

Experiment no 3

Study of Use Case Diagram

Objective-: To study Use Case Diagram in Rational Rose.

References-:

- www.developer.com
- The Unified Modeling Language User Guide by Grady Booch
- Mastering UML with Rational Rose 2002 by Wendy Boggs

Theory-:

The Use case diagram is used to identify the primary elements and processes that form the system. The primary elements are termed as "actors" and the processes are called "use cases." The Use case diagram shows which actors interact with each use case.

The above statement pretty much sums up what a use case diagram is primarily made up of—actors and use cases.

A use case diagram captures the functional aspects of a system. More specifically, it captures the business processes carried out in the system. As you discuss the functionality and processes of the system, you discover significant characteristics of the system that you model in the use case diagram.

So who should normally be involved in the creation of use cases? Normally, domain experts and business analysts should be involved in writing use cases for a given system. Use cases are created when the requirements of a system need to be captured. Because, at this point no design or development activities are involved,

technical experts should not be a part of the team responsible for creating use cases. Their expertise comes in use later in the software lifecycle.

✚ Elements of a UML Use Case Diagram

A use case diagram is quite simple in nature and depicts two types of elements: one representing the business processes and the other representing the business roles.

Take a closer look at what elements constitute a use case diagram:

- **Actors:** An actor portrays any entity (or entities) that performs certain roles in a given system. The different roles the actor represents are the actual business roles of users in a given system. An actor in a use case diagram interacts with a use case. For example, for modeling a banking application, a customer entity represents an actor in the application. Similarly, the person who provides service at the counter is also an actor. But it is up to you to consider what actors make an impact on the functionality that you want to model. If an entity does not affect a certain piece of functionality that you are modeling, it makes no sense to represent it as an actor. An actor is shown as a stick figure in a use case diagram depicted "outside" the system boundary, as shown in below figure.



To identify an actor, search in the problem statement for business terms that portray roles in the system. For example, in the statement "patients visit the doctor in the clinic for medical tests," "doctor" and "patients" are the business roles and can be easily identified as actors in the system.

Use case: A use case in a use case diagram is a visual representation of a distinct business functionality in a system. The key term here is "distinct business functionality." To choose a business process as a likely candidate for modeling as a use

case, you need to ensure that the business process is discrete in nature. As the first step in identifying use cases, you should list the discrete business functions in your problem statement. Each of these business functions can be classified as a potential use case. Remember that identifying use cases is a discovery rather than a creation. As business functionality becomes clearer, the underlying use cases become more easily evident. A use case is shown as an ellipse in a use case diagram as shown in figure below-:



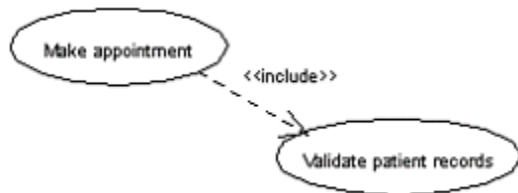
Figure shows two uses cases: "Make appointment" and "Perform medical tests" in the use case diagram of a clinic system. As another example, consider that a business process such as "manage patient records" can in turn have sub-processes like "manage patient's personal information" and "manage patient's medical information." Discovering such implicit use cases is possible only with a thorough understanding of all the business processes of the system through discussions with potential users of the system and relevant domain knowledge.

- **Relationships in UML Use Cases**

UML use cases share different kinds of relationships. A relationship between two use cases is basically a dependency between the two use cases. Defining a relationship between two use cases is the decision of the modeler of the use case diagram. This reuse of an existing use case using different types of relationships reduces the overall effort required in defining use cases in a system. A similar reuse established using relationships, will be apparent in the other UML diagrams as well. Use case relationships can be one of the following:

- **Include:** When a use case is depicted as using the functionality of another use case in a diagram, this relationship between the use cases is named as an

include relationship. Literally speaking, in an *include* relationship, a use case includes the functionality described in the another use case as a part of its business process flow. An include relationship is depicted with a directed arrow having a dotted shaft. The tip of the arrowhead points to the parent use case and the child use case is connected at the base of the arrow. The stereotype "`<<include>>`" identifies the relationship as an include relationship.



- Extend: In an extend relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case. An extend relationship is depicted with a directed arrow having a dotted shaft, similar to the include relationship. The tip of the arrowhead points to the parent use case and the child use case is connected at the base of the arrow. The stereotype "`<<extend>>`" identifies the relationship as an extend relationship, as shown in Figure 3.5.



Figure 3.5: an example of an extend relationship

Figure 3.5 shows an example of an extend relationship between the "Perform medical tests" (parent) and "Perform Pathological Tests" (child) use cases. The "Perform Pathological Tests" use case enhances the functionality of the "Perform medical tests" use case. Essentially, the "Perform Pathological Tests" use case is a specialized version of the generic "Perform medical tests" use case.

- Generalizations: A generalization relationship is also a parent-child relationship between use cases. The child use case in the generalization relationship has the underlying business process meaning, but is an enhancement of the parent use case. In a use case diagram, generalization is shown as a directed arrow with a triangle arrowhead (see Figure 3.6). The child use case is connected at the base of the arrow. The tip of the arrow is connected to the parent use case.



Figure 3.6: an example of a generalization relationship

On the face of it, both generalizations and extends appear to be more or less similar. But there is a subtle difference between a generalization relationship and an extend relationship. When you establish a generalization relationship between use cases, this implies that the parent use case can be replaced by the child use case without breaking the business flow. On the other hand, an extend relationship between use cases implies that the child use case enhances the functionality of the parent use case into a specialized functionality. The parent use case in an extend relationship cannot be replaced by the child use case.

An example should help you understand things better. From the diagram of a generalization relationship (refer to Figure 3.6), you can see that "Store patient records (paper file)" (parent) use case is depicted as a generalized version of the "Store patient records (computerized file)" (child) use case. Defining a generalization relationship

between the two implies that you can replace any occurrence of the "Store patient records (paper file)" use case in the business flow of your system with the "Store patient records (computerized file)" use case without impacting any business flow. This would mean that in future you might choose to store patient records in a computerized file instead of as paper documents without impacting other business actions.

Now, if you had defined this as an extend relationship between the two use cases, this would imply that the "Store patient records (computerized file)" use case is a specialized version of the "Store patient records (paper file)" use case. Hence, you would not be able to seamlessly replace the occurrence of the "Store patient records (paper file)" use case with the "Store patient records (computerized file)" use case.

✚ **Creating Use Case Diagrams:-**

Rational Rose supports the development of the majority 2 use case models, as follows:

- Business Use Case diagram
- Use Case diagram

Business Use Case:-

- Business Use Case diagrams are used to represent the functionality provided by an organization as a whole.
 - They answer the questions:
 - "What does the business do?"
 - "Why are we building the system?"
- They are used extensively during business modeling activities to set the context for the system and to form a foundation for creating the use cases.

- An example of a simplified Business Use Case diagram for a financial institution is shown in Figure 1.8.

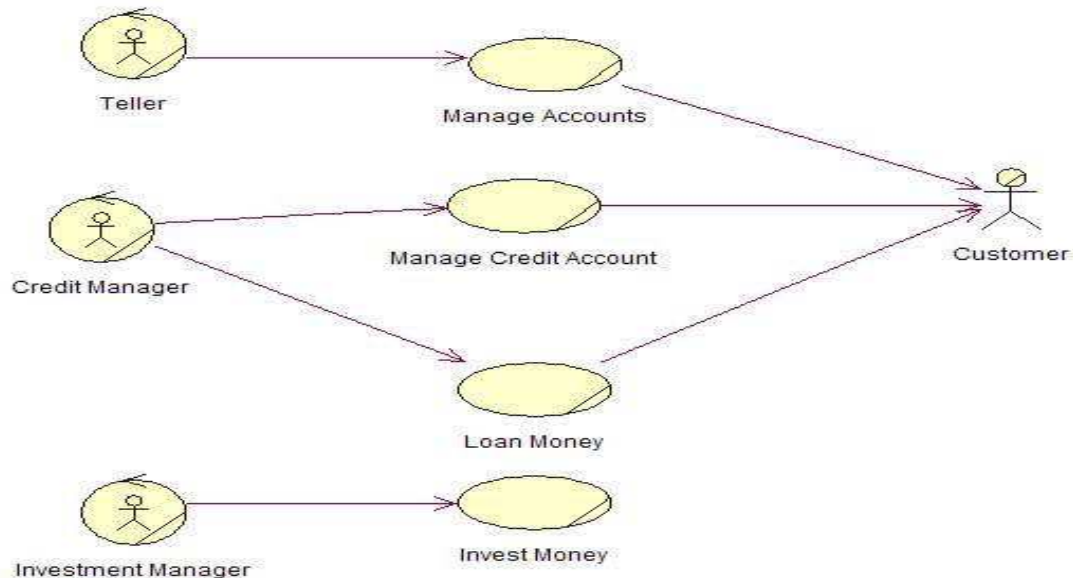


FIGURE 1.8 Business Use Case Diagram for a financial institution
REF: Boggs & Boggs, 2002

- Business Use Case diagrams are drawn from the organizational perspective.
- They do not differentiate between manual and automated processes.
- Use Case diagrams, which will be discussed next, focus on the automated processes.
- Business Use Case diagrams show the interactions between business use cases and business actors.
- Business use cases represent the processes that a business performs, and business actors represent roles with which the business interacts, such as customers or vendors.

- In other words, business actors represent anyone or anything outside the business that interacts with the business; they do not represent roles or workers within a business.

Use Case Diagrams

- Use Case diagrams show the interactions between use cases and actors.
- Use cases represent system functionality, the requirements of the system from the user's perspective.

Actors represent the people or systems that provide or receive information from the system; they are among the stakeholders of a system

- Use Case diagram can illustrate the requirements of the system.
- While Business Use Case diagrams are not concerned with what is automated, Use Case diagrams focus on just the automated processes.

An example of a Use Case diagram for an Automated Teller Machine (ATM) system is shown in Figure 1.9. This Use Case diagram shows the interactions between the use cases and actors of an ATM system.

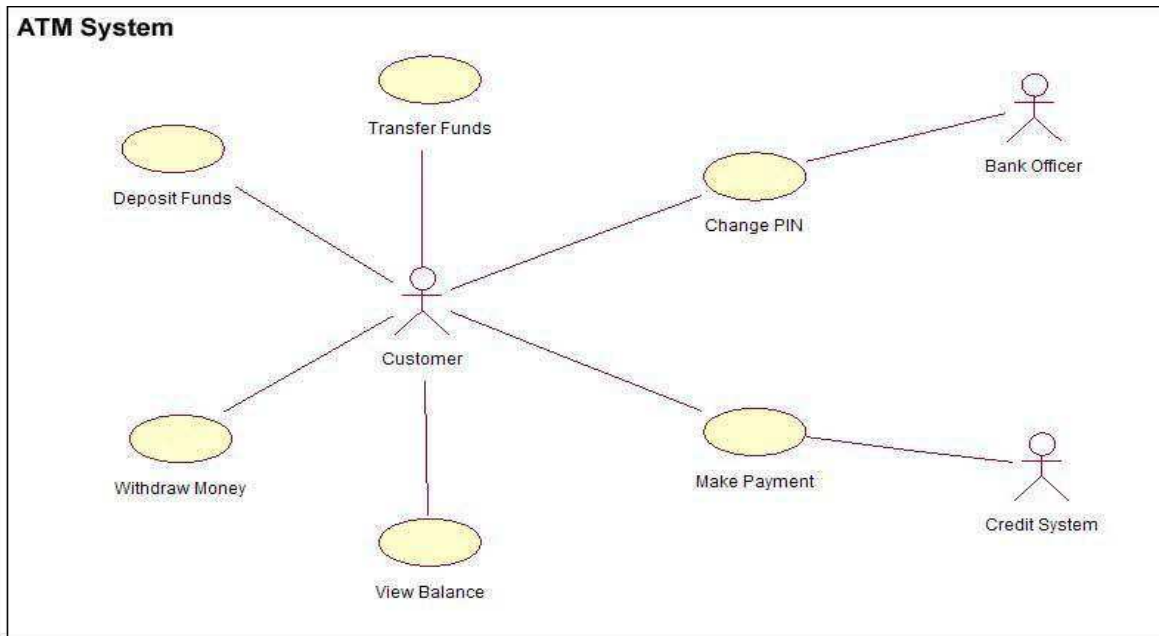
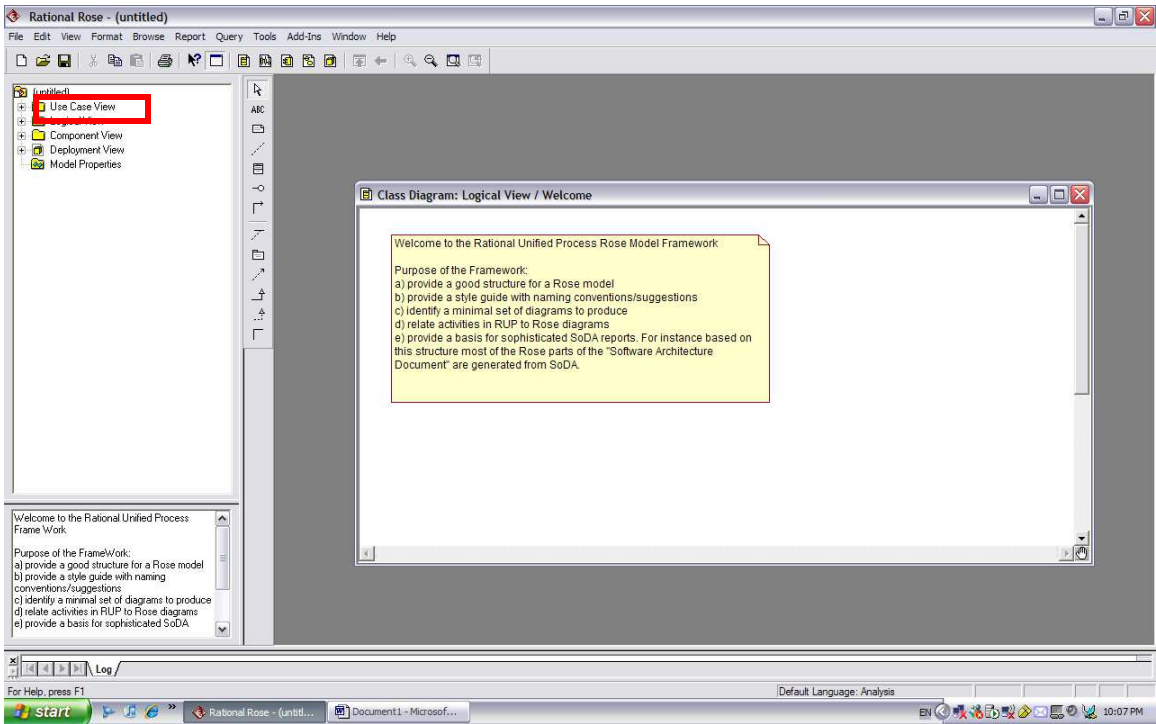
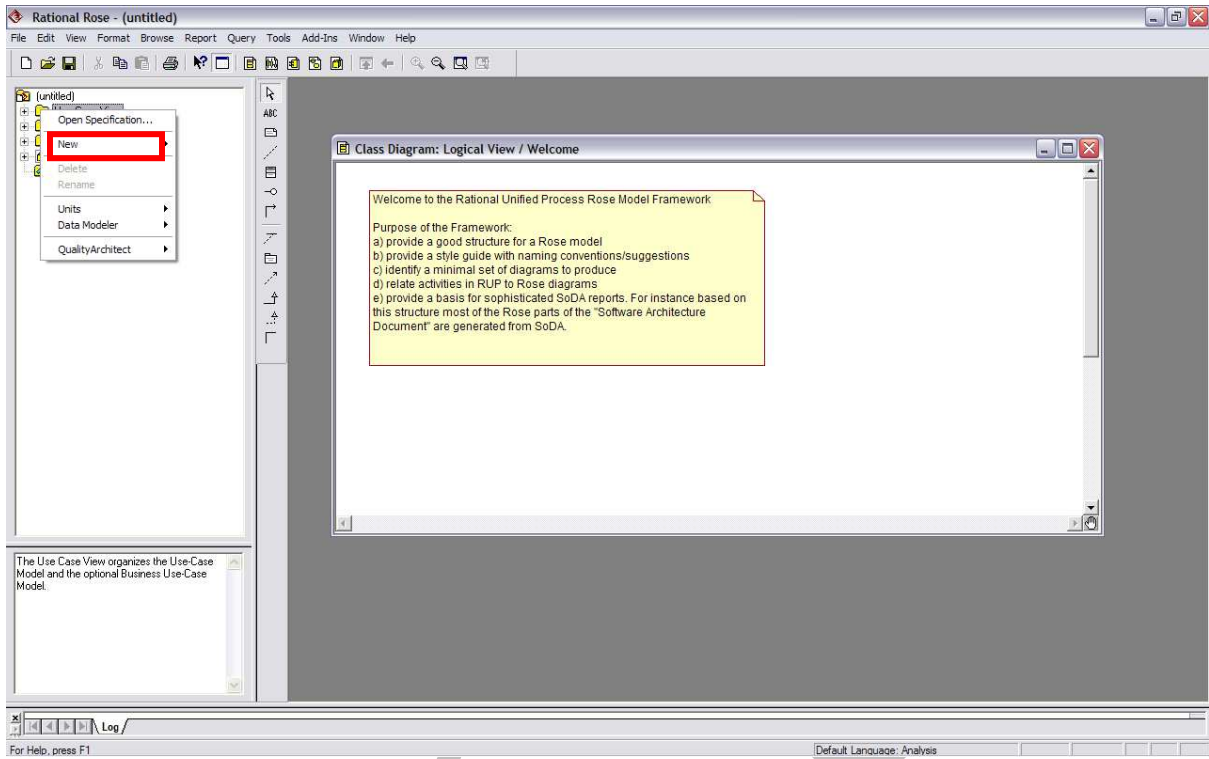


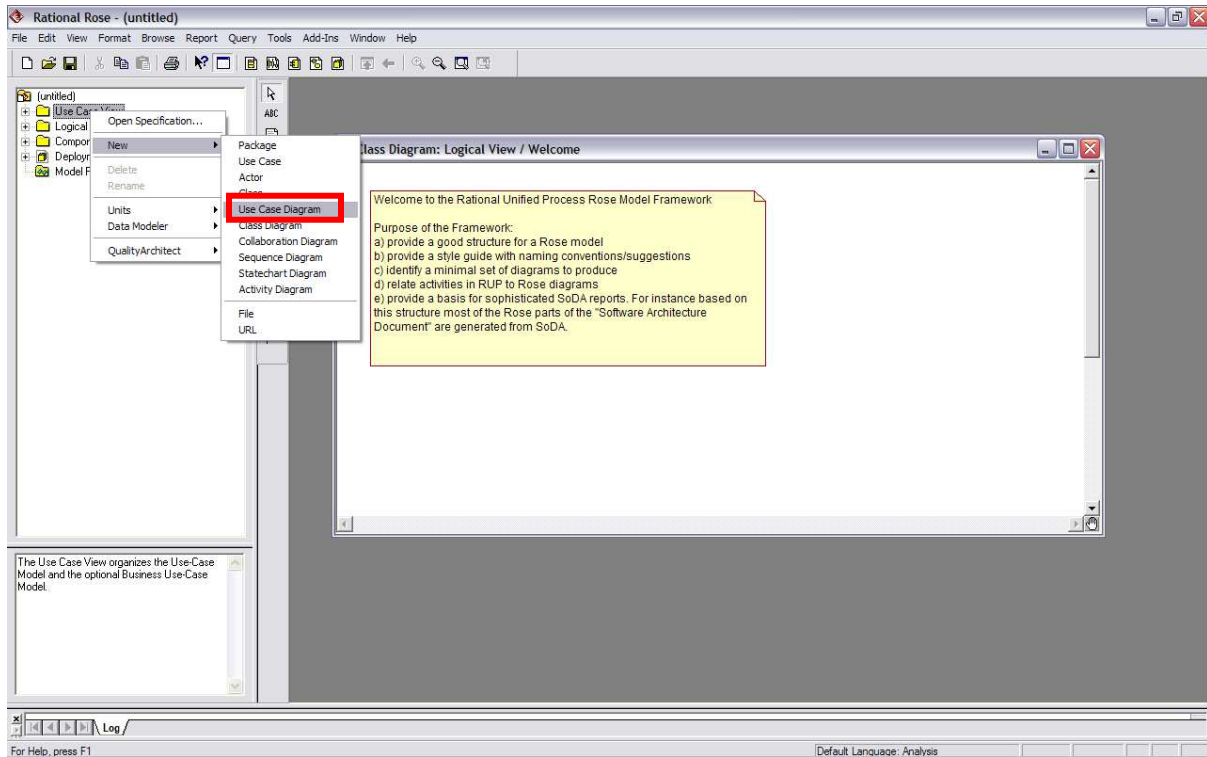
Figure 1.9 Use Case diagram for an Automated Teller Machine (ATM) System
 REF: Boggs & Boggs, 2002

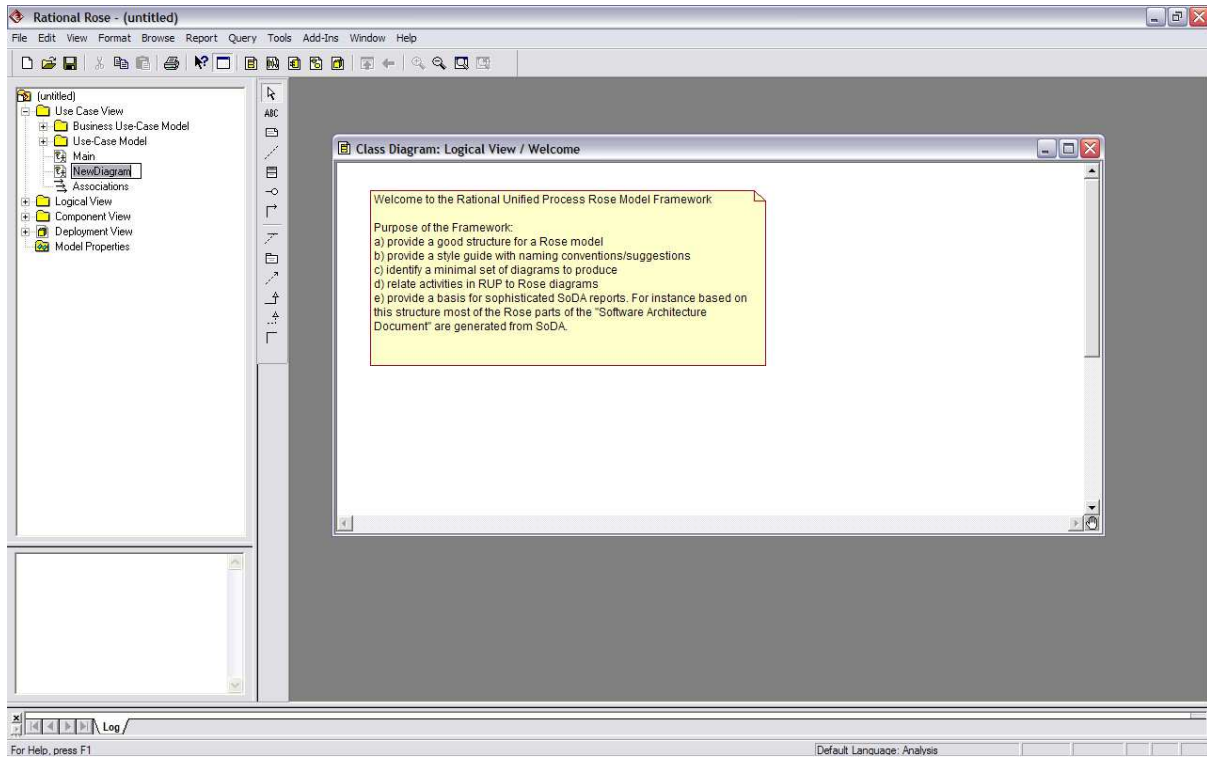
- Much information can be gleaned from viewing Use Case diagrams.
 - This one diagram shows the overall functionality of the system.
 - Users, project managers, analysts, developers, quality assurance engineers, and anyone else interested in the system as a whole can view these diagrams and understand what the system is supposed to accomplish.
1. To create a new Use Case diagram:
 - Right-click Use Case View
 - Select New
 - Select Use Case Diagram



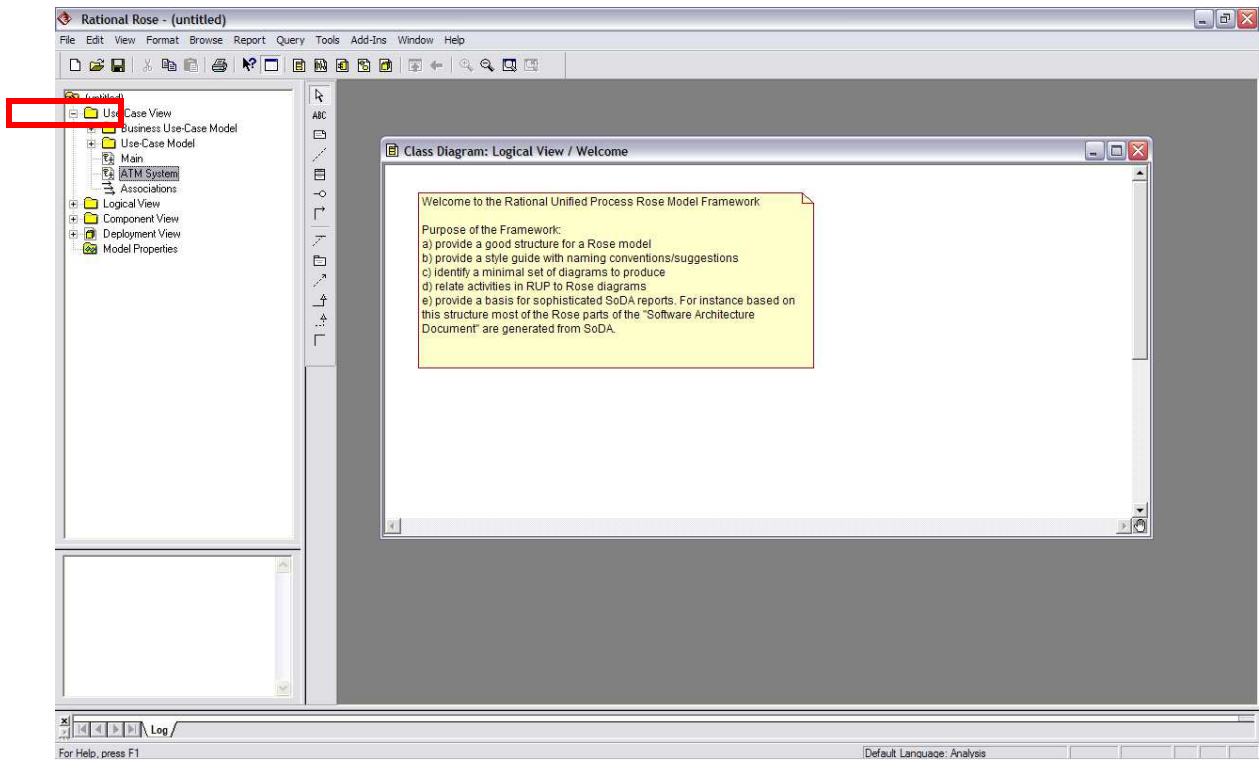


- Select New → Use Case Diagram and Select Use case Diagram → New Diagram

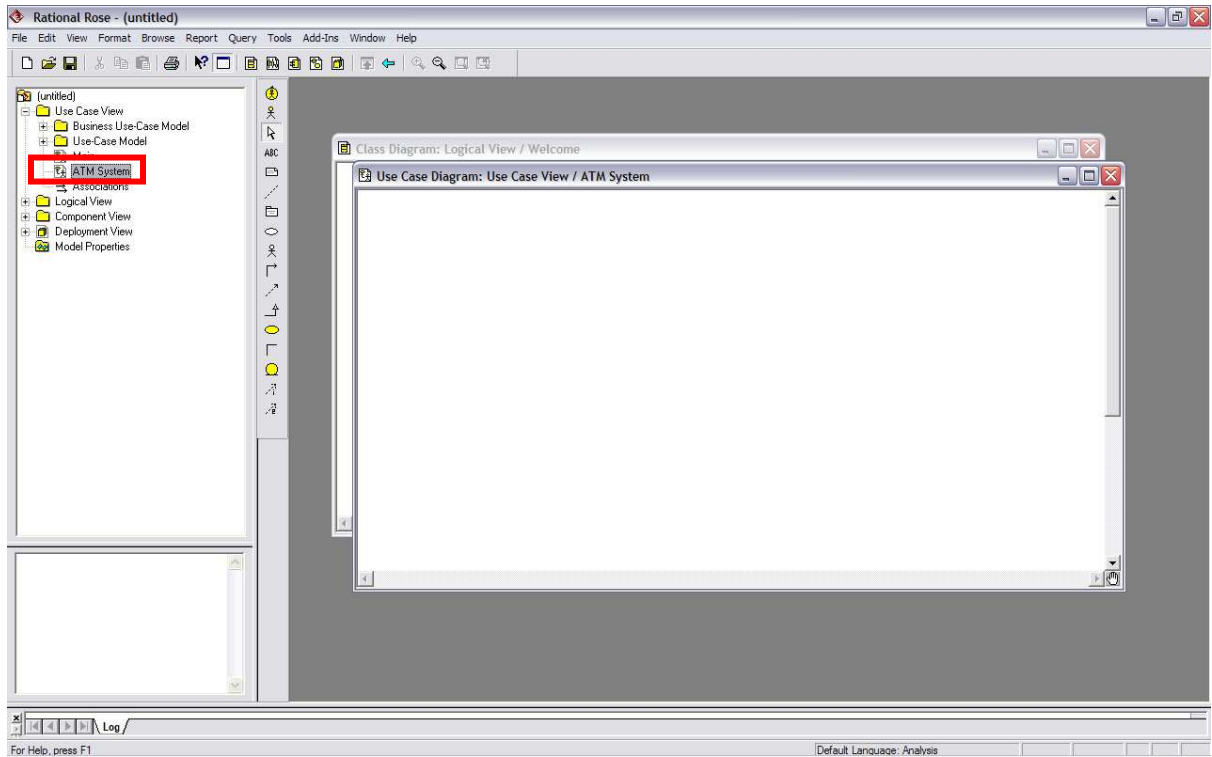




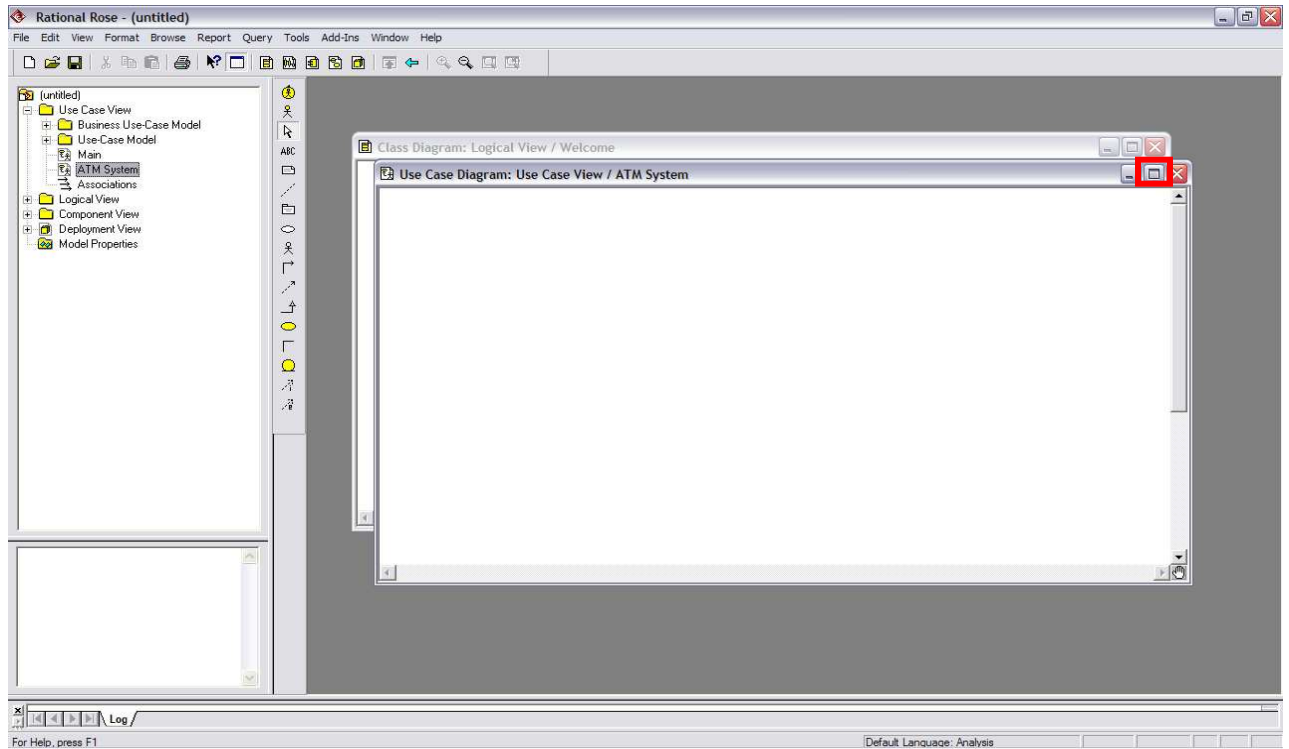
- Name the diagram as “ATM System” on the New Diagram Box.



- Left Double- Click the ATM System button to start working with the Use Case diagram.

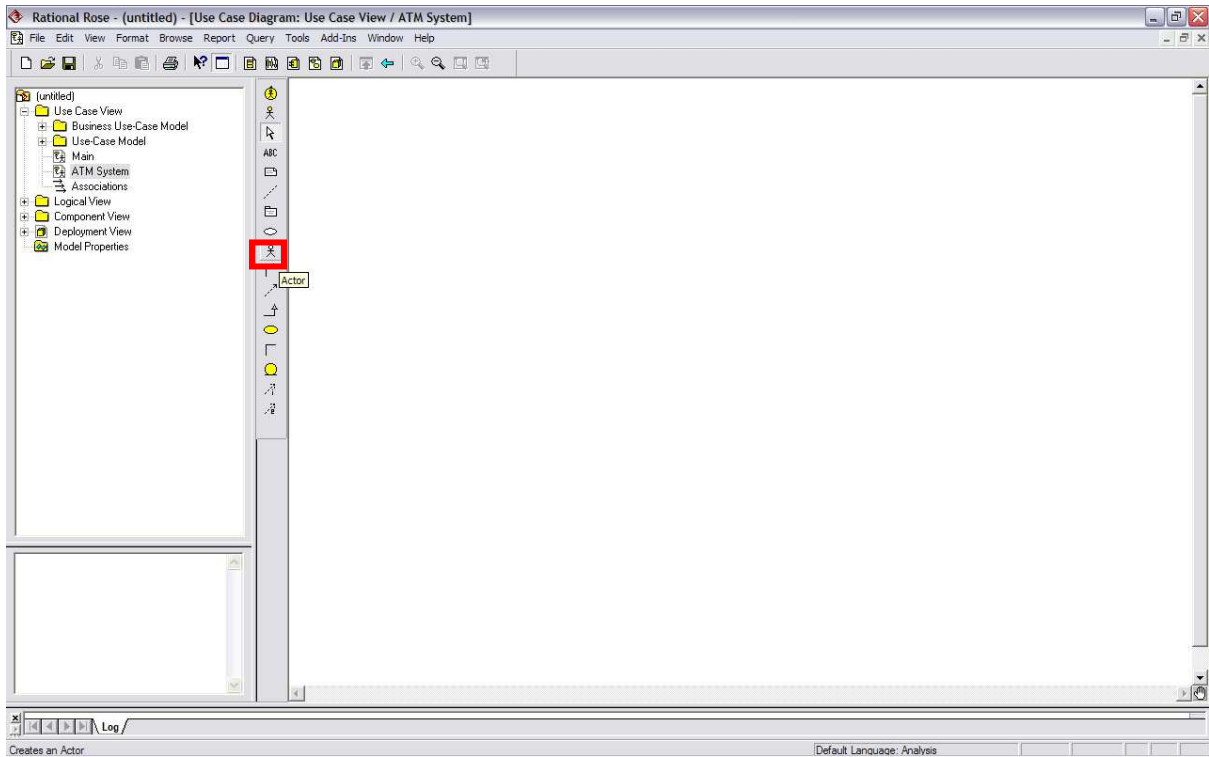


- In order to work with the diagram more easily, maximize the size of the diagram.

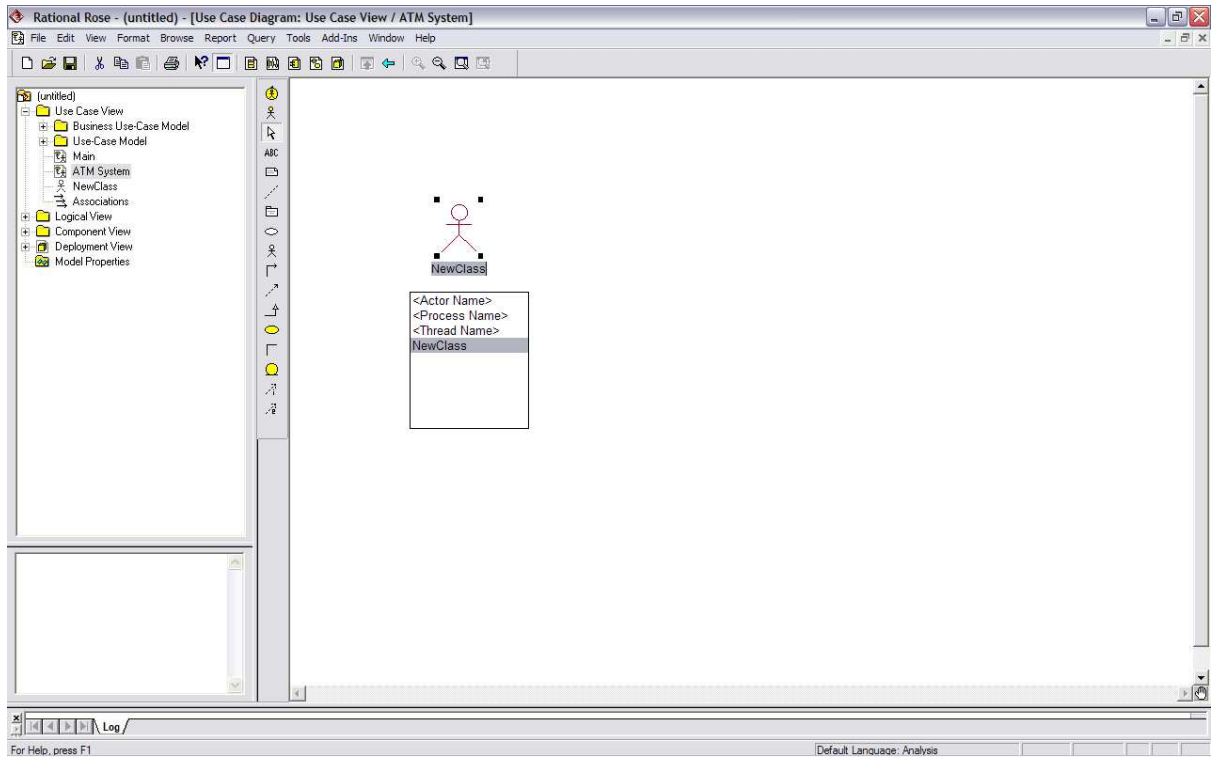


2. In order to add actors to the diagram:

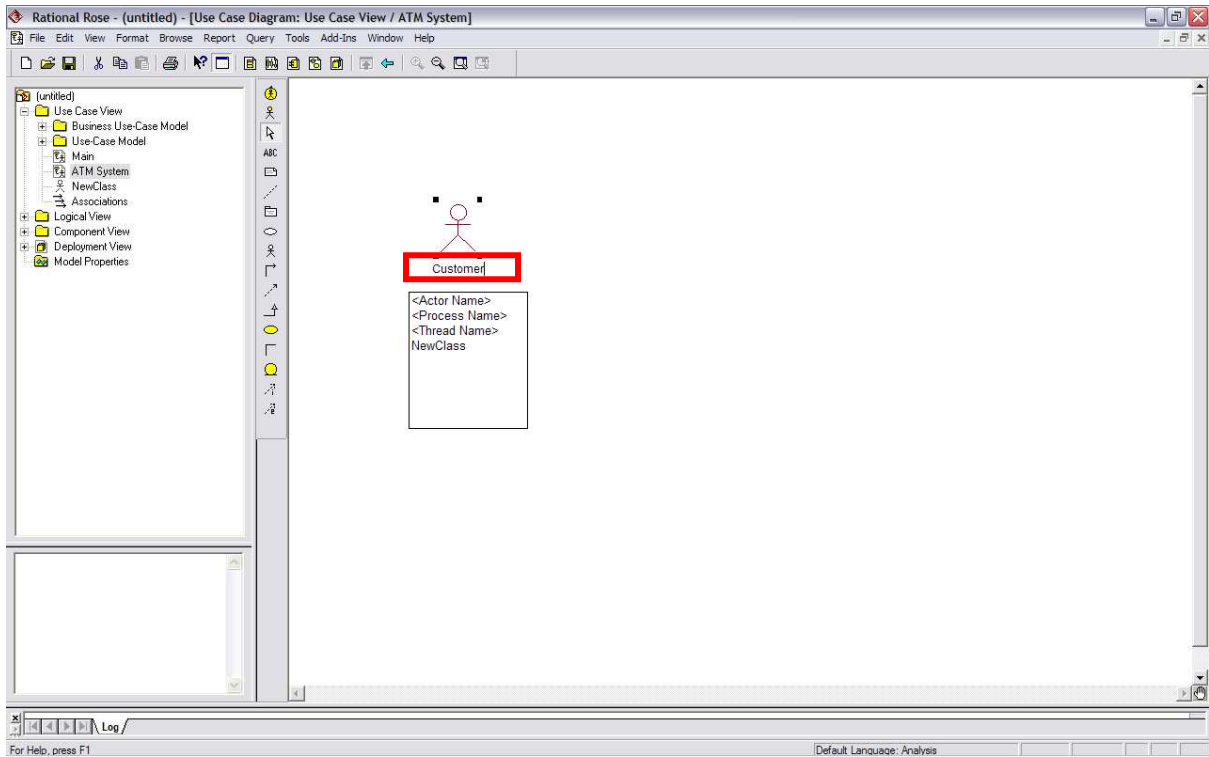
- Determine the actors for your system. In this example there are three actors: Customer, Credit System, and Bank Officer.
- Select Actor toolbar button



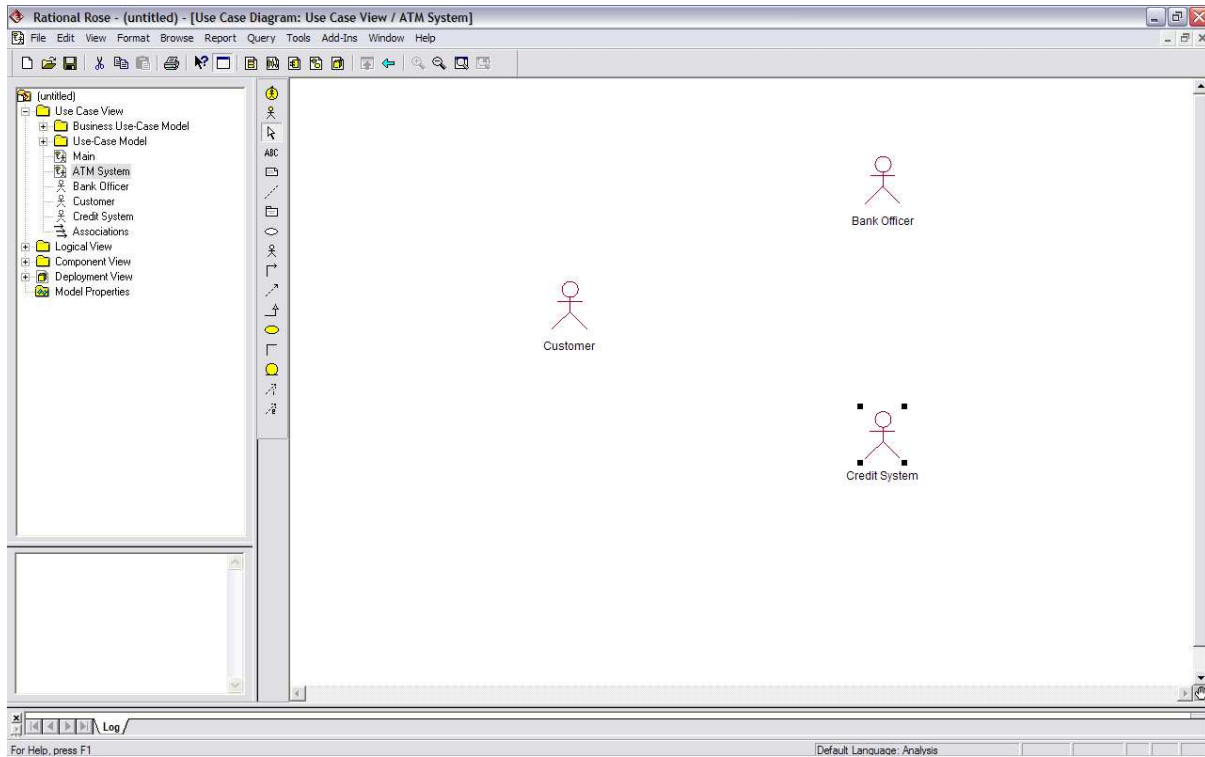
- Left- Click Actor toolbar button to add an Actor to the Use Case diagram at a point marked by a cross () sign on the screen.



5.4 Name the actor as Customer

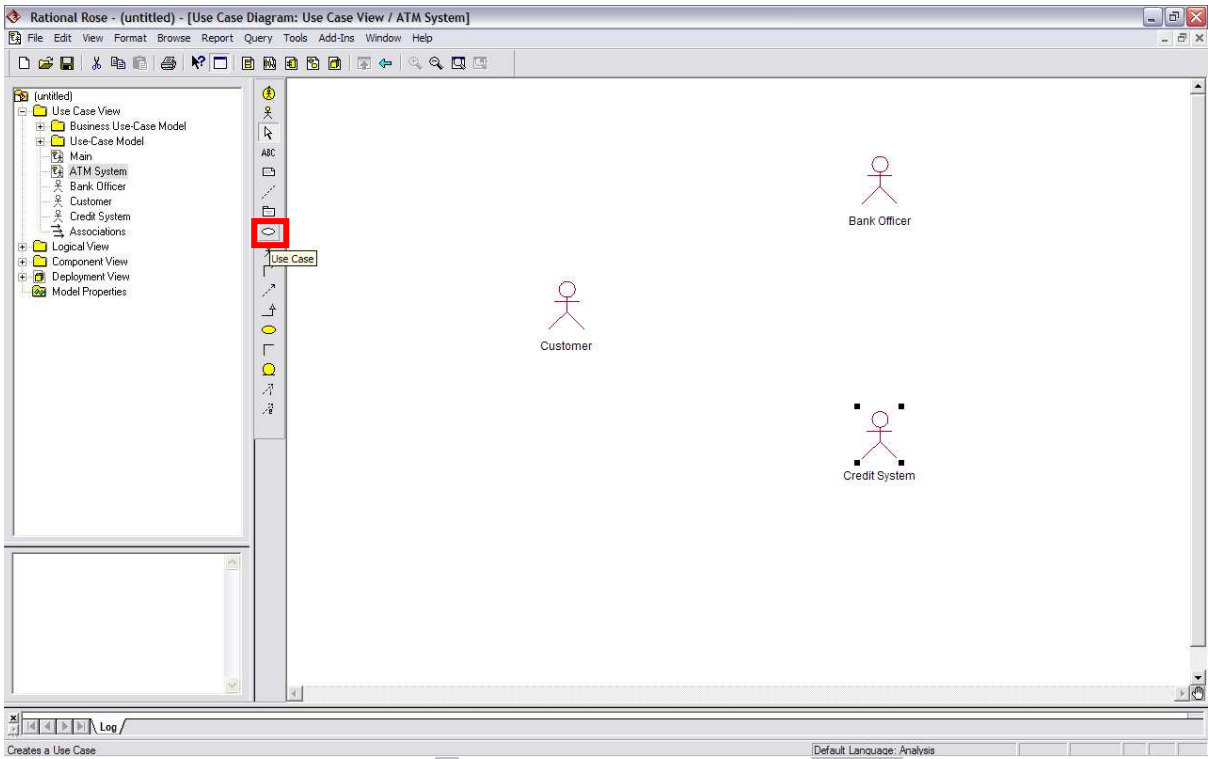


- Repeat Steps for the actors Credit System and Bank Officer



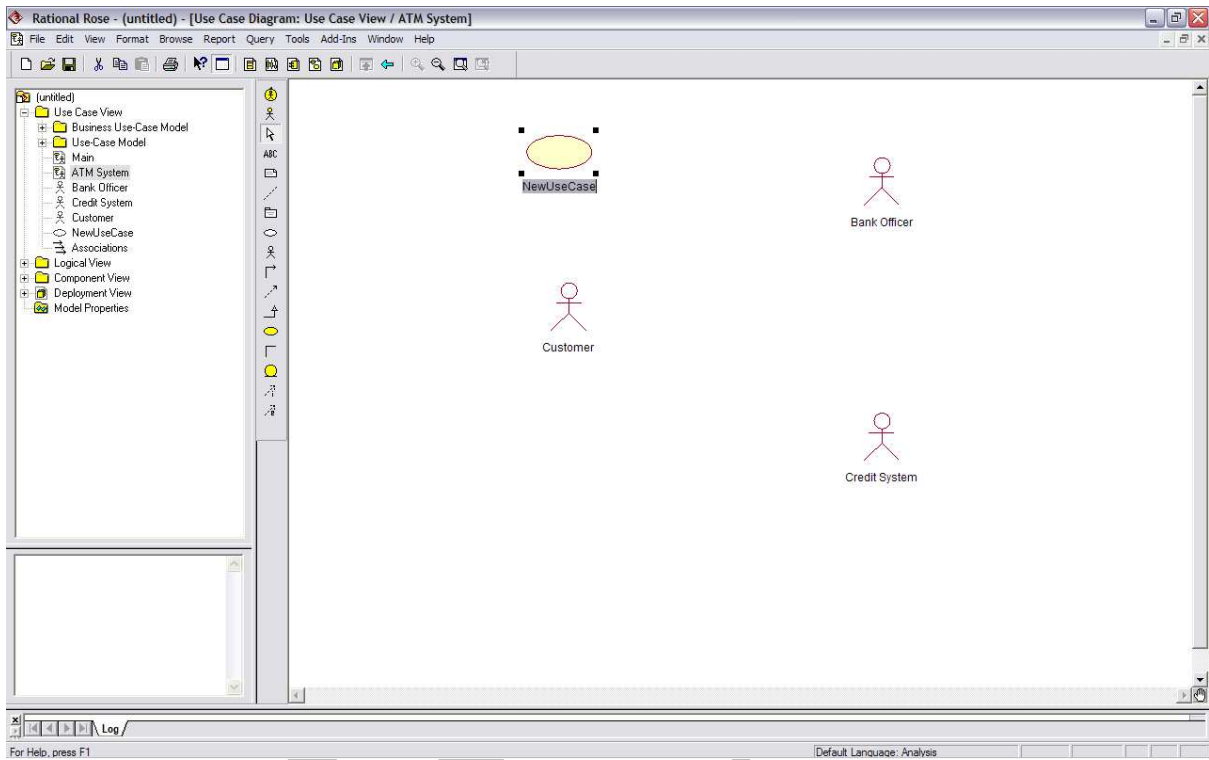
3. To add use cases to the diagram:

- Determine the use cases for your systems. In this example, use cases are “Transfer Funds”, “Change PIN”, “Make Payment”, “View Balance”, “Withdraw Money”, “Deposit Funds”
- Select the Use Case toolbar button

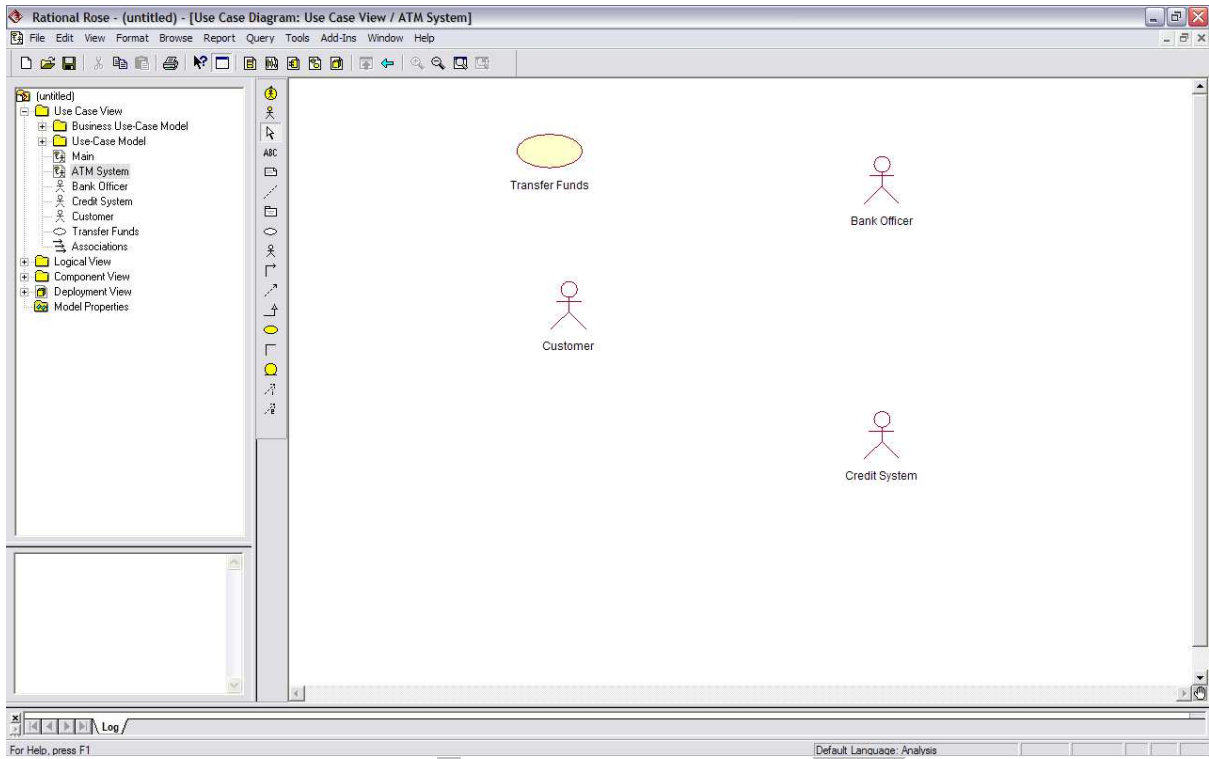


WU

- Left-Click the Use Case box to add the use case into the diagram.



6.1 Give the use case name "Transfer Funds".

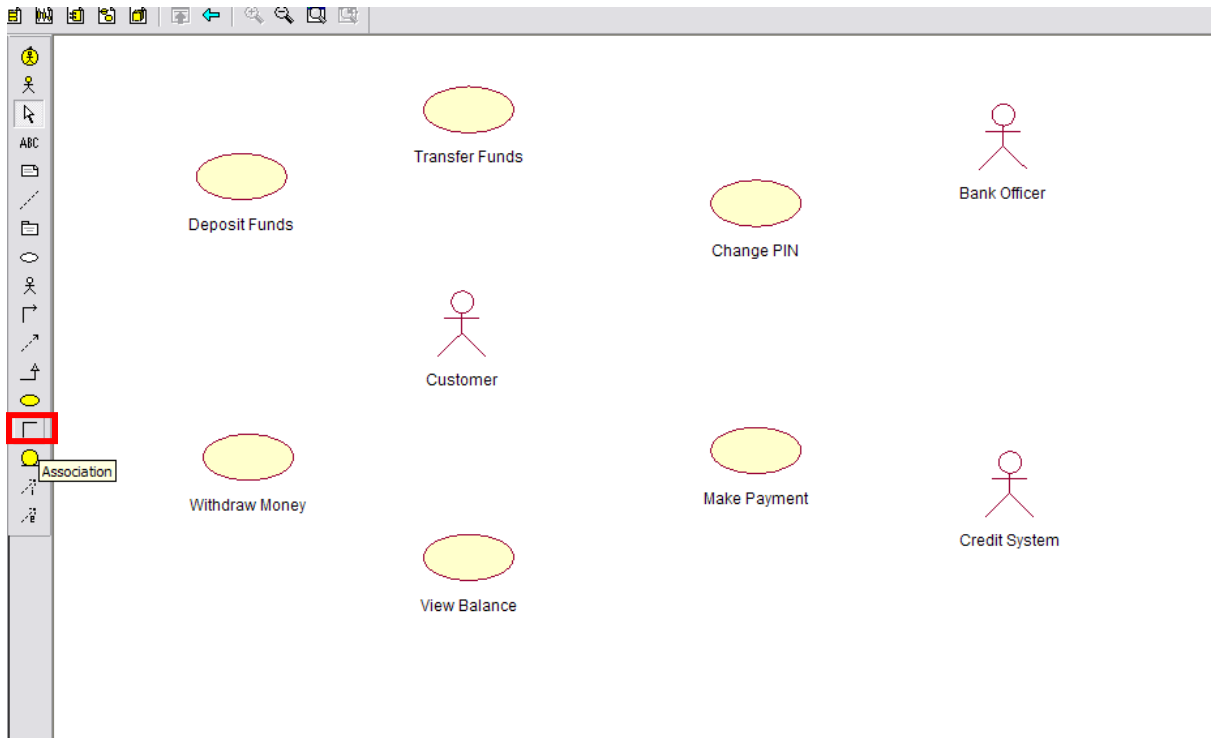


- Repeat Steps for the rest of the use cases.

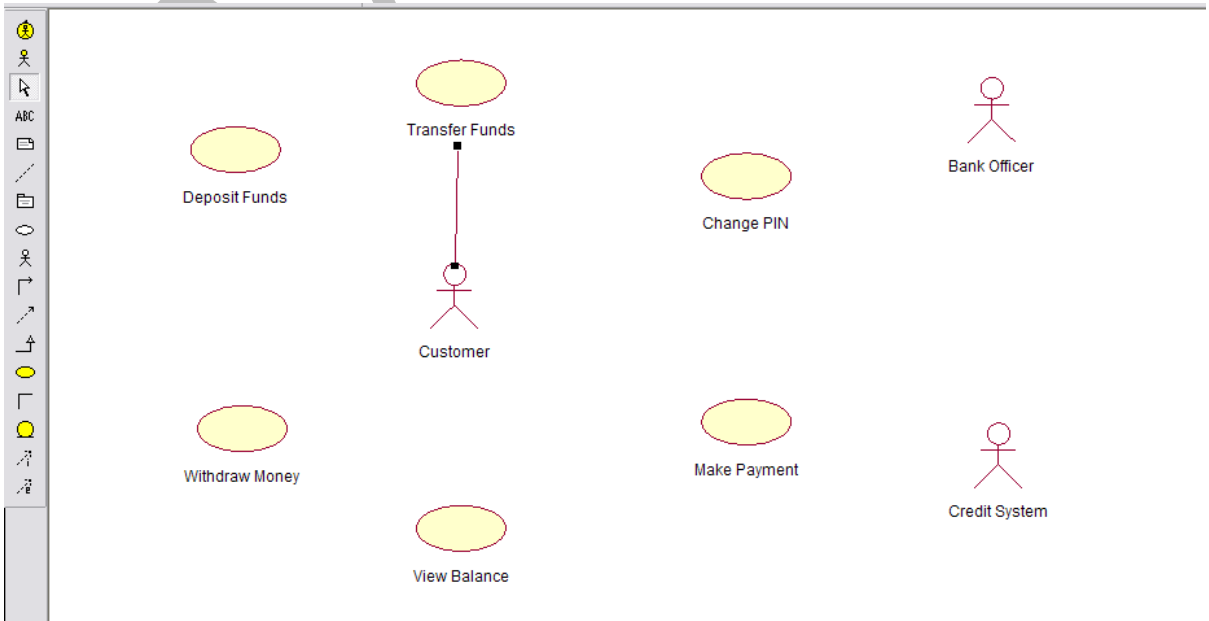
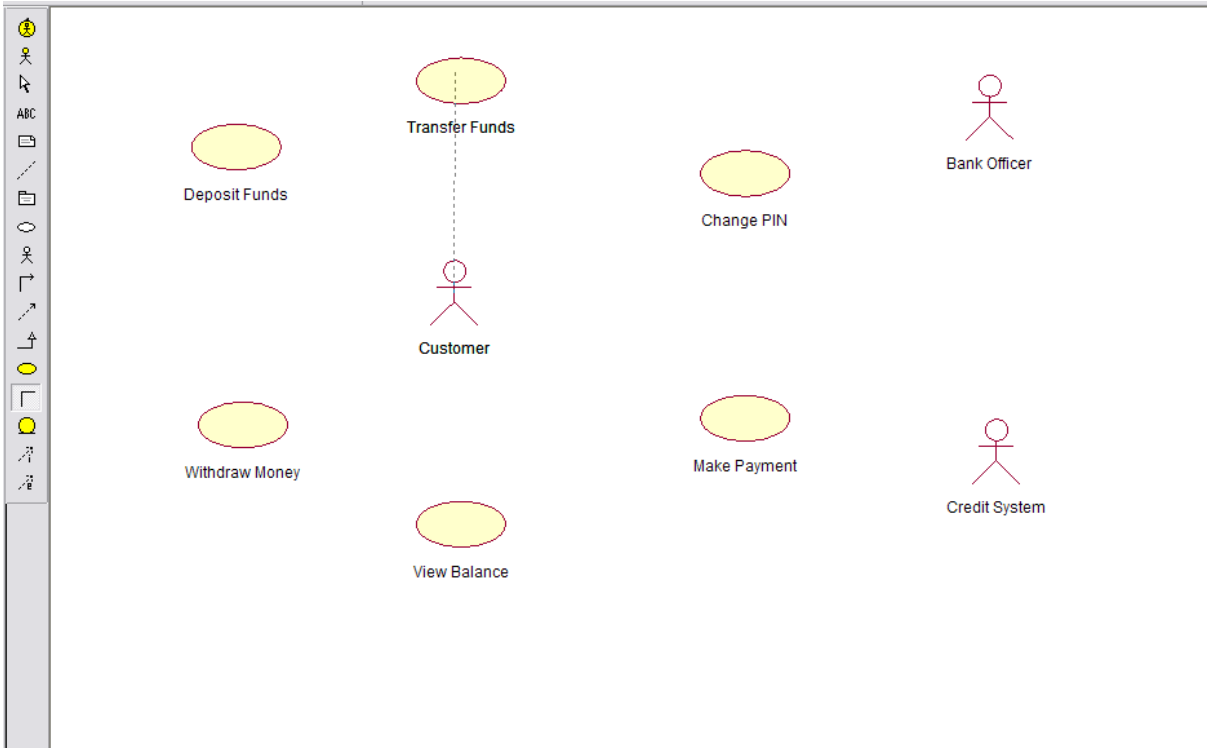


4. In order to add relationships between actors and use cases:

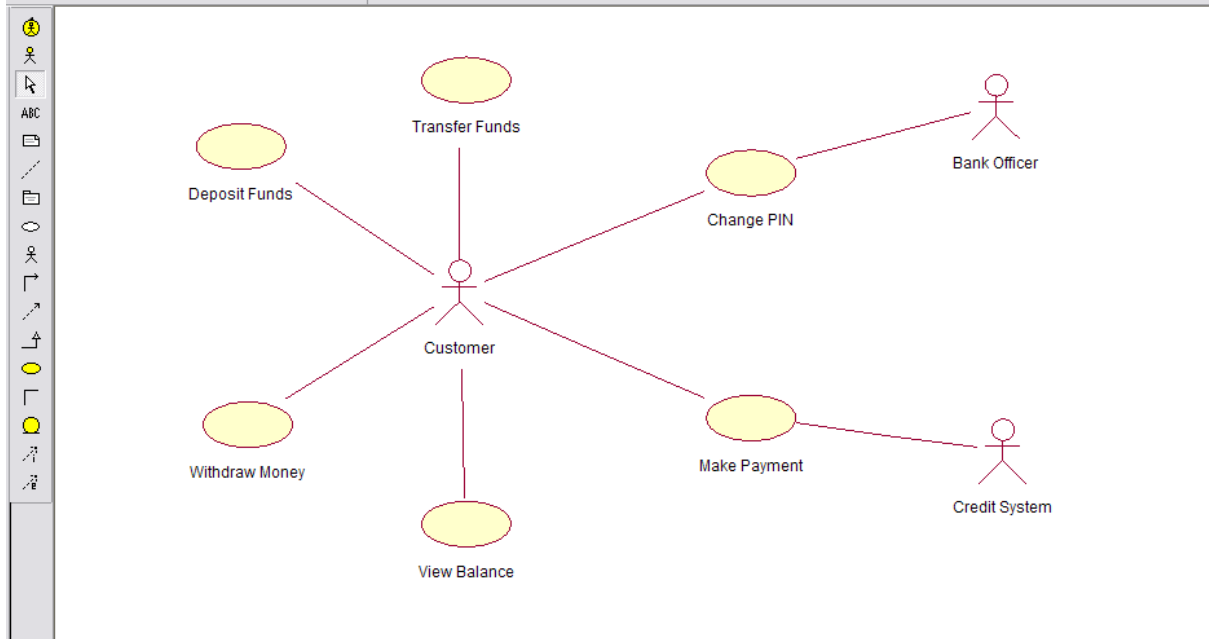
- Select the related Association toolbar button
- If the proper button is not available in the vertical toolbar , one can add it by right-clicking any one button and then select Customize and add the missing button to the tool bar.



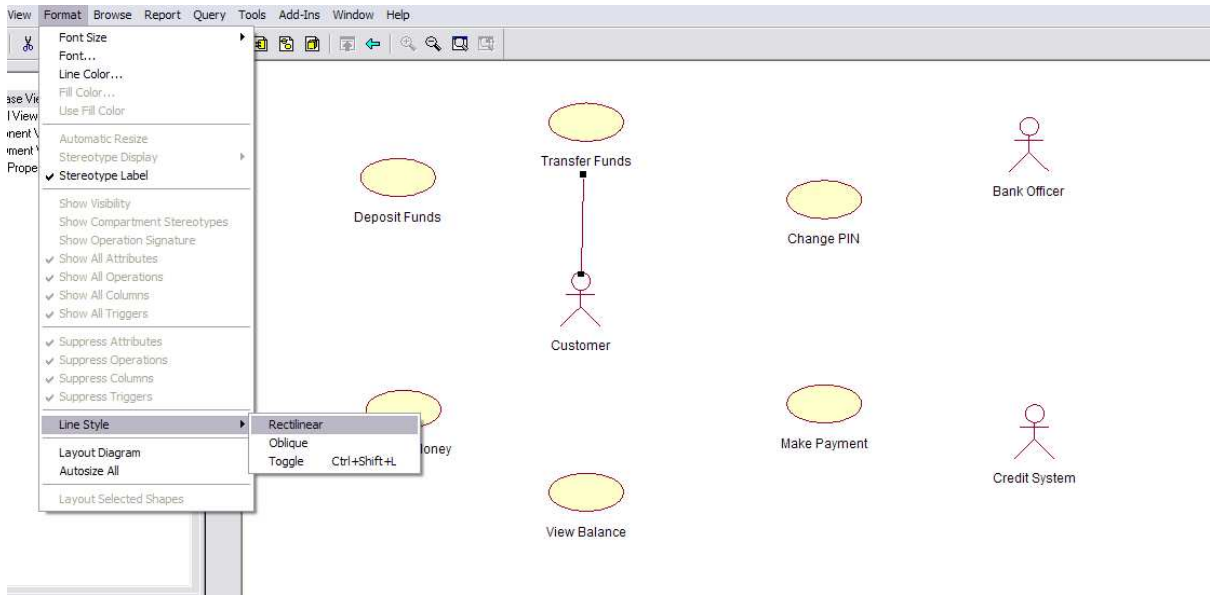
- Drag an association line from the use case to the proper actor.



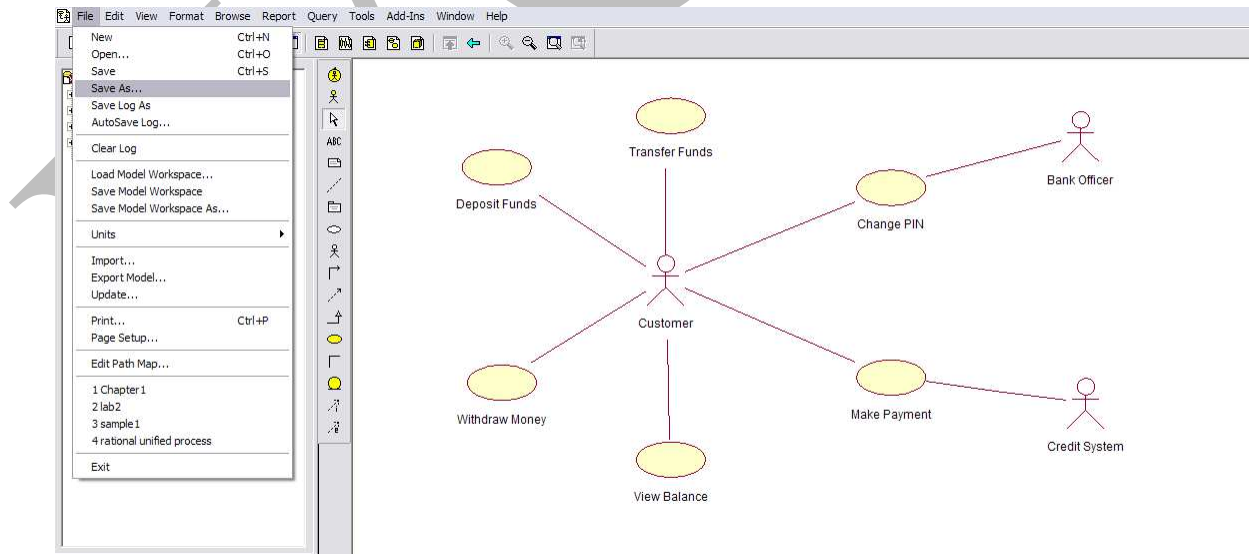
- Repeat Steps for each association.

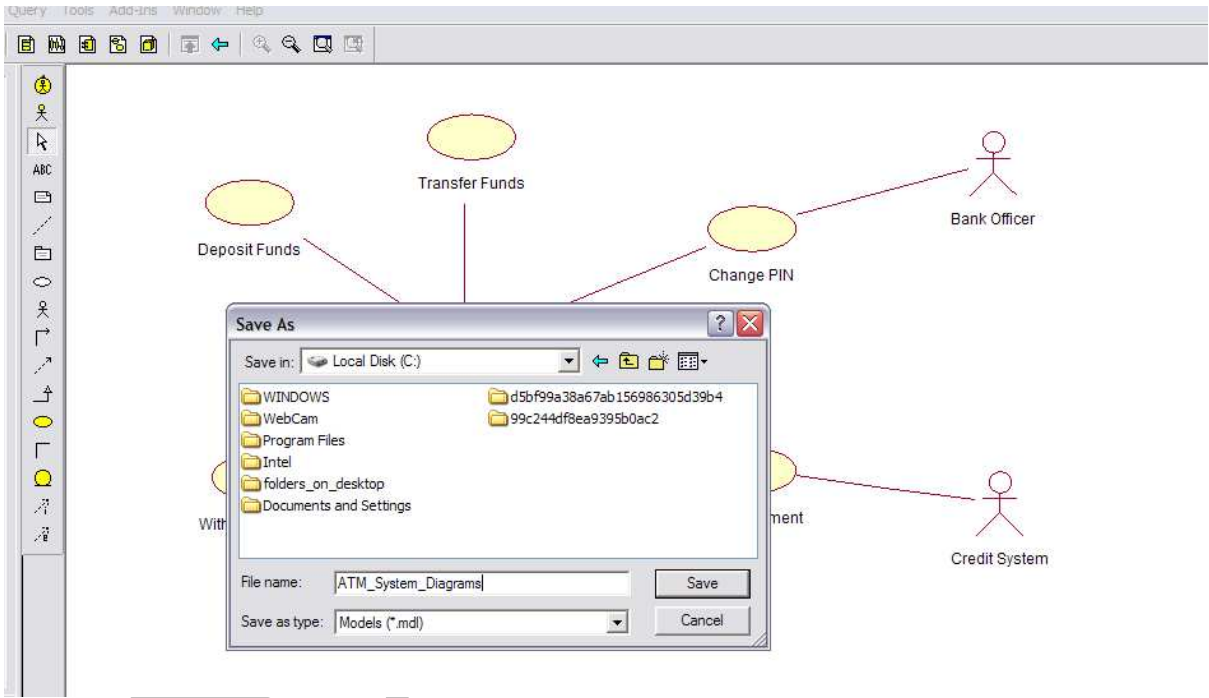


- If you want to format style of association Select Format at the top toolbar. Line Style→ Rectilinear; to change fonts Select Format → Font; to change line or fill colors Select Format → Line Color or Select Format → Fill Color from the menu.



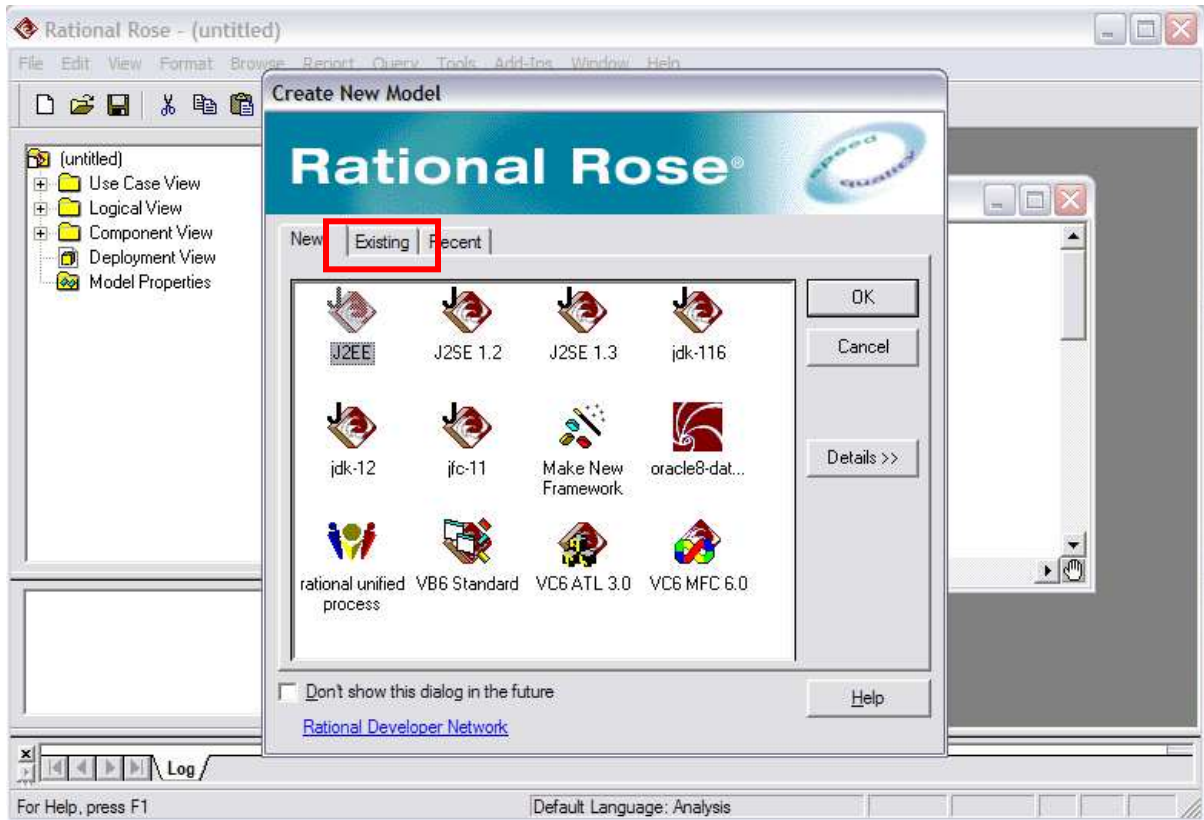
- Save the model as “ATM_System_Diagrams” under folder C by Selecting File → Save As or Using Save button:

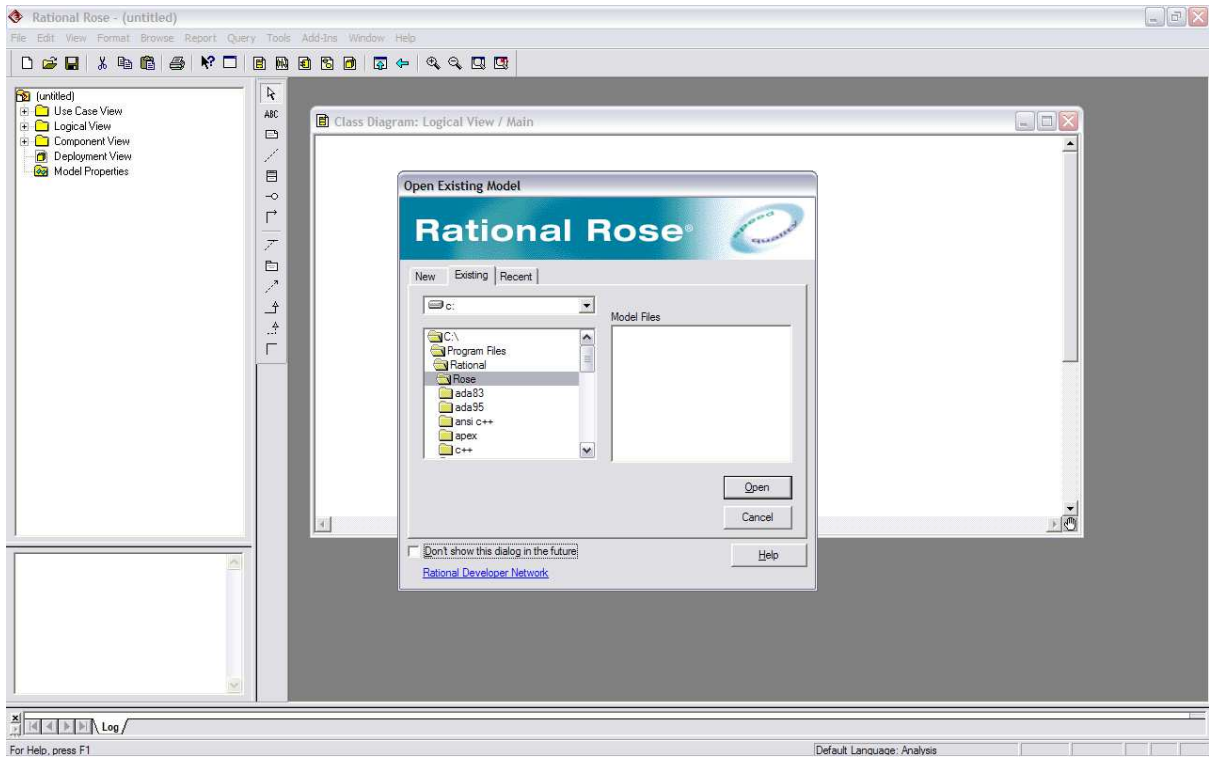




In order to open an existing file:

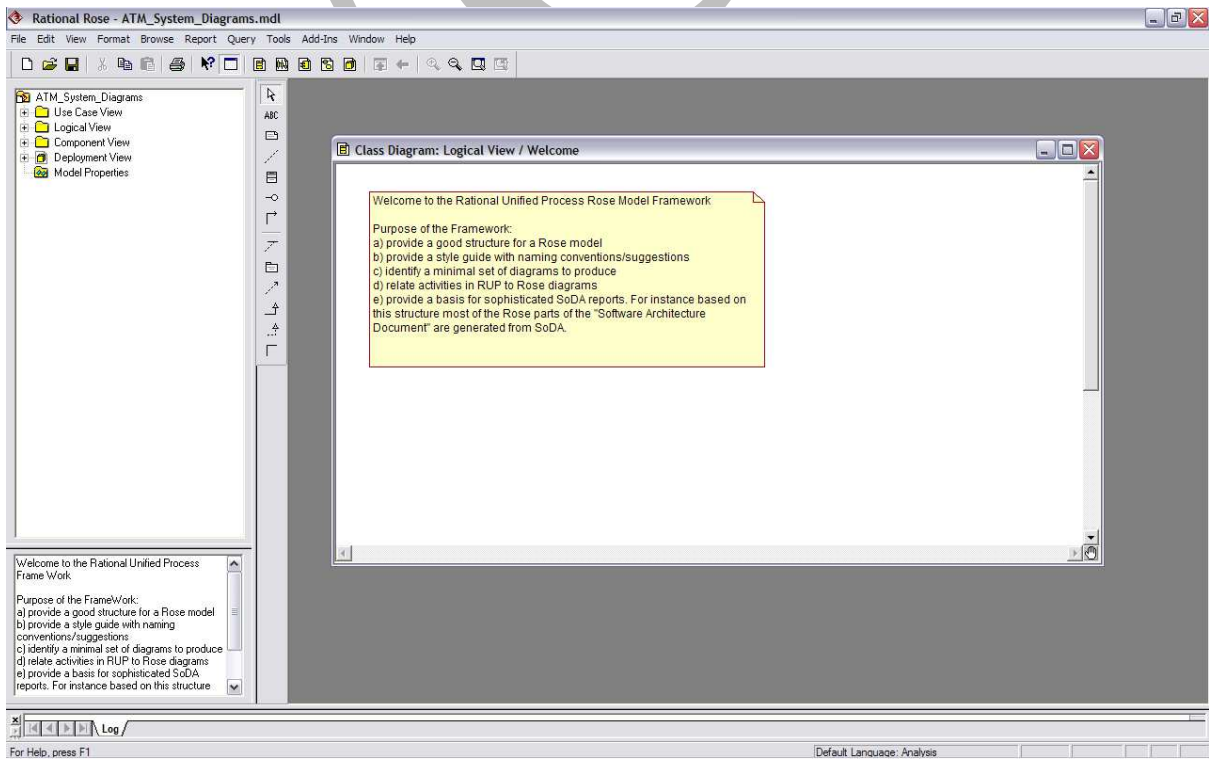
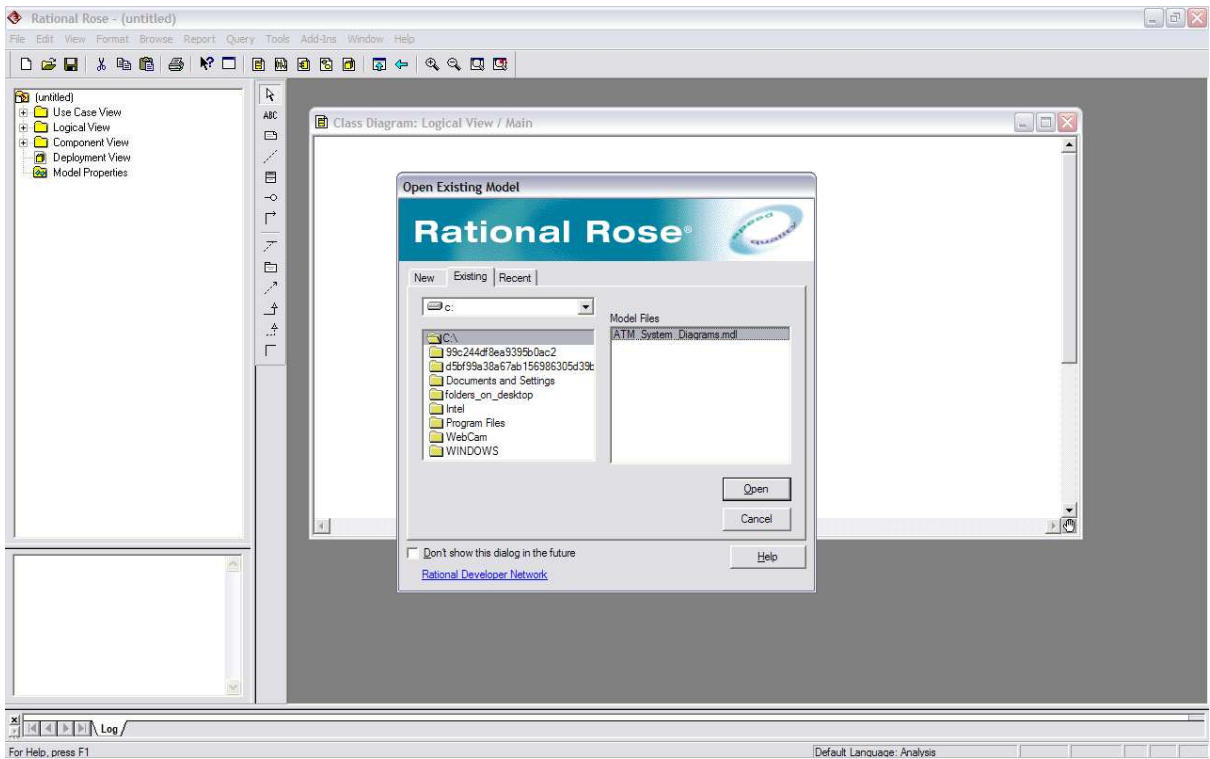
- Click “Existing” or “Recent” Tab on the opening screen of Rational Rose.





- Now, open the file "ATM_System_Diagrams" under C: drive.





- Click first Use Case View under ATM_System_Diagram of Browser part. Next click ATM System under Use-Case Model.
- In order to delete a particular diagram, right click the diagram name and delete.

Conclusions-:

In this way, we studied use case diagram in Rational Rose 2002.

WAVE