Using Rational Rose to Create Object-Oriented Diagrams

This is a brief overview to get students started in using Rational Rose to quickly create object-oriented models and diagrams. It is not by any means a complete introduction to Rational Rose, but it should get you started.

A. GETTING STARTED:

Click on Rational Suite Enterprise 1.0, and then on Rational Rose 98I Enterprise Edition (**not the one that has the arrow next to it)**. You will now see a window appearing that says "**Create New Model**". This is for working with Java, Oracle, and VB. Just press "Cancel".

Next, you see the main window called "**Rational Rose-[untitled]**" and a smaller window within it called "**Class Diagram: Logical View / Main**". There is also a browser on the left side of the page that has the diagrams that you are working on in the project. If you ever want to see the specifics on any diagram that you are working on, go to the browser, click on the " **+** ", and you'll be able to see what is in that view. To close the view, click on the " **–** " .

B. USE-CASE DIAGRAM:

1. The first diagram you'll be working on is a use-case diagram. Go to Browse and click on **Use Case Diagram.** You now will see a new window called **Use Case Diagram: Use Case View / Main**. You will see a set of tools to the left of this window. These tools are specific to the Use Case diagram and will be used to create it. To make a Use Case, go to the tools and click on the one that looks like an oval. This is the Use Case button. (Note: If you ever want to know if it's the right button, just hold the mouse arrow over the button for two seconds, and it will tell you what the button is.)

2. Once you have clicked on the Use Case button, position the new mouse pointer wherever you want on the page and click again. The oval will now appear on the page. Now, type in a name for your Use Case. To reposition the Use Case or change its name, just click once on the Use Case.

3. Now, we need to add some actors, which are actually the classes in this project (you'll see this later as you continue). Click on the stick figure in the toolbox. Now, click anywhere on the Use Case Diagram to place the actor. Once again, name your actor. As with the use cases, to reposition the actor or change its name, just click once on the actor.

4. Finally, we want to connect the use cases with the actors. In order to do this, click in the toolbox on the box that has the curved arrow that says association. This will allow

you accomplish this task. You are now finished with the Use Case Diagram portion of the project. **Don't forget to save!!!**

C. CLASS DIAGRAM:

1. Now that we have defined our use-cases and actors, we need to create the class diagram. The first step is to go into the correct screen that is used for producing class diagrams. In order to do this, go to Browse and click on **Class Diagram.** Make sure to click on **Class Diagram: Logical View / Main.** You now will see a new window. Once again, you will see a set of tools to the left of this window. These tools are specific to the Class diagram and will be used to create it.

2. To create the classes, you need to click on the **class** icon. (Note: If you ever want to know if it's the right button, just hold the mouse arrow over the button for two seconds, and it will tell you what the button is.)

3. Once you have clicked on the **class** button, position the new mouse pointer wherever you want on the page and click again. A rectangle representing the class will now appear on the page. Now, type in a name for your class. To reposition the class, just click once on the class. To change its name, click twice.

4. Now we need to add some attributes to the class. Let's insert an attribute. To do this, **right click** on the class. Select **new attribute**. Now, type in the name of the attribute of the class.

5. Now, we want to give the attributes their types, initialize the types, and distinguish the attributes as either public, private, or protected. To do this, **double click** on the class anywhere that is **blank**. A new window called **Class Specification** will pop up. Click on **attributes**. **Double click** on the attribute that you want to select. Another new window called **Class Attribute Specification** pops up. In this window, you can initialize your attribute to any value, specify if it's public, private, or protected, and give it the type.

6. Next, we add some methods to our class. To insert a method of the class, **right click** on the class and this time select **new operation**. Now, type in the name of the method of the class. We also want to add arguments for the new operation. To do this, once again, **double click** on the class anywhere that is **blank**. The **Class Specification** window will again pop up. Click on **operations**. **Double click** on the method that you want to select. Another new window called **Operation Specification** pops up. In this window, the G**eneral** tab is already selected. (If not, select it). Here you can decide if the methods will be public, private, or protected. After doing this, select the **Detail** tab. **Right click** in the space below **Name**. Select **insert**. Now, type in the name of the argument. Hit tab to insert the argument's **type**. Hit tab again if you want to set the argument to a default. Press OK and you're back to **Operations.**

7. Great. Now that we have all the classes created, we can create an **ASSOCIATION** connection between classes. Click on the **ASSOCIATION** icon. Now, click on the first class, hold the mouse button down, drag the mouse to the other class, and release the mouse button.

   To name the association, **double click** on the association connection line. The **Association Specification** window pops up. Type the Association name in the space provided. Press OK. Now, we need to insert the cardinality of the class association. To do this, **right click** on the association connection line, next to the first class. Select multiplicity and finally select the correct cardinality. Do the same for the second class.

   ** If you nave more than 1 association between 2 classes, be careful. If you select association between the 2 classes again, it will overlap the first one. Click on the line, drag the other association, and release it. Also, you can move the association names and cardinality around by clicking on them.

8. It's time for **AGGREGATION**. It's basically done the same way that Association is done.

   ** Be careful! Create the aggregation by starting off on the class that you want the diamond attached to.

   Don't forget to save!!!

D. STATE DIAGRAMS

1. Up until now, we dealt with the entire system, all the classes and use cases. Now, we're going to pick one of the classes and create a state diagram for it, specifically the **book** class. You need to let Rational Rose know which class you're going to use for the State Diagram. Select the book class by clicking on it once. Now, in the Browse menu, select **State Diagram.** The State Diagram window pops up.

2. The first thing to do is get the **start state** of the diagram. To do this, click on the black dot icon. This is the **start state.** Position the new mouse pointer wherever you want on the page and click again.

3. Next, click on the state icon. Once you have clicked on the state button, position the new mouse pointer wherever you want on the page and click again. A rounded rectangle representing the state will now appear on the page. Now, type in a name for your class by first **double clicking** on the state. To reposition the class, just click once on the class. To change its name, click twice.

4. Let's now give the transition between the states. To do this, click on the **state transition** icon on the tool bar. Click first on the state that the transition is coming

from, and drag it to the state that the transition is going towards. To name the transition, **double click** on the transition and type in the name.

5. Continue this with all states. Finally, you should have an **end state**. The end state is reached when the item in the state diagram is thrown out. Connect the transition to this state in the same manner as you did with the other states.

   Don't forget to save!!!

E. SEQUENCE DIAGRAMS

1. The last diagram in this exercise is the Sequence Diagram. Once again, this is a dynamic diagram, like the state diagram. Go to the Browse menu and select **Interaction Diagram**. In the **Select Interaction Diagram** window that pops up, choose **Logical View** and click on OK. When the window **New Interaction Diagram** pops up, choose sequence as the diagram type and type the title.

(Note: A Collaboration Diagram is almost the same as a Sequence Diagram. In this exercise, we will use the Sequence Diagram).

2. First, we need to choose the class objects that will be interacting in the diagram. To do this, choose the **object** icon. Position the new mouse pointer wherever you want on the page and click again. To name the class object, **double click** on the object.

3. To show a transition in the Sequence Diagram, click the right arrow icon. This is the "Object Message" icon used for transitions. Click on the vertical dotted line that extends from the bottom of the class object and drag it to the vertical dotted line of the second class object.

4. To make a self-transition, click on the "u-turn" arrow. This is the "Message to Self" icon. No dragging is necessary.

5. To name either transition, **double click** on the transition and type in the name.

** Note: You can move the transitions slightly up or down as needed by clicking on them and dragging them.

6. Once you make a transition, the vertical dotted line of the class objects extend.

OK, you're done with all the diagrams. Now, let's generate some code!!!