

# Electronic Commerce Architecture Project

## LAB TWO: Introduction to UML

### Introduction to UML. by Lawrence Leff

UML stands for Unified Modeling Language. It is a visual technique for diagramming the requirements, actions and physical distribution of software systems. (It might also be used for diagramming manual flows such as paper handling in a court or hardware.)

In the mid 1990's, there were several popular object-oriented diagramming methods. Fortunately, each of their proponents agreed to develop a new system and standard that has the best qualities of all of them. I list below, the UML books that would be most helpful to learn about it.

The following are the important types of diagrams:

### Use Cases

These represent the types of people (roles) that will be using the software being developed or documented. It also indicates what they will be doing. The labs describe more material.

### Association Diagrams

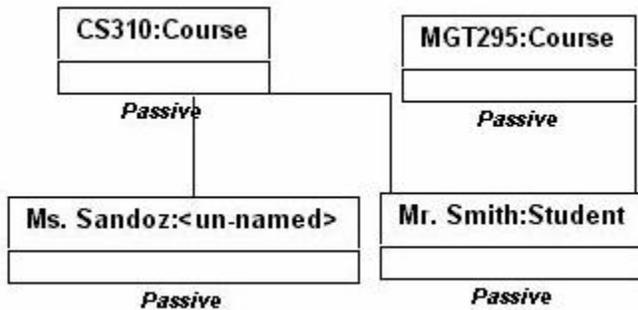
These represent the data or objects in the system and their relationships. For example, a school registrar system would show courses and students, and the fact that the software can determine from a student, which courses they are taken. And, given a course, it can generate a list of students. It would show that each course could have several students and each student might be taking several courses.

These serve the same purpose as entity relationship diagrams, a technique developed by the database community, which you can see inside of Microsoft Access. A simple association diagram showing the relationship between course and student follows.

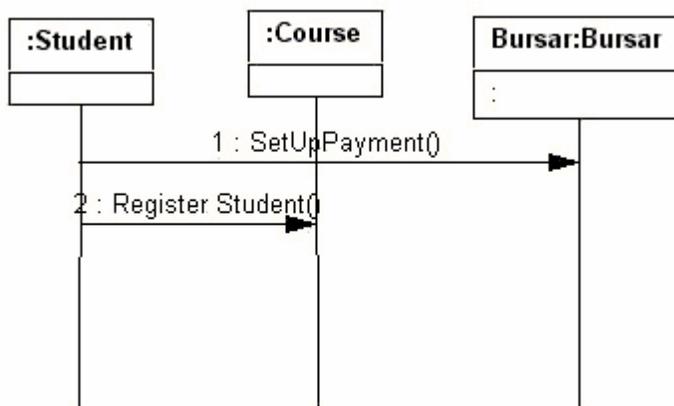


## Collaboration and Sequence Diagrams

These diagrams show the time dependent behavior of a system. Unlike association diagrams, they are intended for showing specific examples. When, we describe an operation of a system, one might give a verbal description of a situation: Mr. Smith registered for course CS310 and MGT295 and Ms. Sandoz registered for CS310. One might draw a collaboration diagram showing the connections between the courses and people to illustrate the verbal description of the example below.



Collaboration diagrams and sequence diagrams are used to show time-dependent changes as the software is used. In a University environment, it might show that the payment activities associated with registration would occur before a student is associated, i.e., registered, for a course. Collaboration diagrams and sequence diagrams show the same sequencing activities. However, the two forms of diagrams are visually very different. The information shown is the same--Rational Rose has a feature to automatically convert a collaboration diagram to a sequence diagram and vice versa.



# Component and Deployment Diagrams

Both of these diagrams are specific to the collection of the basic parts of programs (a single file or program) and how they are grouped in modules by linkage editing as well as how the different programs would run on different servers and computers. Needless, to say, these are used by those concerned with the implementation of the software, i.e., computer programmers and software engineers. Component diagrams represent the collection of the individual programs such as pieces of Java or C into object modules and executables. Deployment diagrams show the computers that compose a distributed system, which executables run on which communication, and the communication pathways between them.

## Laboratory Procedure

The first part of the lab shows you how to draw UML use cases with the Adobe PageMaker in the School of Architecture and Planning Computerized Classroom. The second part of the lab shows you how to use GDPRO to do UML use cases. You can download this to your personal computer from GDPRO.com. .LP we also hope to obtain Rational Software's UML product, Rational ROSE. At the time of writing this lab, Rational has failed to send it to us.

Ms. Goel set up some figures for the standard forms that appear in a UML Use case diagram. We will copy them first; then use them to create an example of diagrams that might appear in situation.

The oval represents a "use case." This is simply a scenario of activities for a common purpose. An example would be the sequence of screens to add or drop a course.

The stick figures represent actors which are people in some role. For example, a person might be in the role of student when they add a course. Note, that occasionally, the same human might serve two different roles. A professor in one department might happen to take a course elsewhere at the University. We would consider that person two different actors, depending upon their role.

When a user activates a scenario, we show with an arrow with an open head.

Occasionally, one might have non-human act's that have inputs and/or outputs from your system. These represent software or hardware not part of the system you are designing. For example, in designing the software in a burglar alarm control box, the telephone system by which it communicates and the motion sensors would be non-human actors. If one is designing a payroll system, the Internal Revenue Service computer to which one sends W-4's would be a non-human actor. All non-human actors are drawn as rectangles.

Often, several activities or scenarios have some common sequence. For example, whether a student is adding or dropping a course, they still enter their login and password. To

illustrate a subsequence of activities, one draws an open-head arrow between the oval representing one use case and the one representing the common subsequence. This is called an extends relationship.

We can show generalizations between categories of actors with the open arrow. Graduate students are a type of student. Managers and programmers are both types of employee, etc. Of course a subcategory can implement all the scenarios that the general ones can. For example, if all employees in the firm can check their paycheck status, then so can managers and programmers.

One draws an open arrow from the more specific type of actor to the more general category. Thus, in the example below, we draw an open-headed arrow from student to graduate student.

### **Procedure to use Adobe PageMaker to do UML use cases**

1. Log into the one of the computers in the School of Architecture and Planning computerized classroom
2. Select **Network Neighborhood**
3. Select **Archfile** and **Archfile01**
4. Select **a\_goel**
5. Select **public\_html** and **Summer**
6. Drag the **UML-IMAGES** folder to your desktop.
7. (You can now close all the directories that were just opened.)
8. Open the **UML-IMAGES** directory. Copy the **Test** item to the name **LABTWO**. (This will give you a copy of the images to work with when you make your diagram, leaving the original set to copy again for a homework assignment.)
9. Open **LabTWO**, the image will reopen.  
(*You will now see the figures of which a standard UML use case diagram is composed.*)
10. We will make a use case diagram for the following scenario:
  - Any student performs the following actions,
    1. registers for classes (adds)
    2. drops classes
    3. Changes to pass fail status

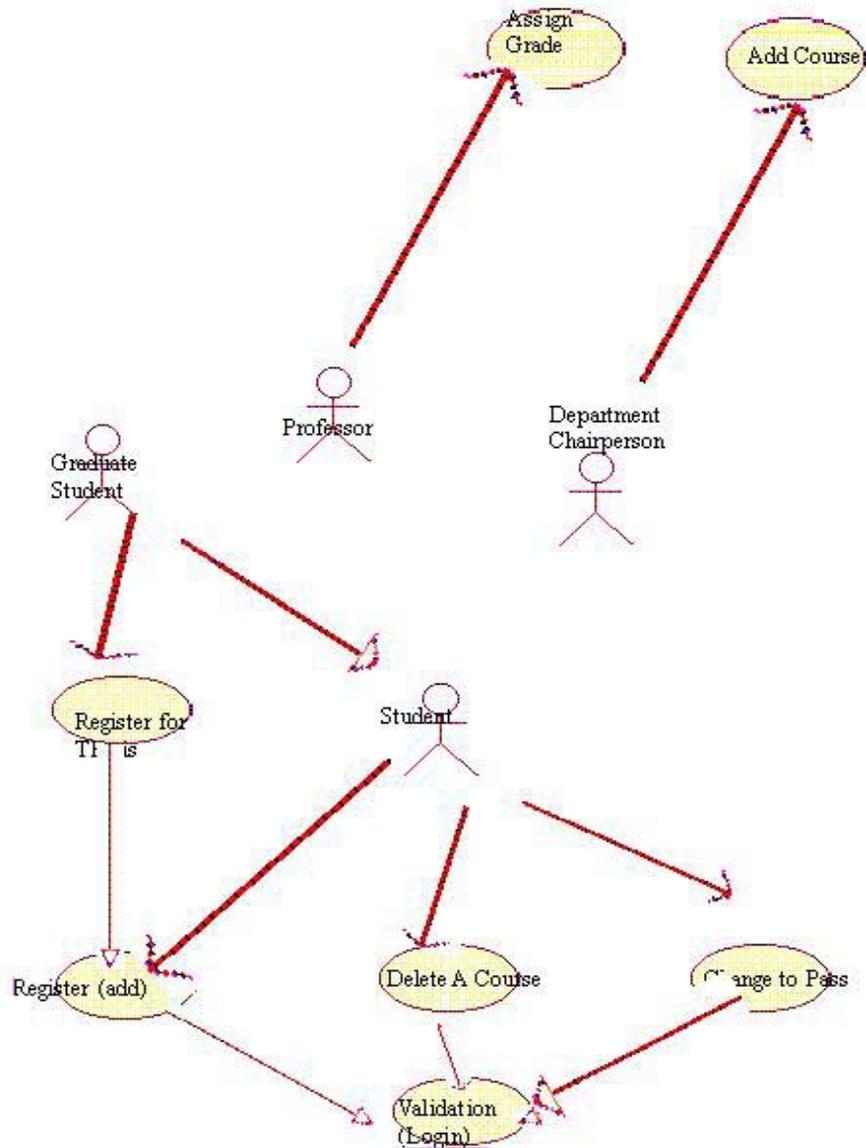
All of these use cases require a validation process (logging in)

- Professors assign grades
- Department Chairperson's
  1. create classes
  2. assign professor to class

There is one more wrinkle. Graduate students are a type of student. They register for thesis. This involves (extends) the process of registering or adding for class. To think

about this in the right way, a graduate student registering for thesis has to log in and enter a class number, etc. just like any class. However, they have additional actions like choosing an advisor and typing in the name of thesis, abstract, and expected lab materials budget.

Your image should look something like below. I suspect that your work will be neater than mine.



## Procedure to Download and Install GDPRO

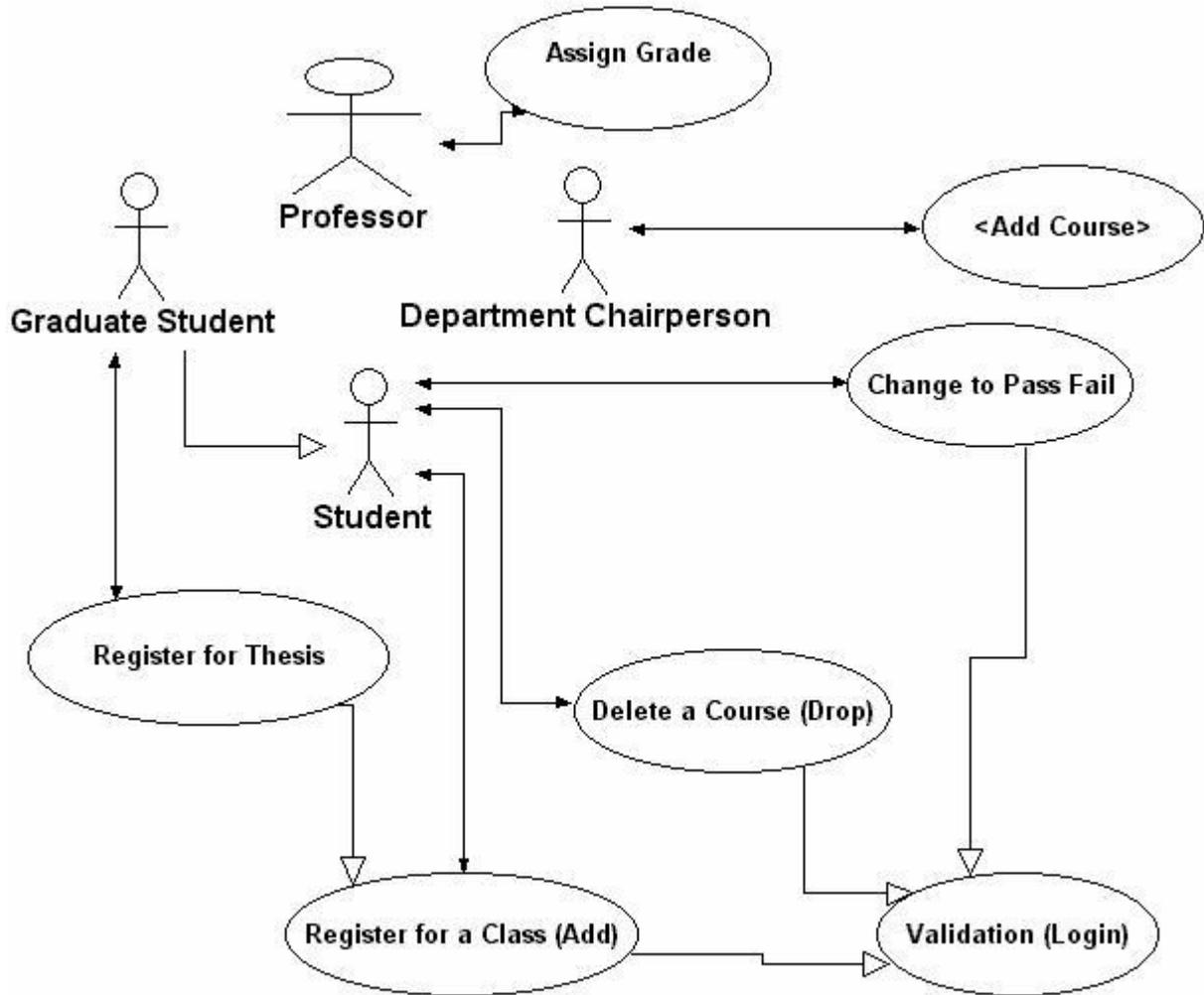
1. Go to [www.gdpro.com](http://www.gdpro.com) on your browser.
2. Select **evaluate**
3. Enter the information on your form and click on **Submit Completed Agreement**
4. Select "download via ftp" under GDPro 4.1 for Windows (43MB)
5. Select "Save this program to disk" and choose allocation to put the download compressed executable. (This will be a temporary file that will be expanded later.)
6. Execute that file. You can do that by going to MS-DOS. Changing to the directory in the previous step and typing **GDPRO\_~1.exe**.
7. Click **Next** and **OK**
8. Click **Yes** on the Software License Agreement.
9. Enter your name and company into the "User Information" dialogue. Select **Next**.
10. Be sure that **GDPRO Client** is checked. The other two will not be checked. Click on **Next**
11. Click on **Next** on the **Choose Destination Path**. (It is possible; you may have to change directories if you don't have room on your C: drive.)
12. Click on **Next** when asked about accessing across a network.
13. Click on **Next** on the Select Program Folders dialogue.
14. Click on **Next** on the Start Copying Files dialogue. The system should start the installation process.
15. Click on **OK** twice.
16. You will be asked about restarting your computer. To be absolutely safe, go ahead and restart. I think it will work if you don't restart.

## Procedure to Use GDPRO

1. GDPRO should be available as GDPRO on the **Start** and **Programs** menu. Select it.
2. Select GDPRO on the sub menu.
3. You will be told how many days you have left on the evaluation copy. Click on **OK**.
4. Leave the user name as Guest on the menu with the big GD PRO 4.1 on a picture. Click on **OK**.
5. Click on **Close** on the Next Tip dialogue box.
6. Click on **File** and **New System**
7. Enter a system name and click **OK**. All your diagrams about a particular software system are kept in one GDPRO "system."
8. Enter a diagram **Name** and be sure that the **Diagram Type** says **Use Case Diagram**. Click **OK**.
9. The middle column should contain some diagrammatic figures that would appear on use case diagrams. Click on the Actor button. Drag over four actors.
10. Click on the leftward pointing arrow (the selection tool). To change the label on an actor, right click on it. Select **Properties**. Enter the new name and press the **ENTER** key.

11. Label your actors as follows:
  - Professor
  - Graduate Student
  - Student
  - Department Chairperson
12. Find the button with two actors, an arrow, and the label "AL"  
Click on it.
13. Connect from Graduate Student to Student. An open-ended arrow will be created between these two actors. (This represents that Student is a generalization of Graduate Student.)
14. Click on the Use Case button at the top of the middle bar. Drag over seven Use case ovals.
15. Click on the arrow (selection tool)
16. To change the label on a use case, right click on it, select **Properties**, change the "name" field, and press the **OK** button.  
Change your use case labels as follows:
  1. Register for Thesis
    - 2. Register for a Class (Add)
    - 3. Delete a Course (Drop)
    - 4. Validation (Login)
    - 5. Change to Pass Fail
    - 6. Add Course
    - 7. Assign Grade
17. Click on the button that has two blue ovals, an open-headed arrow and UL.
18. Connect "Register for Thesis" to "Register for a class" (This shows that register for thesis uses the steps for registering for a class in its scenario.) In other words, when one registers for thesis, one also does the same steps that one would when registering for a class. However, registering for a thesis would include some additional actions.
19. Also, connect "Change to Pass Fail" to "Validation(Login),  
"Delete a Course (Drop)" to "Validation(Login),  
"Register for a Class (Add)" to "Validation (Login)
20. Now click on the button with an actor, a blue oval, and a "CL" label.
21. Use it to connect:
  0. "Student" to "Register for a Class (Add)"
  1. "Student" to "Delete a Course (Drop)"
  2. "Student" to "Change to Pass Fail"
  3. "Graduate Student" to "Register for Thesis"
  4. "Professor" to "Assign Grade"
  5. "Department Chairperson" to "Add Course"
22. To save in JPEG format (e.g., to put on your web page), click on **File** and **Export**. Be sure that **JPEG** is chosen and click on **OK**. Select a directory and file for your JPEG image.
23. Click on **File** and **Exit** to save.

Your picture should look as follows:



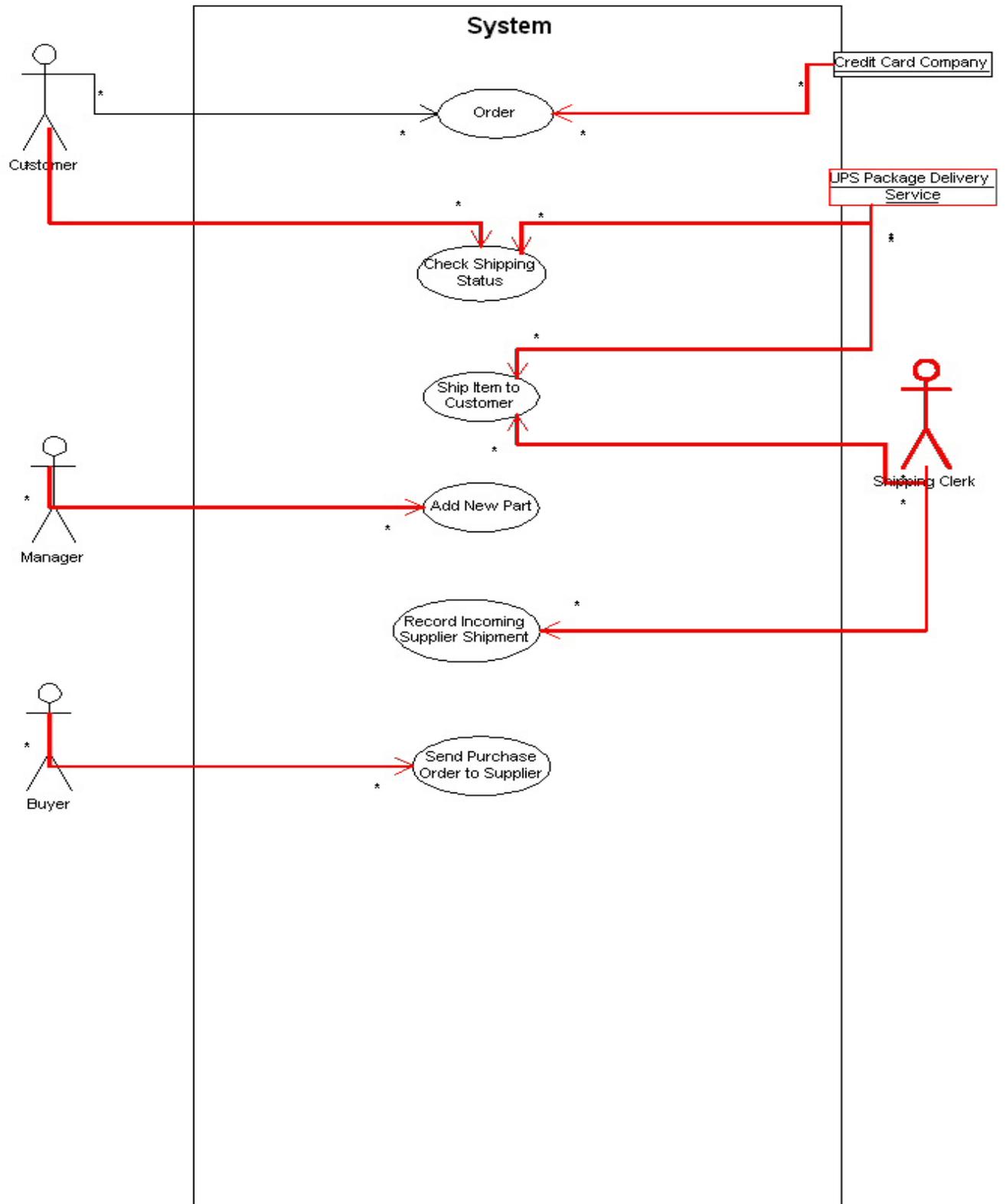
## Another Example

We design a set of use cases for a commercial wholesaler supporting electronic commerce with its customers. A purchaser (customer) will make orders. Making an order will involve checking the credit card of the customer and it communicates with the non-human actor (the credit card company, or more precisely its computer systems). From time to time, customers will check the shipping status.

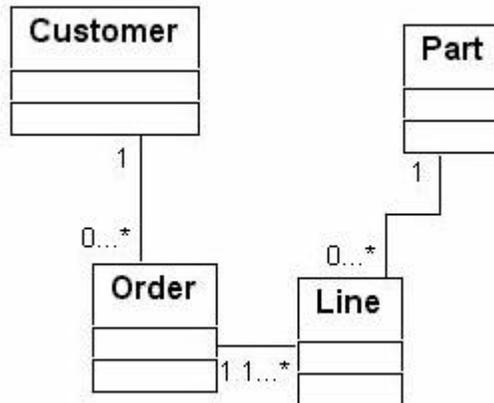
The shipping clerk is responsible for preparing the outgoing order. This involves a communication with UPS to register the outgoing item. UPS is thus shown communicating with "Ship Item to Customer" as well as "Check Shipping Status."

The manager adds new parts that the firm decides to sell.

The shipping clerk is also responsible for receiving incoming shipments. We also show that the buyer is responsible for ordering new quantities of parts when they run out.



The association diagram represents the data and information that has to be stored. Each customer will have one or more orders. Each order has one or more lines. (A line in order represents the fact that a customer ordered a particular quantity of a particular part.) Each



line is connected to one or more parts.

## References

1. Booch, Grady, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
2. Pooley, R. J., and Perdita Stevens. *Using UML : software engineering with objects and components*. Addison-Wesley, 1999
3. Quatrani, Terry, and Grady Booch. "Visual Modeling with Rational Rose 2000 and UML." *The Addison-Wesley Object Technology Series*. Addison Wesley, 1999.