



## **LAB 2: Introduction to Use Case Modeling**

### **OBJECTIVES**

- Familiarize with the concept underlining Use Case Diagram for requirement analysis.
- Exposure on all elements grouped under UML Case Diagram.
- Analyze functional requirements and problem statements of an intended system to be able to translate into use case modeling.
- Be able to document use case description and specifications.

### **INTRODUCTION**


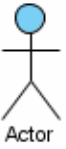
#### **Definition of Use Case Diagram**

- A Use Case is a set of scenarios describing an interaction between a user and a system.
- Use Case diagram displays (in symbolic form) the relationship among actor and use cases.
- Use case represents the system's functionality, the requirements of the system from the user's perspective.


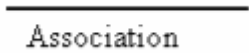

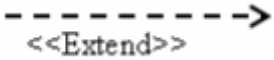
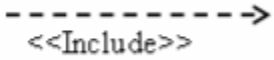
#### **When to use Use Cases**

- Use cases are used in almost every project.
- Very helpful in exposing requirements and planning of project.
- Usually done during the early stages of project. Frequently starts by being loose in terms of connection and accuracy, but as project continues, the use cases are defined better.

#### **UML Notation for Use Case Modeling**

<b>Construct</b>	<b>Description</b>	<b>Syntax</b>
Use Case	A description of a set of sequences of actions, including variants, that system performs that yields an observable value to an actor (Booch, 1993).	
Actor	The people or system that provides or receives information from the system; called the stakeholder of the system.	



System Boundary	The boundary between the physical system and the actors who interact with the physical system.	
Association	The participation of an actor in a use case.	
Generalization	A taxonomical relationship between a general use case and a more specific use case.	
Extend	Relationship between the extension use case and the base use case. Specify how the behavior of the extension use case can be inserted into the behavior defined for the base use case.	
Include	Relationship between a base use case and an inclusion use case, specifying how the behavior of the inclusion use case can be inserted into the behavior defined for the base use case.	

## Actor

- Could be human beings, other system, timers and clocks or hardware device.
- 2 types of actor classification:
  - Primary Actors: Actors that stimulate the system and the initiator of events.
  - Secondary Actors: Actors that only receive stimuli from the system.
- Actor Designing Consideration:
  - Who / what will be interested in the system
  - Who / what will want to change the data in the system
  - Who / what will want to interface with the system
  - Who / what will want information from the system

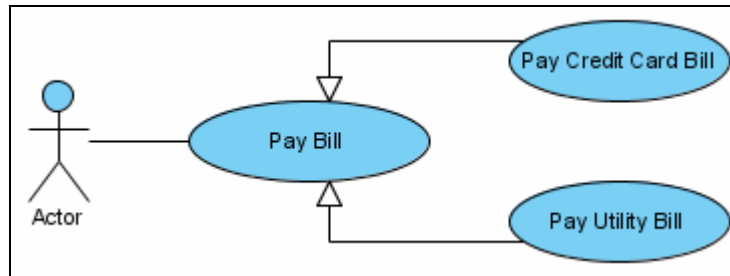
## Use Case

- Use case should ideally begin with a verb.
- Should not be open ended. Register, wrong. Register New User, right.

## Actor / Use Case Generalization

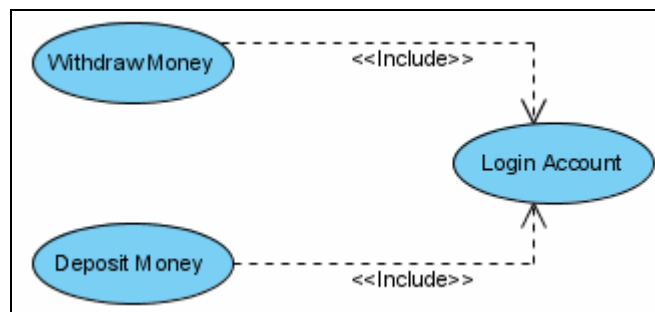
- Used when you find 2 or more use cases that have commonalities in behavior, structure and purpose.
- A parent use case may be specialized into 1 or more child use cases that represent more specific form of the parent.

- Neither parent nor child is necessarily abstract, although the parent in most cases is abstract.
- Children of the same parent are all specialization of the parent.
- For example, suppose the ATM system can be used to pay bills. Pay bills has two child use cases : Pay Credit Card Bill and Pay Utility Bill



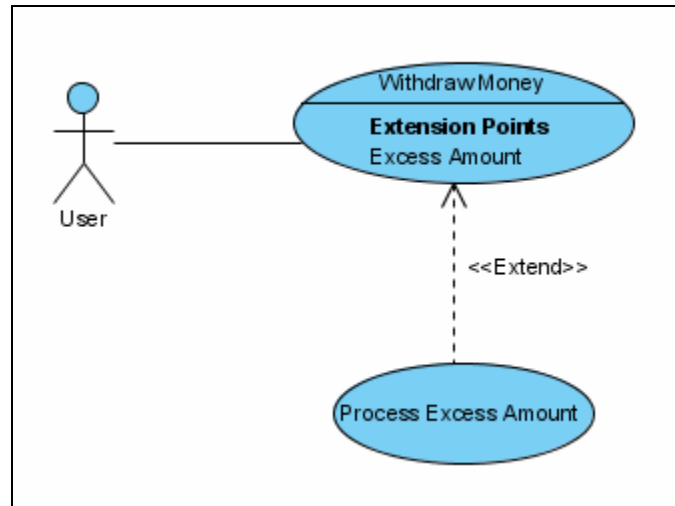
### Include Relationship

- Used when 2 or more use cases share some common portion in a flow of events.
- The common portion is grouped and extracted to form an inclusion use case to be shared among the use cases related.
- Specifies that the source use case explicitly incorporates the behavior of another use case at a location specified by the source.
- For example, in the ATM system example, such as Withdraw Money, Deposit Money or Check Balance, all share the inclusion use case Login Account.



### Extend Relationship

- Extend relationship is used when 2 use cases are similar, but one does a bit more than the other.
- A base case under extend relationship may develop an alternative flow invoking an alternative use case.
- The extension use case adds extra behavior to the base use case.
- Specifies that the target use case extends the behavior of the source.
- For example, Withdraw Money has an optional behavior which handles withdrawal of an excess amount.



A sample extended use case relationship

### In-Class Exercises on <<Include>> and <<Extend>>

- Consider an online flight booking system. A user will search for an available flight time and book the flight if desired. The user will be able to check if there is any seat available, either when searching or booking a flight.
- Consider an online selling / buying website. The website requires the seller and buyer to register their bank account information. A separate process will be called if the bank account is suspended / unavailable / illegitimate.
- Consider a Shop Sales System. A sales officer uses the system to process / record requests of both customer order and faulty goods return. Both processes will need to identify the customer first.
- As part of an online flight booking system, the customer may have the option to upgrade their seat (e.g economy to business class) when making flight booking.
- As part of a Student management system, a student who enrolls in a university will automatically be enrolled in seminars provided by the university, but a security check will be performed if the request to enroll in university is invoked for a new student.

### In-Class Exercises on Generalization

- Consider the Manufacturing Sales System again. The admin or sales officer is required to manage customer goods orderings and requests to return faulty goods. How do you generalize both these use cases into one base case?



- An order management system accepts ordering of product by phone or internet. Write the generalization for this, where that base use case will be used by the order registry clerk.
- To ensure the validity of user of a high security system, 3 behaviors are required to be done on behalf of the user: Password checking, fingerprint scan and retinal scan. Generalize this.

### **DESCRIBING USE CASE (USE CASE DESCRIPTION / SPECIFICATION)**

A use case template captures various pieces of information. A use case is often described in a standard form, using a template similar to the following:

<b>Type</b>	<b>Description / Outline</b>
Use Case Name	Name of the Use Case
Use Case ID	ID of the Use Case
Super Use Case	The name of the generalized use case to which this use case belongs
Actor (s)	The name of the actor(s) who participate in the use case
Brief Description	A description showing how this use case adds value to the organization; that is, what is the purpose or role of this use case in enabling the actors to do their job
Preconditions	The conditions that must be satisfied before this use case can be invoked
Post-Conditions	The conditions that will be established as a result of invoking this use case
Priority	The development priority of this use case
Flow of Events	A step-by-step description of the interactions between the actor(s) and the system, and the functions that must be performed in the specified sequence to achieve a user goal
Alternative Flows and Exceptions	Major alternatives or exceptions that may occur in the flow of events
Non-Behavioral Requirements	The non-functional requirements of the system such as hardware and software platform requirements, performance, security, etc
Assumptions	All the assumptions made about the use case
Issues	All outstanding issues regarding the use case
Source	Reference materials relevant to the use case



<b>Use case name</b>	Place Order
<b>Use case ID</b>	UC-100
<b>Super Use Case</b>	The name of the generalized use case to which this use case belongs.
<b>Actor(s)</b>	Customer Service Assistant
<b>Brief description</b>	A Customer Service Assistant places an order and then submits it for processing.
<b>Preconditions</b>	The member must have registered with the system.
<b>Post-conditions</b>	The Customer's order will be directed to the order process department for processing.
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1. The Customer Service Assistant finds the member's record by entering the member's ID or name. The system displays a list of members that match the information entered by the Customer Service Assistant.</li><li>2. The Customer Service Assistant selects the required member record. The system displays the details of the member.</li><li>3. The Customer Service Assistant selects "Place Order." A new order form and order ID are then generated and displayed.</li><li>4. The Customer Service Assistant selects items from the catalog and adds them to the order.</li><li>5. The Customer Service Assistant submits the order for processing. The system records the order and forwards it to the Order Processing Clerk.</li></ol>
<b>Alternative flows and exceptions</b>	At any time the Customer Service Assistant can decide to suspend the ordering process and come back to it later, or to cancel the order.
<b>Priority</b>	High
<b>Non-behavioral requirements</b>	The system should be able to handle 20,000 new orders per day.
<b>Assumptions</b>	
<b>Issues</b>	Is there any limit on the amount of an order?
<b>Source</b>	User Interview Memo 21, 8/9/01

Sample Use Case Description for use case Place Order



## **GUIDE TO USE CASE MODELLING**

\*Goods = product

To illustrate how a software development project is initiated / conducted using UML Use Case diagram, a sample problem statement is introduced below concerning a system that handles goods\* management for customers by mail.

### **A Sample Problem Statement example for the Mail Order System:**

In order to improve the operational efficiency of a mail order company, the chief executive officer is interested in computerizing the company's business process. The major business activities of the company can be briefly described as follows:

A customer registers as a member by filling in the membership form and mailing it to the company. A member who has not been active (no transactions made) for a period of one year will be removed from the membership list and he/she needs to re-apply for the reinstatement of the lapsed membership.

A member should inform the company of any changes of personal details such as home address, telephone numbers, etc. A member can place an order by filling out a sales order form and faxing it to the company or by phoning the Customer Service Assistant with the order details.

The Customer Service Assistant first checks for the validity of membership and enters the sales order information into the system.

The Order Processing Clerk checks the availability of the ordered items and holds them for the order. When all the ordered items are available, he/she will schedule their delivery.

The Inventory Control Clerk controls and maintains an appropriate level of stock and is also responsible for acquiring new items.

If there is a problem with an order, members will phone the Customer Service Assistant. The Customer Service Assistant will take appropriate action to follow up the sales order.

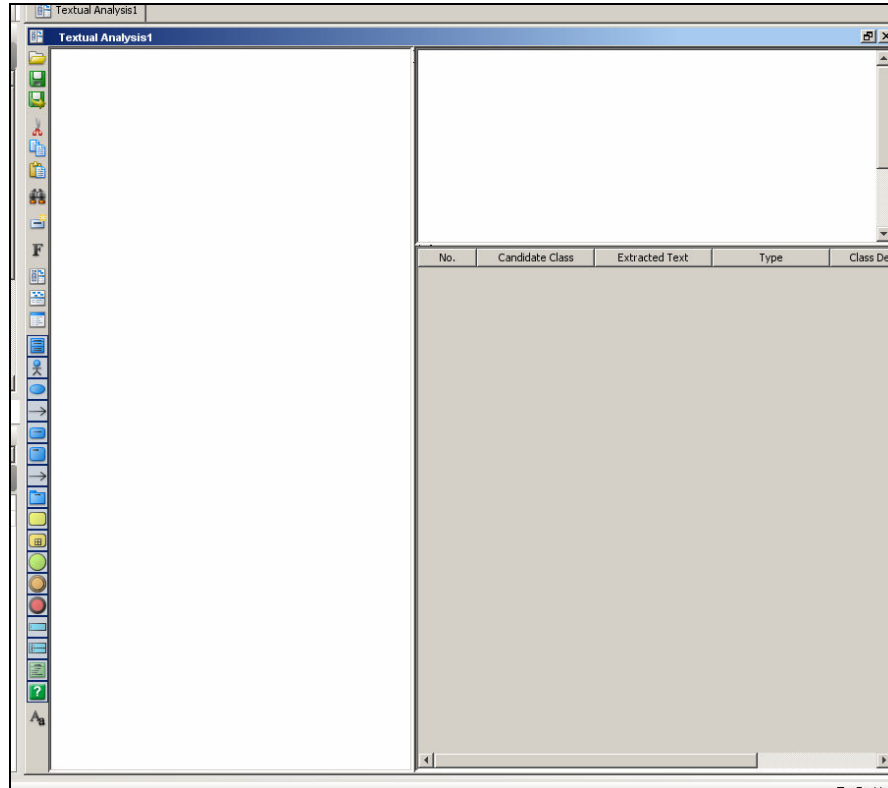
Members may return defective goods within 30 days and get their money back.

The system will record the name of the staff member who has initialized an updated transaction to the system.

### **Step 1: Prepare the Problem Statement**

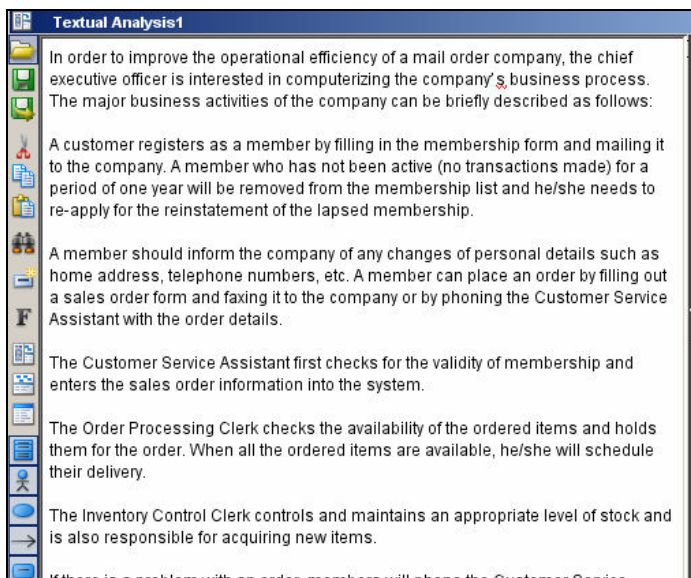
The problem statement is prepared through interviews with the stakeholders of the system. The problem statement can be entered into the VP-UML case tool for further requirement analysis. Simply follow the steps below.

- Create a new Textual Analysis by clicking on the "New Textual Analysis" on the application toolbar.



Textual Analysis working area

- Type in the problem statement from the Mail Order System sample problem statement into the textual analysis text box.



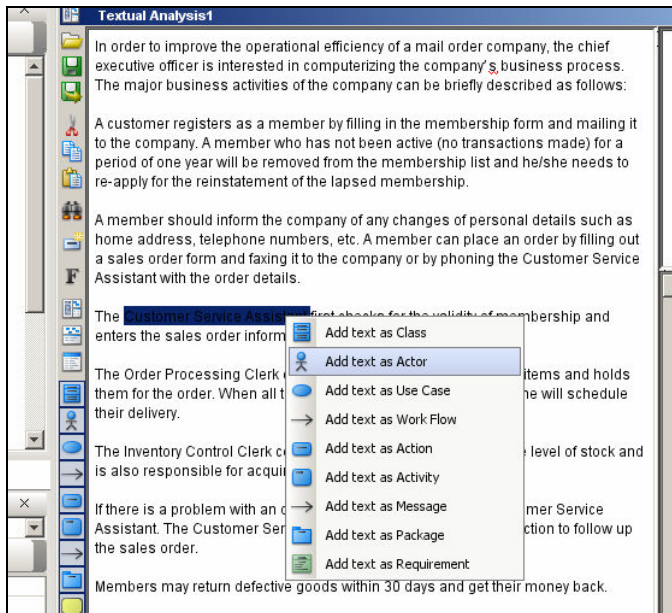
Entering problem statement for Textual Analysis



## Step 2: Identify Major Actor(s)

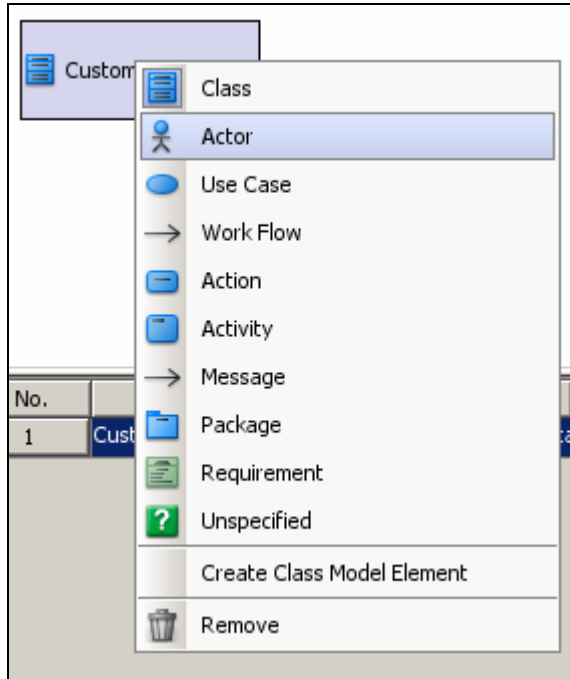
Once the problem statement is in the case tool, the next step is to identify actors in the **Textual Analysis** working area.

- Highlight the phrase “Customer Service Assistant” in the problem statement as a candidate actor and drag it to the **Candidate Class Container** at the top right-hand corner. Note that all occurrences of the same actor in the problem statement are automatically highlighted.

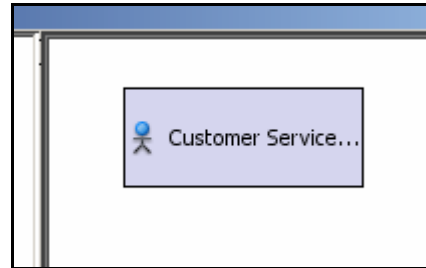


Identifying major actors

- Now “right click” on the newly created candidate class in the **Candidate Class Container**. A pop-up menu will appear > Select the **Actor** option in the pop-up menu to declare the candidate class as an actor.  
Note that the icon of the candidate class in the **Candidate Class Container** has changed from class to actor and the type of the candidate class has also changed to **Actor**.

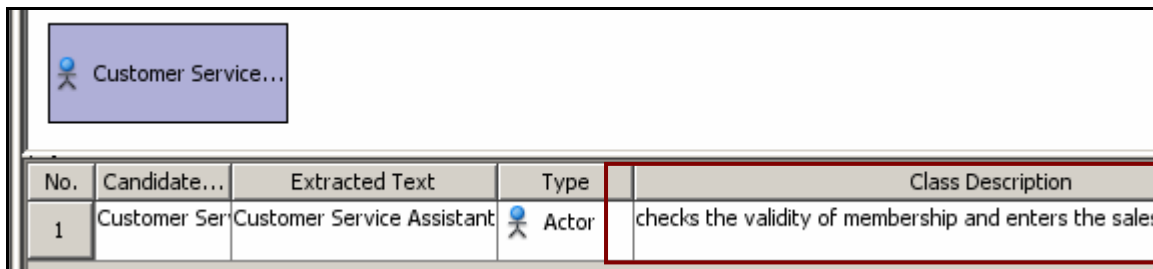


Defining actor type



Candidate Actor in Candidate Class Container

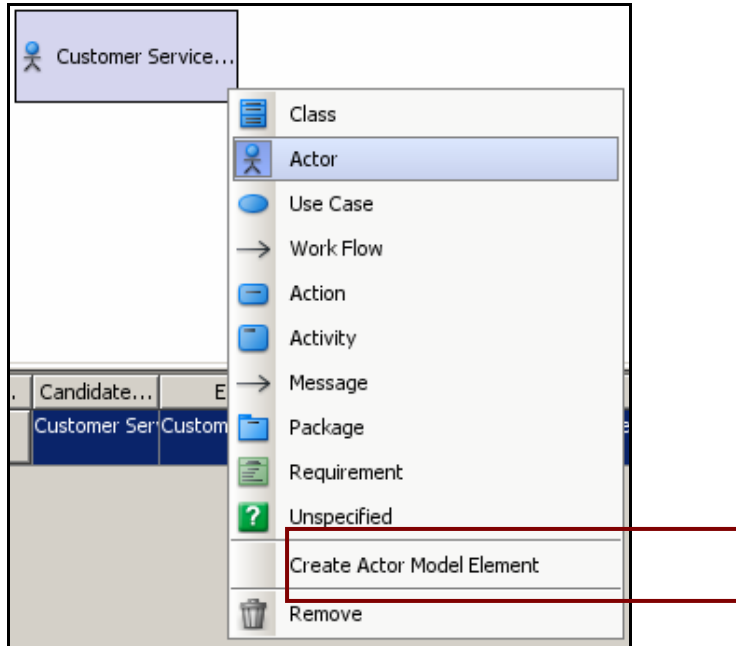
- To enter the description of an actor:
  - Go to the **Class Description** cell under the multi line box in the bottom right corner.
  - Type in a brief description such as the task(s) performed by the actor.



Entering actor description in Class Description

The candidate actors can be added into the model repository. Elements in the model repository can be retrieved for later use, e.g. to draw a use case diagram.

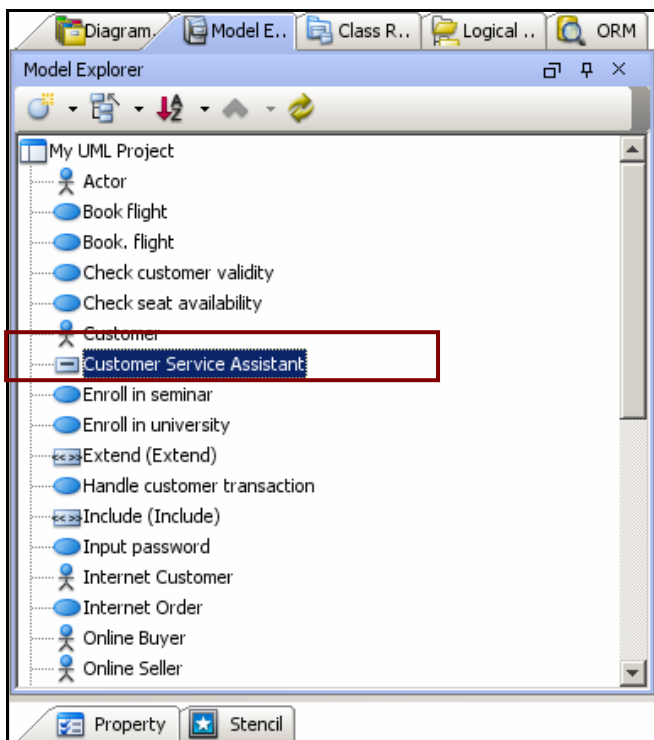
- To add Customer Service Assistant (candidate actor) into the model repository, “right click” on the Customer Service Assistant.
- A pop-up menu will appear. Select **Create Actor Model** in the pop-up menu



Creating an Actor in model repository

The Customer Service Assistant is now added to the **Model Repository Tree**.

- To see the newly created actor model, click on the **Model Explorer** tab in the **Project Explorer Pane** (shown below).




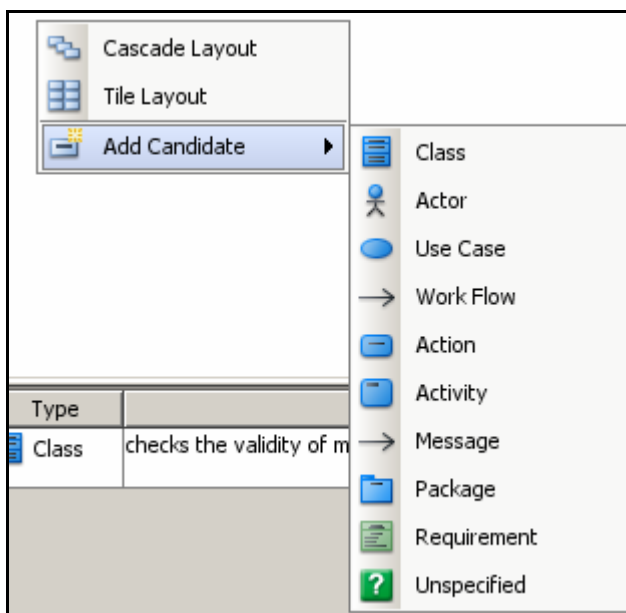
Customer Service Assistant actor in Model Repository Tree

- Repeat the above steps to identify and create actor models for the actors:
  - Order Processing Clerk
  - Inventory Control Clerk

### Step 3: Identify Use Cases

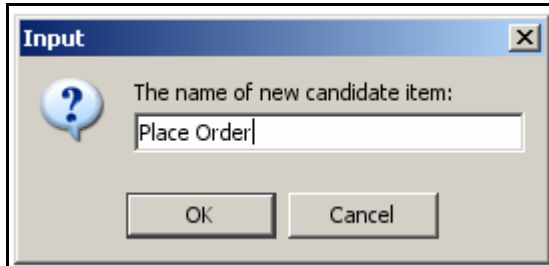
Let us identify a candidate use case from the problem statement. It is necessary to read through the text carefully to identify a use case. Follow the steps below to create a use case directly from the **Candidate Class Container**.

- To hide the actors in the **Candidate Class Container**, click on the **Show Candidate Actors** toggle button  in the **Textual Analysis toolbar**. However, note that the actor models still exist in the **Model Repository Tree**
- “Right click” on the **Candidate Class Container** > A pop-up menu will appear. Then select **Add Candidate** in the pop-up menu; a cascading menu will appear > Select **Use Case** in the cascading menu.



Creating a candidate use case in Candidate Class Container

- An input dialog will appear. Enter Place Order in the input dialog

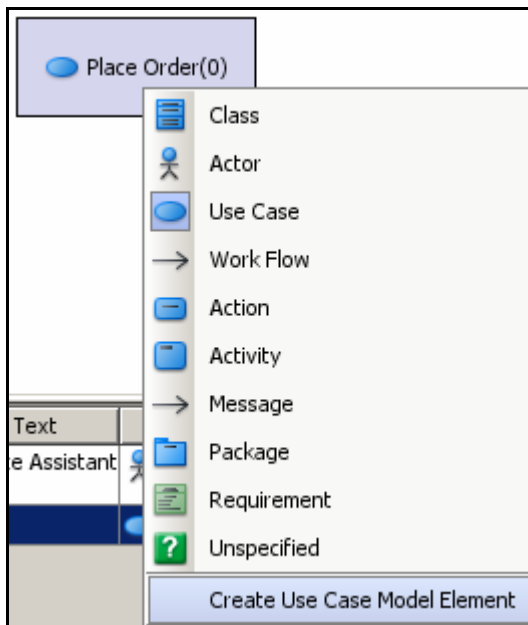


Naming a new candidate use case

- “Click **OK** in the input dialog”. A new candidate use case is then created in the **Candidate Class Container**

Now edit the use case brief description for the candidate use case the same way as you would edit the actor.

- To add a candidate use case into the model repository, “right click” on the desired candidate use case in the **Candidate Class Container**. A pop-up menu will appear > Select **Create Use Case Model**



A new use case in Model Repository Tree

A new use case is added to the Model Repository Tree.

- Repeat the above steps to identify all other candidate use cases below:
  - Check Order Status
  - Handle Goods Return
  - Update Membership Record
  - Archive Membership

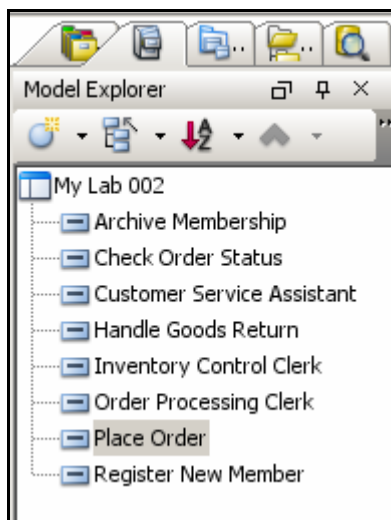


- Register New Member
- Process Order
- Schedule Delivery
- Order Goods
- Receive Goods
- Deliver Goods

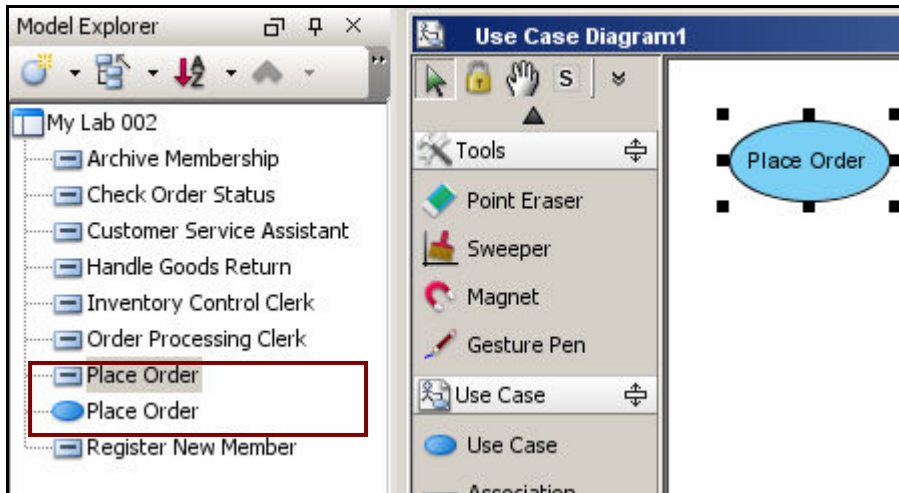
#### Step 4: Create Initial Use Case Diagram

Having identified all the use cases, create the use case diagrams with the case tool following the steps below:

- Create a new Use Case Diagram.
- Click on the “Model” tab in the “Project Explorer”. A list of model elements will be presented.
- Select the use case Place Order from the **Model Explorer** and drag it to the desired location in the diagram pane. A use case is automatically placed in the diagram with the name Place Order

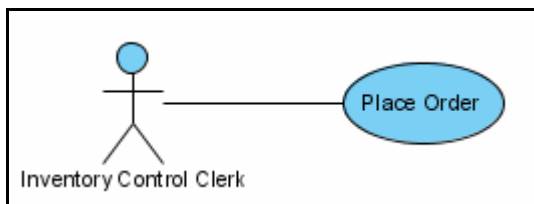


Models in Model Repository Tree



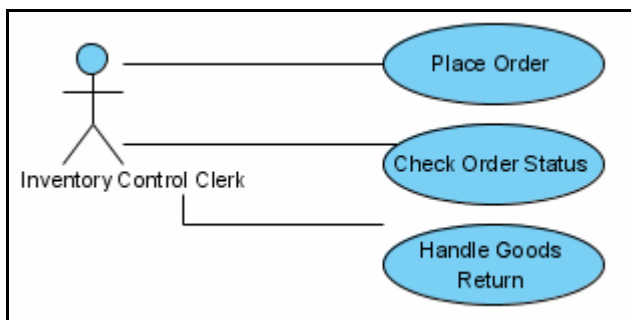
Use case Place Order is automatically created / listed in the model explorer

- Select Customer Service Assistant from the **Model Explorer** and drag it to the desired location in the diagram pane. An actor is then placed in the diagram with the name Customer Service.
- Associate the actor and use case displayed in the diagram.



Creating an association relationship using resource-centric icon

- Repeat the above steps to create the following use cases and their association relationships with the Customer Service Assistant actor:
  - Check Order Status
  - Handle Goods Return

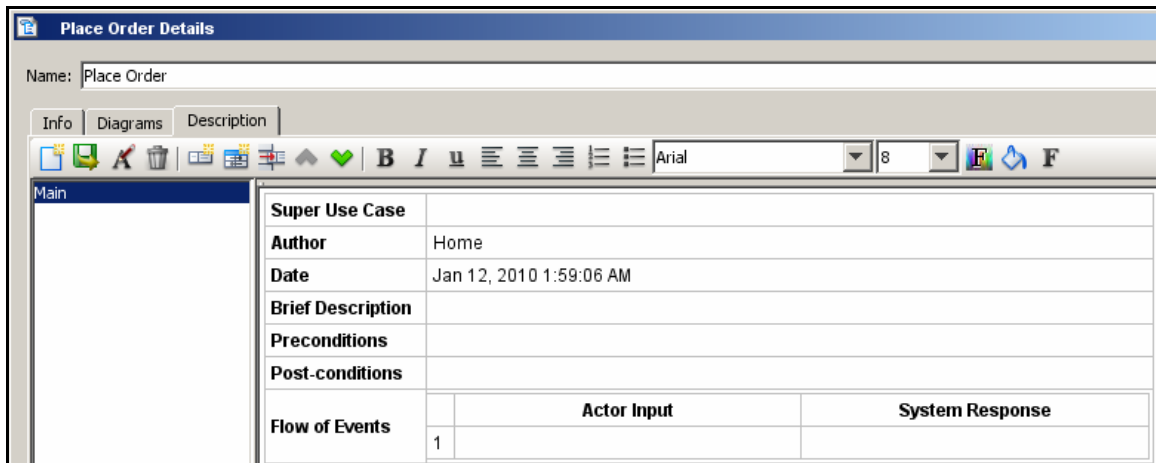


Creating more use cases with Model Repository Tree

## Step 5: Describe Use Cases

The use cases created require further elaboration so that the next phase of the analysis can be performed. Follow the steps below:

- “Right click” the use case Place Order, and choose “Use Case Details”.
- Select the “Description tab”. A Specification Dialog about the files associated to the element will be displayed.



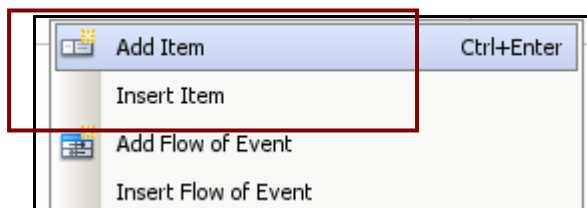
The screenshot shows a dialog box titled "Place Order Details" with a "Name" field containing "Place Order". It has three tabs: "Info", "Diagrams", and "Description". The "Description" tab is active, showing a rich text editor with a toolbar. Below the editor is a table for "Flow of Events".

Super Use Case		
Author	Home	
Date	Jan 12, 2010 1:59:06 AM	
Brief Description		
Preconditions		
Post-conditions		
Flow of Events		
	Actor Input	System Response
1		

Use case specification and template

- Enter the contents for each of the elements in the use case template and click on the **OK** button to confirm the use case description.

You can also add / insert additional heading / element / texts by right clicking anywhere in the Description window.



- Try to key in the full use case description for (use case) “Place Order” as given in the table earlier (page 6) and try to translate its flow of event in terms of Actor Input and System Response (2 columns format).

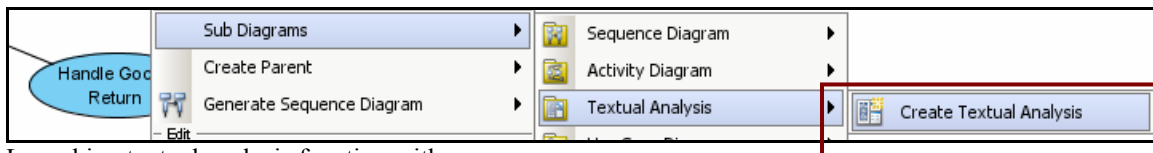


## Step 6: Perform Textual Analysis

A textual analysis needs to be performed for each of the use cases based on their descriptions; this will yield a set of candidate classes in the process. These classes will then be considered for inclusion in the domain class model which serves as a preliminary class model for the future development of the initial class model.

Now, let us perform a textual analysis on the Schedule Delivery use case.

- “Right click” the use case Schedule Delivery and then > select Sub Diagrams > select **Create Textual Analysis** from the pop-up menu.



Launching textual analysis function with a use case

- A new **Textual Analysis** window will appear. Enter the following text in the text pane:

The Order Processing Clerk selects an order from the list of filled sales orders. The system displays the sales order details, together with the member telephone number and address. The Order Processing Clerk enters the delivery date and time after talking with the member over the phone. The system records the delivery date and time of the sales order. The system records the name of the Order Processing Clerk who has handled the sales order.

- Now highlight the word order as a candidate class, right click on the word order, and select **Add Text as Class** in the pop-up menu.  
[Note all occurrences of the same actor in the problem statement are now automatically highlighted.]
- A new candidate class is automatically created in the **Candidate Class Container** on the right-hand side and all occurrences of the same class in the problem statement are automatically highlighted.
- Select the **Class Description cell** next to the class Order. Enter a brief description about the Order class.




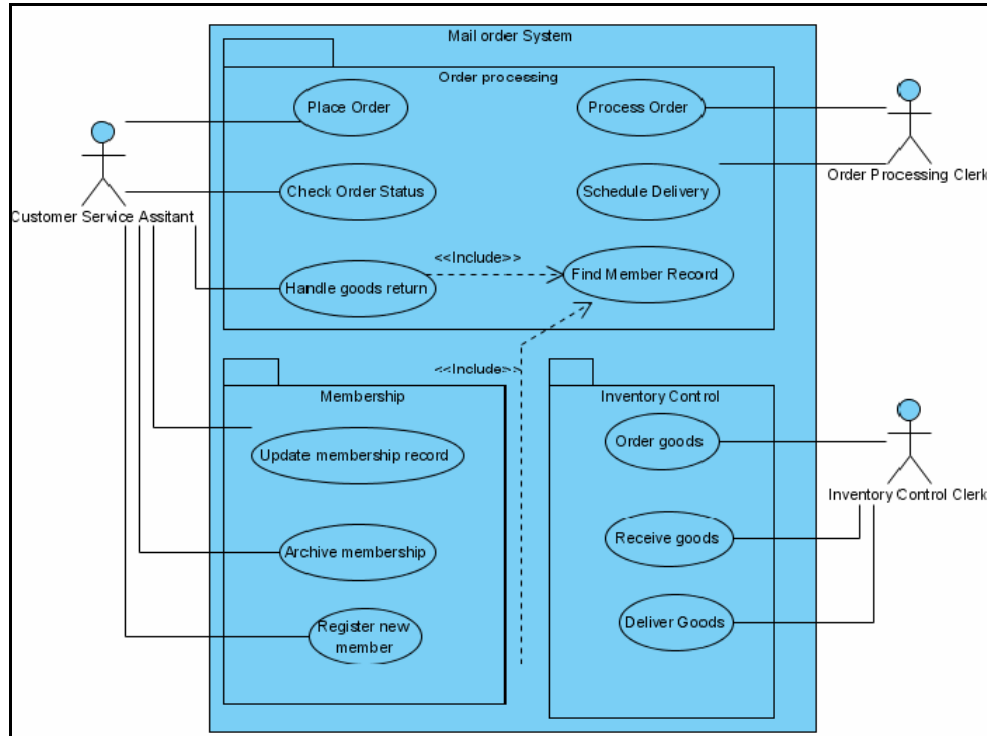
- Repeat the above steps to create the following classes:
  - Sales Order
  - Member
  - Delivery

### Step 7: Structuring Use Cases with System Boundary

Use cases with similar domain can actually be grouped into packages.

In the Mail Order System example, we can identify three packages which are Inventory, Membership and Order Processing which are associated with the actors. The ultimate goal is to organize the use cases into packages to maximize consistency within the individual packages and minimize coupling among these packages.

- Create a package by first clicking on the **Use Case Diagram** palette.
- Place the mouse pointer in the design area and click once. A package symbol will then appear in the design area. Rename the new package as Inventory Control. Press Ctrl \_ Enter to finish the operation.
- Move each of the use cases by dragging them into the package region where it belongs or where it is a member.
- Add the system boundary to the use case model by clicking  on the **Use Case Diagram** palette and move the mouse pointer to the desired location on the diagram pane.



Complete use case model structured into packages