case definition. You will see a grid for Use Case Steps.
3. Put your cursor in the **Name** column of the first row of the Use Cases Steps grid, and type in the first step: **Customer Queries for Available Rooms**. Hit your **Enter** key to add it to the grid.

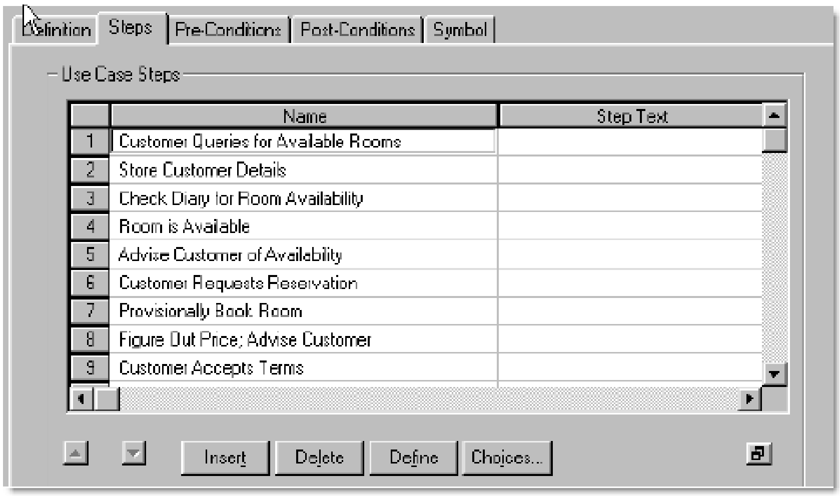
Note that if you place your cursor at the column borders at the head of the grid, you can increase or decrease the widths of each column.


Model Object - Use Case - Make Reservation

Name: Make Reservation

	Name	Step Text
1	Customer Queries for Available Rooms	
*		

4. Add the remaining 11 Use Case Steps listed at the start of this section, hitting your **Enter** key after typing in each one to add it to the grid.

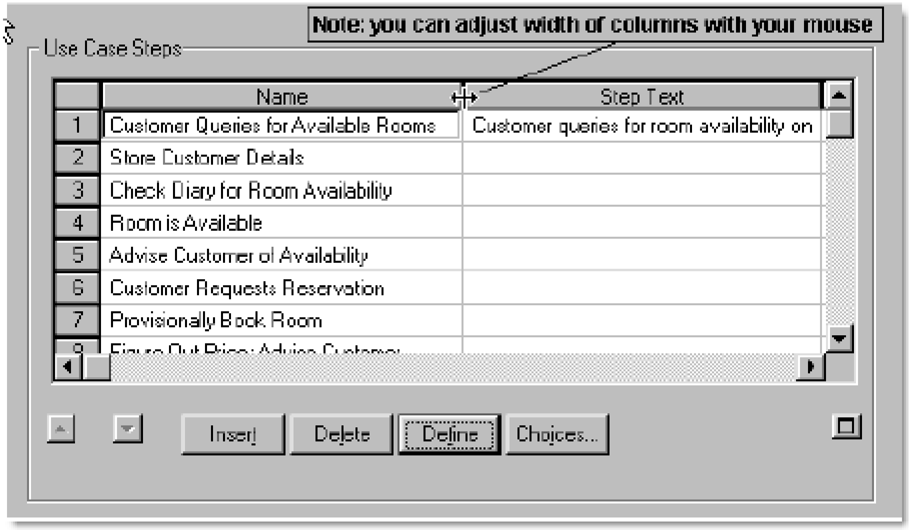


	Name	Step Text
1	Customer Queries for Available Rooms	
2	Store Customer Details	
3	Check Diary for Room Availability	
4	Room is Available	
5	Advise Customer of Availability	
6	Customer Requests Reservation	
7	Provisionally Book Room	
8	Figure Out Price; Advise Customer	
9	Customer Accepts Terms	

Define a Step

You may specify a textual description of each step. You may type the Step Text directly into the grid, in the cell next to the step you wish to describe, or place your cursor in the step's Name box and click on Define to open the full definition of the step, and type the text in the Step Text field.

5. In the Step Text column, in the row for Customer Queries for Available Rooms, add the Step Text: **Customer queries for room availability on certain dates.**


Note: you can adjust width of columns with your mouse

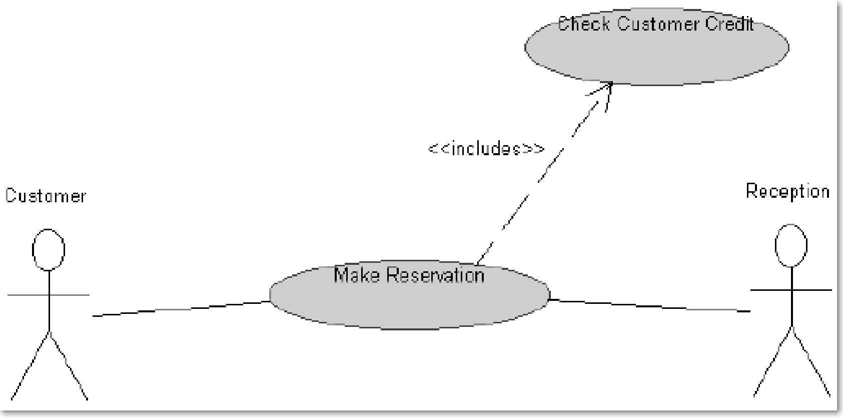
	Name	Step Text
1	Customer Queries for Available Rooms	Customer queries for room availability on
2	Store Customer Details	
3	Check Diary for Room Availability	
4	Room is Available	
5	Advise Customer of Availability	
6	Customer Requests Reservation	
7	Provisionally Book Room	
8	Figure Out Price; Advise Customer	

6. Finally click the **OK** button to close the dialog box

1.10. Create an Includes Relationship between Use Cases

When modeling Use Cases, you will model a separate Use Case for every unique scenario. There will be certain sub-scenarios that will occur over and over again within Use Cases of your system. Instead of describing each sub-scenario over and over again, you can describe it once, in a separate Use Case, and have other Use Cases "Include" this Use Case.

Within our system, the sub-scenario of Checking a Customer's Credit is an ideal candidate to be pulled into its own Use Case. The Check Customer Credit Use Case is already included in this Tutorial encyclopedia - we will reuse it and establish a relationship between Make Reservation and it.

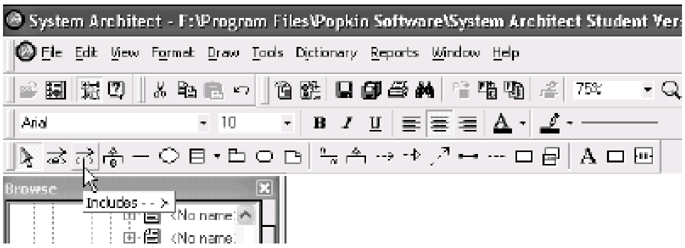


Select the **Select Mode** pointer tool from the toolbar and right-mouse click in any empty area on the Use Case diagram workspace. Select **Choices** from the drop-down list. The **Select and Drag** dialog gives you a list of all modeling elements that already exist in this encyclopedia that can be reused on this diagram.

Select-and-drag the Use Case **Check Customer Credit** onto the diagram workspace, above and to the right of **Make Reservation**.

Click the **x** in the upper-right-hand corner of the **Select and Drag** dialog to close it

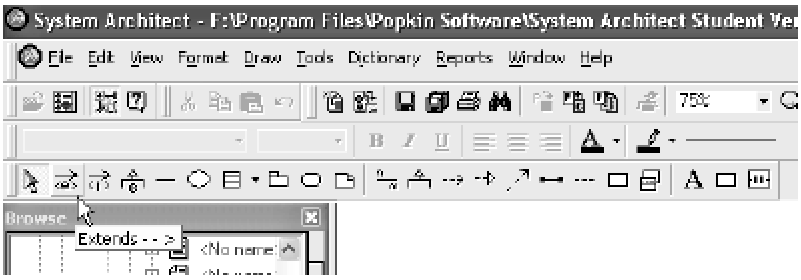
From the toolbar, select the **Includes** line drawing tool (third from the left) and draw an **Includes** line from the Use Case **Make Reservation** to the Use Case **Check Customer Credit.** You do not need to name the line, simply hit your **Enter** key (or left mouse click) to draw the line unnamed. Once drawn, you may adjust the positioning of the **<<includes>>** stereotype on the line.


System Architect - F:\Program Files\Popkin Software\System Architect Student Ver
File Edit View Format Draw Tools Dictionary Reports Window Help
75%
Arial
10
B I U
Browse
Includes - - >
<No name>
<No name>

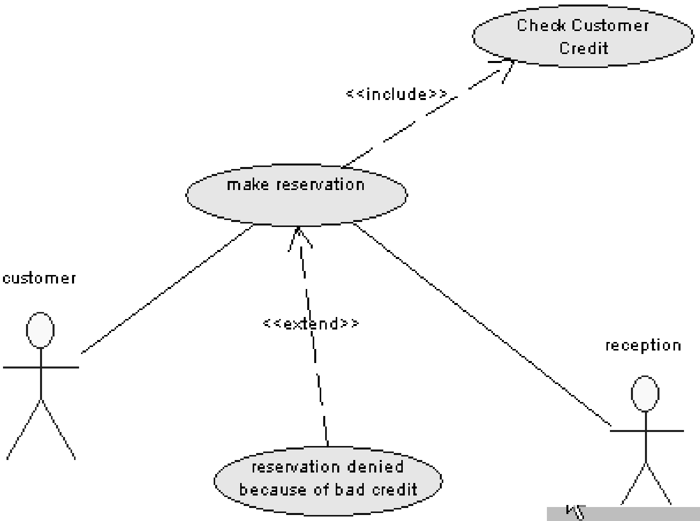
1.11. Extend a Use Case

When modeling Use Cases, there will often be alternative courses of action that we want to model. For example, for the Use Case **Make Reservation**, we assume that the customer's credit is good. If it is no good, the steps of the Use Case will be identical up until the step of checking the customer's credit, after which the steps will be different. We can model all of this behavior in one Use Case or we can simply extend the Use Case Make Reservation with another Use Case, called say, **Reservation Denied Because of Bad Credit**.

1. From the **Draw** toolbar, select the oval Use Case symbol. Place a Use Case symbol on the diagram underneath Make Reservation, name it **Reservation Denied Because of Bad Credit**, and hit **Enter**.



2. From the **Draw** toolbar, select the **Extends** line drawing tool and draw an **Extends** line from the Use Case **Reservation Denied Because of Bad Credit** to the Use Case **Make Reservation**. You do not need to name the line. Once drawn, the **<<extends>>** stereotype shows on the line.

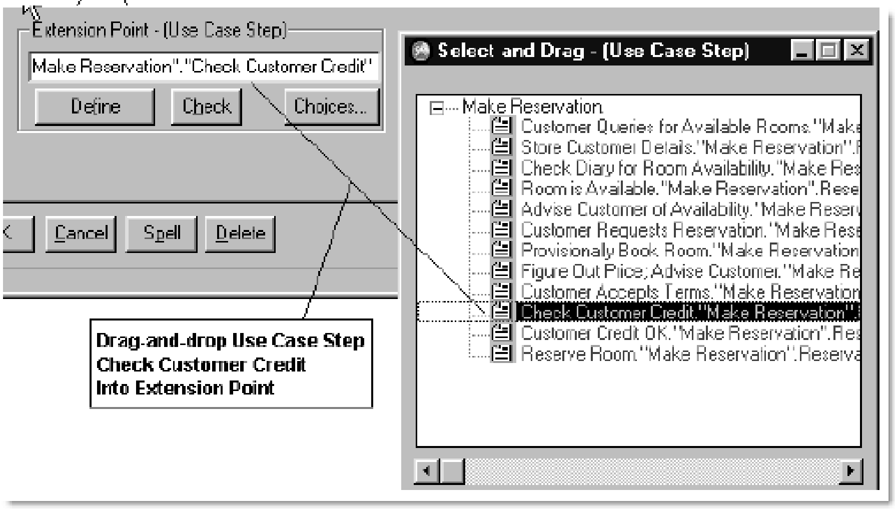


Specify the Extension Point

Open the definition of the extends line between **Reservation Denied Because of Bad Credit** and **Make Reservation** (double-click on the line or right-mouse click on it and select **Edit**).

1. For the **Extension Point** property, click on the **Choices** button to get a list of steps of the extended Use Case.

2. Drag-and-drop the step **Check Customer Credit** into the **Extension Point** field.


Extension Point - (Use Case Step)
Define
Check
Choices...
Cancel
Spell
Delete
Select and Drag - (Use Case Step)

Make Reservation"."Check Customer Credit"

Make Reservation
- Customer Queries for Available Rooms."Make
- Store Customer Details."Make Reservation".
- Check Diary for Room Availability."Make Res
- Room is Available."Make Reservation".Rese
- Advise Customer of Availability."Make Reser
- Customer Requests Reservation."Make Rese
- Provisionally Book Room."Make Reservation
- Figure Out Price; Advise Customer."Make Re
- Customer Accepts Terms."Make Reservation
- Check Customer Credit."Make Reservation".
- Customer Credit OK."Make Reservation".Res
- Reserve Room."Make Reservation".Reserva

Drag-and-drop Use Case Step Check Customer Credit Into Extension Point

When to Stop Decomposing Use Cases

General Use Case theory recommends that at least one Use Case be prepared for each significantly different kind of scenario instance. Each scenario shows a different sequence of interactions between actors and the system, with all decisions definite.

When you have arrived at the lowest Use Case level, you can create a child Sequence diagram (and accompanying Collaboration diagram) for the Use Case. With these Sequence and Collaboration diagrams, you can model the implementation of the scenario.

The information on this page is basically trying to get you to realize that various objects you create are 'managed' by SA (the CASE tool) and because of this you can trace the origin or each one

1.12. Tracking Requirements against Use Cases

Use Cases serve a dual purpose on the requirements level - they can be used to help you find or establish your requirements, or, if you already have set requirements for your system, you can attach those requirements to your Use Cases so that you know your designed models meet your requirements.

In the Business Modeling section of this tutorial, we created a number of business requirements and entered them into System Architect. In this section, we will attach them to the Use Case which describes the part of the system that satisfies those requirements.

1. Open the definition of the Use Case **Make Reservation** (double-click on it or right-mouse click and select **Edit**).
2. Select the **Requirements** tab. This tab and the properties within it were added to the Use Case definition through System Architect's extensibility mechanism - USRPROPS.TXT.
3. In the **Related Requirements** list box, click on **Choices** in the bottom right-hand corner. The **Select and Drag** dialog displays all of the business requirements that you entered in the business modeling section of this tutorial.
4. Right-mouse click in the **Select and Drag** dialog and select **Details** from the drop down list to view a summary box at the bottom of the dialog.

The requirement **Retrieve Credit Information From Consumers** is satisfied by this Use Case.

5. Select the requirement **Retrieve Credit Information From Consumer** in the **Select and Drag** dialog. Drag and drop it into the **Requirements** list box. Close the **Select and Drag** dialog.

Requirements Traceability

6. Select the requirement **Retrieve Credit Information From Consumer** and click **Define** at the bottom of the **Related Requirements** list box to open the requirement's definition dialog.
7. Notice within the requirement's definition there is a tab for **Related Requirements and Process**. Within this tab you can see a list box for related **Business Objectives**.

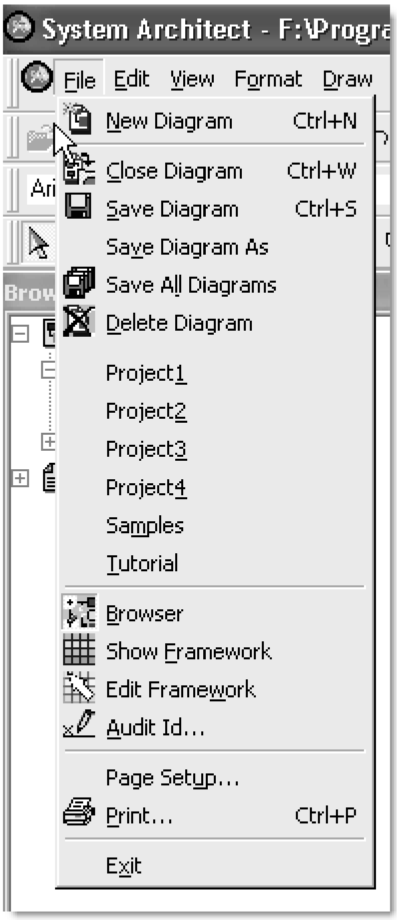
Through these dialogs we can see the requirements traceability; we see why the Use Case exits -- what requirements it satisfies, and for each requirement we see why it exists -- what high-level business objectives it satisfies.

8. Click **OK** to close the dialog.

1.13. Save your work before closing SA

You will make use of the objects you have created in this session in the next SA tutorial. To make sure you have saved your work before closing down SA/SE:

Click on the **File** menu then choose the **Save all Diagrams** option.



3. Summary

In the exercises in the next section, we will discover objects and use Sequence diagrams to model further details on how the objects perform the scenarios described in this section in which you have learnt about Use Cases.

4. Tip- Showing the Browse window after you have closed it

Sometimes you may want to extend the space available to draw with in SA/SE by closing the Browse window. To restore the window you can click on the **Browser** item in the **File** menu. As shown in the above diagram.