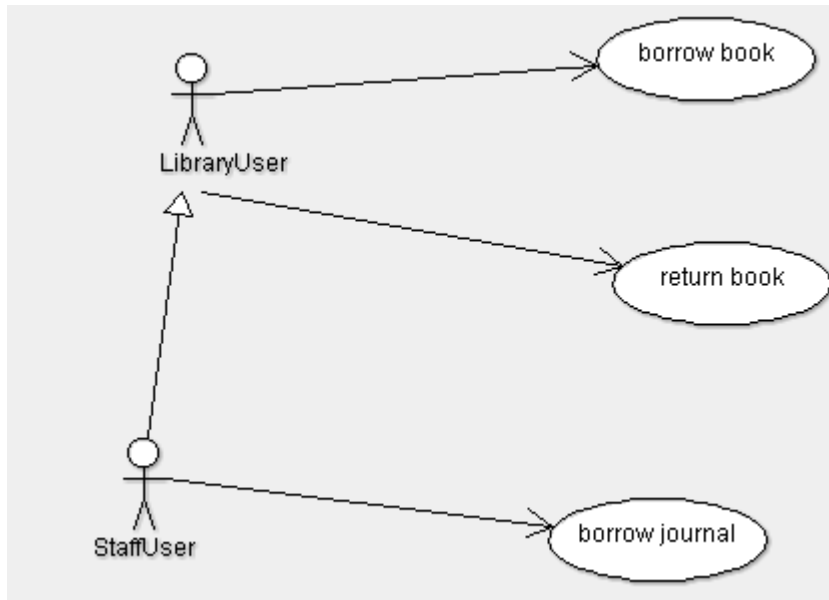


## Analysis Modelling for Library

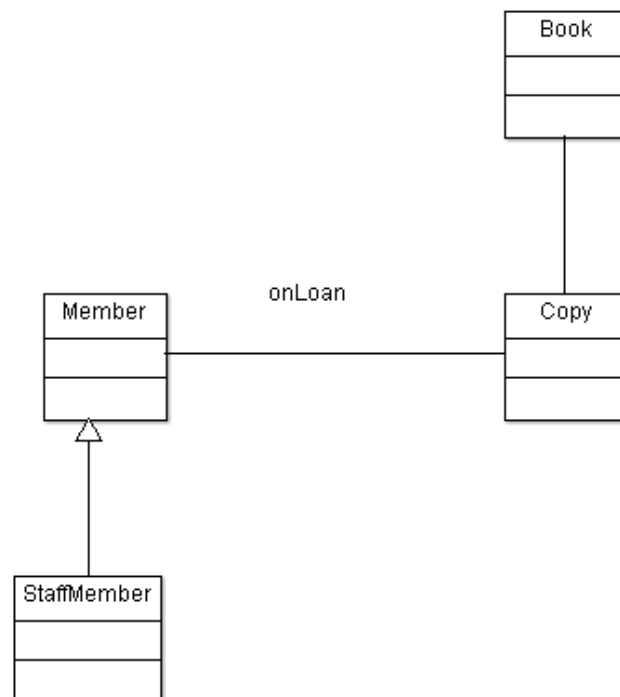
### Class & Sequence Diagrams

Open your ArgoUML file with the library activity diagram and add the following use case diagram:

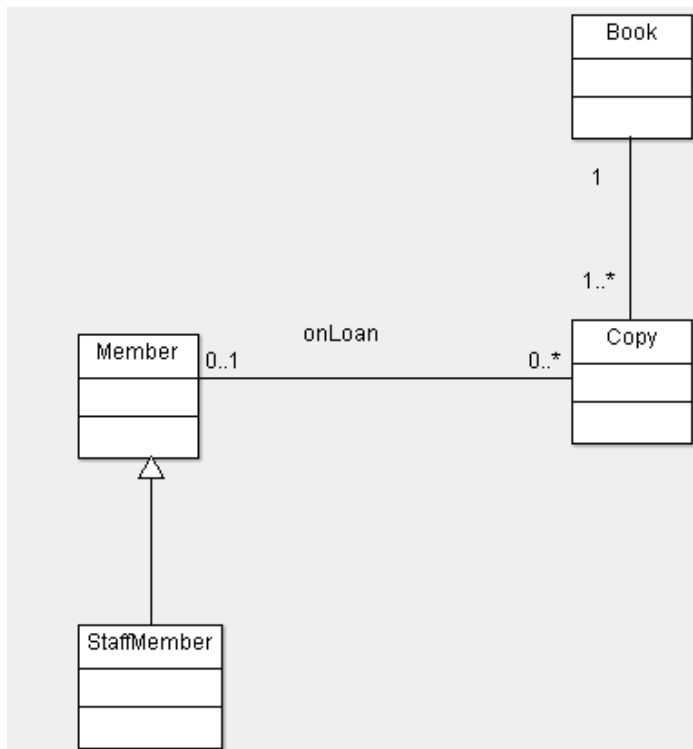


### Adding a Class Diagram

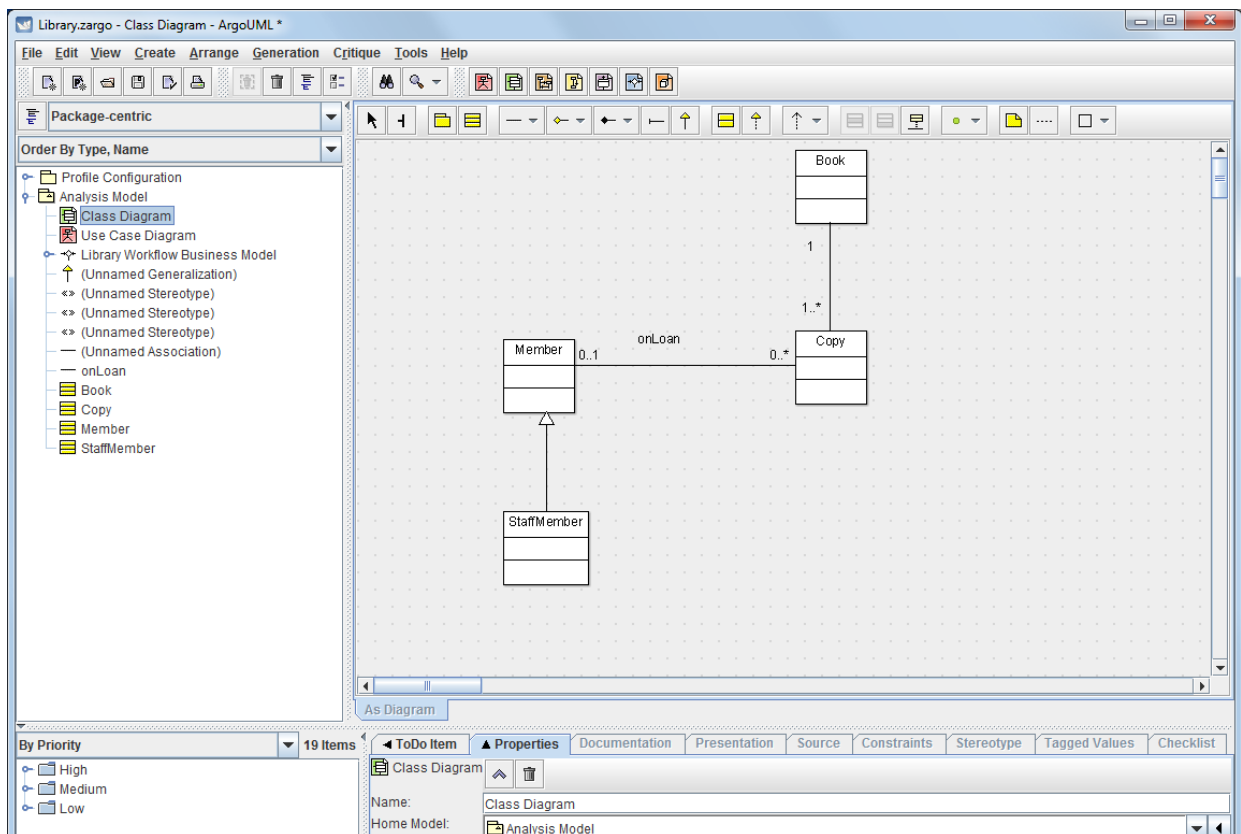
Add a class diagram to your model and draw the following class diagram. Be careful to distinguish between **generalisation** and **associations**.



Select the association between *Book* and *Copy*. Right click on it and add multiplicities as shown below. This might be tricky. May be easier to type they multiplicities in, they go in rectangular box rather than square one.



ArgoUML should now look like:

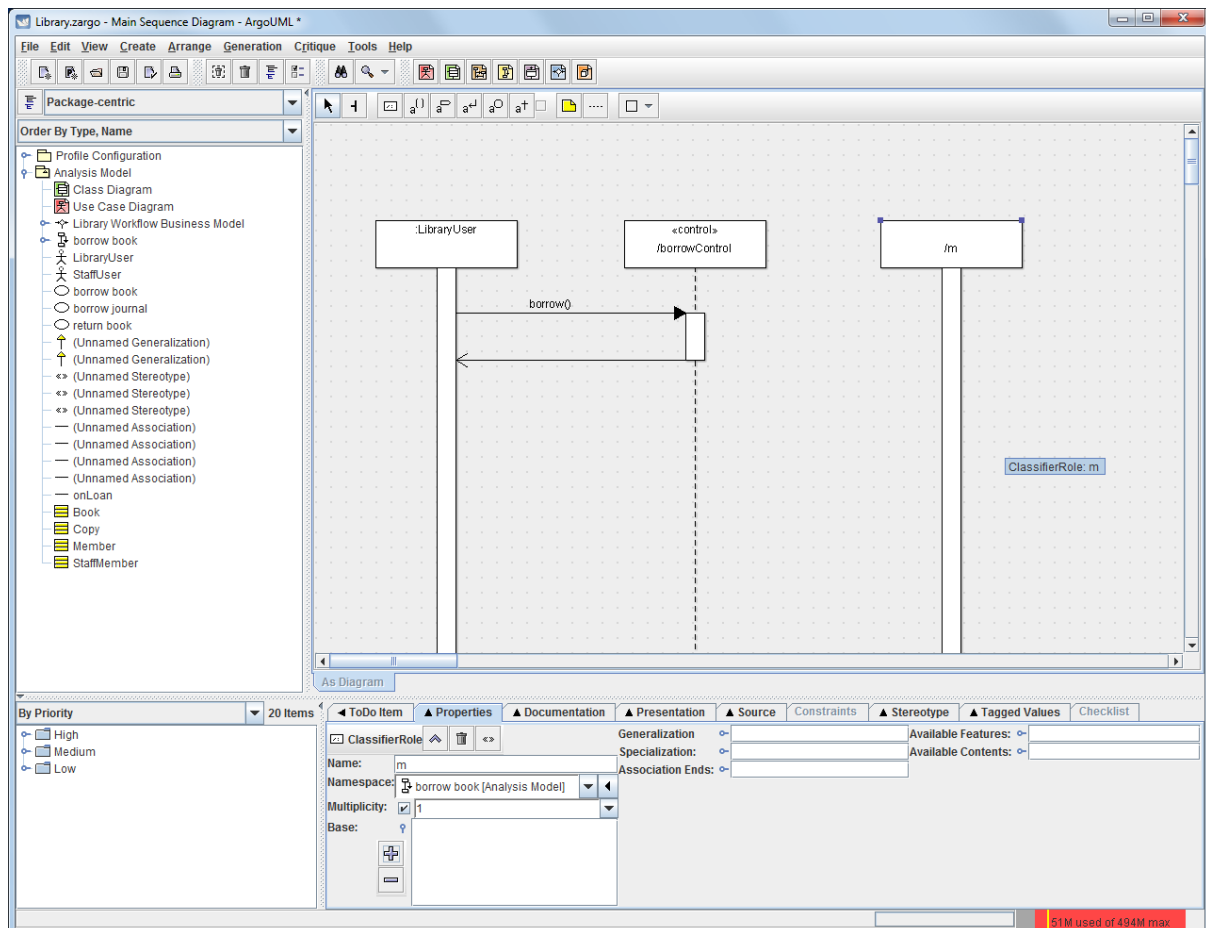


## Adding a Sequence Diagram

Add a sequence diagram to your model called borrow book. Then drag and drop LibraryUser actor onto your sequence diagram. Next using the third icon from left side of toolbar (New Classifier Role), insert an object named **borrowControl**.

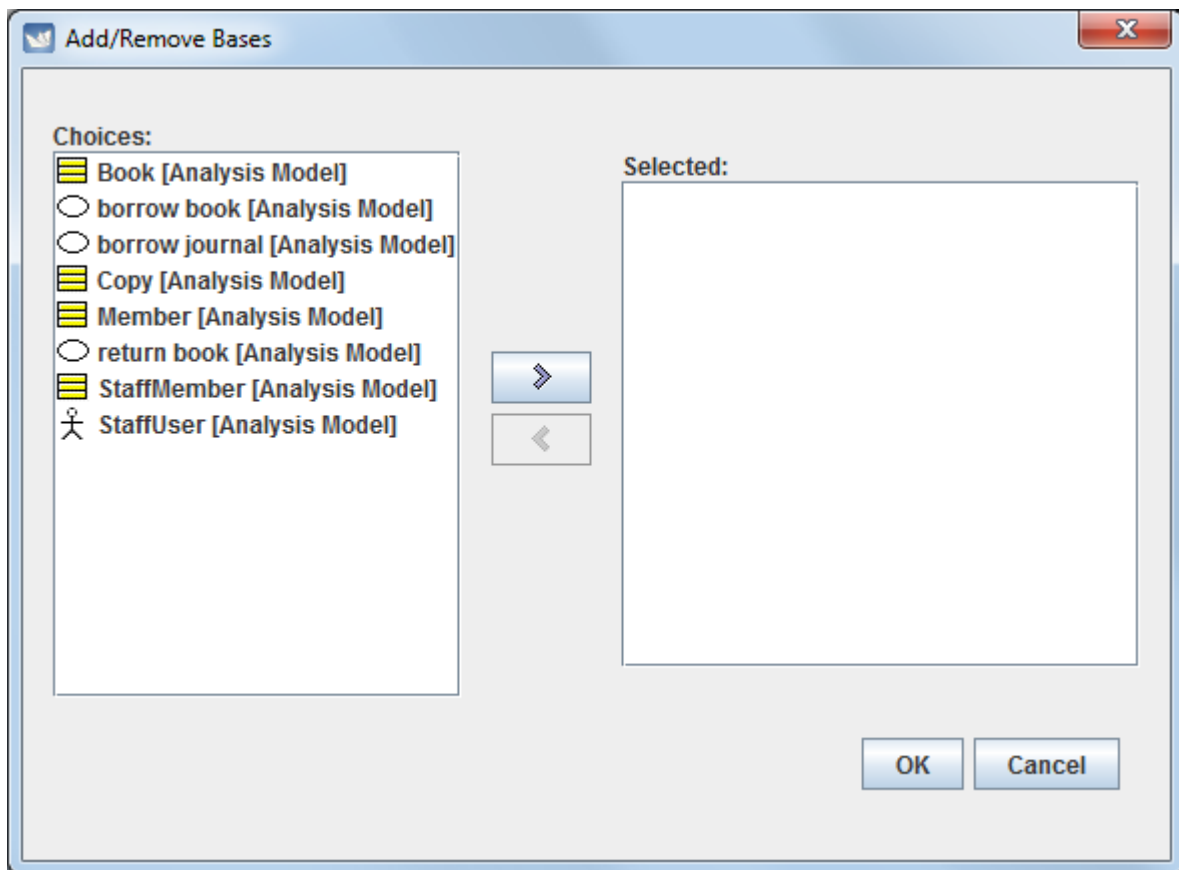
Using 4<sup>th</sup> icon (New Call Action), add a message from LibraryUser to borrowControl as shown below.

Create another object called m as shown.

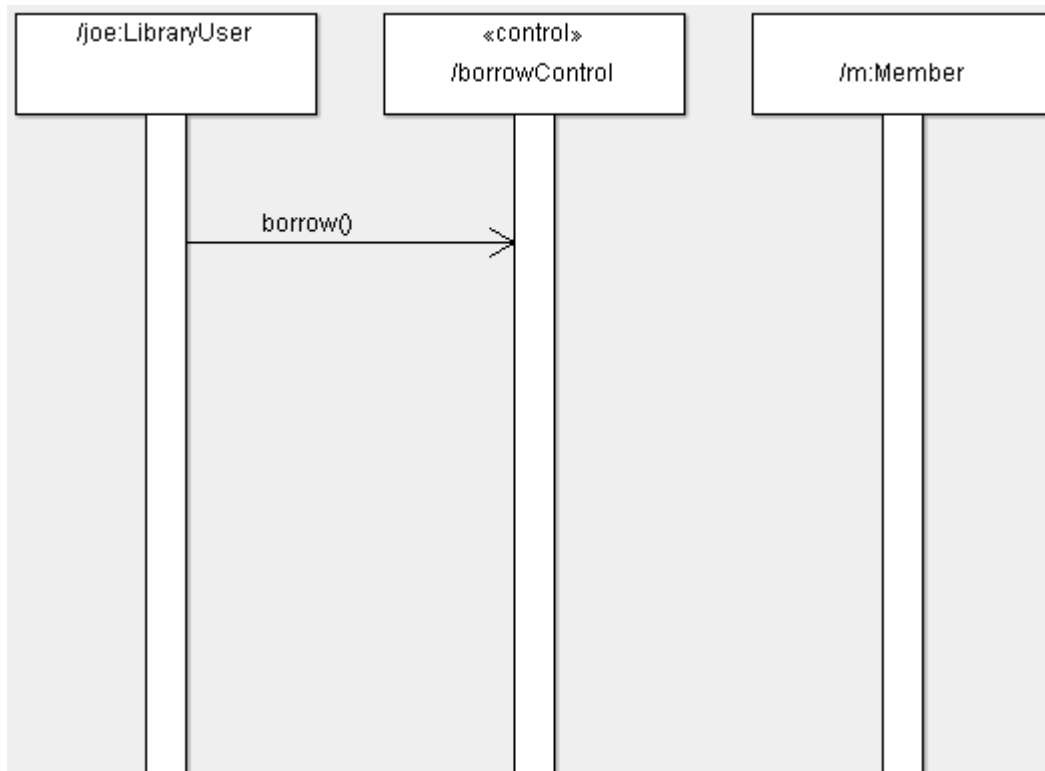


Select object **m** so that its Properties tab is visible.

In the Base field of the Properties tab for object named m, click on the + symbol. Should get:

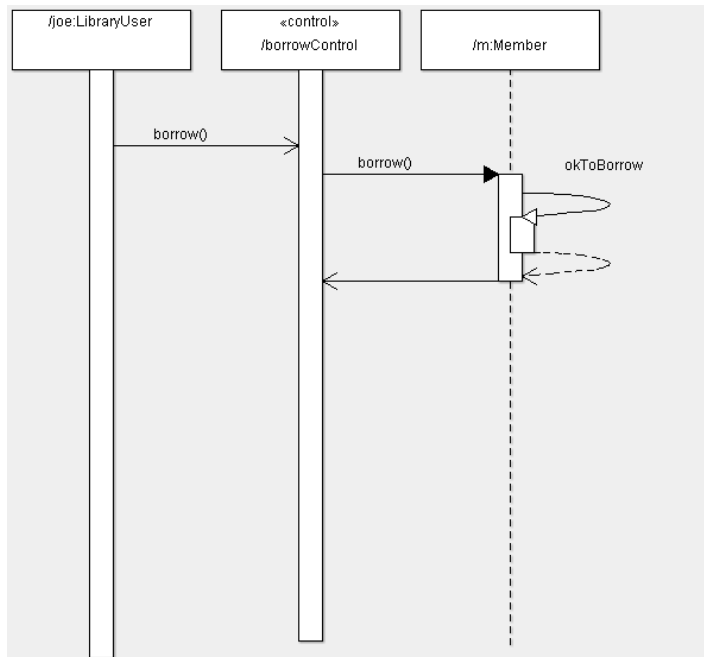


Select and add **Member** as the class for object **m**. Should now have something like:

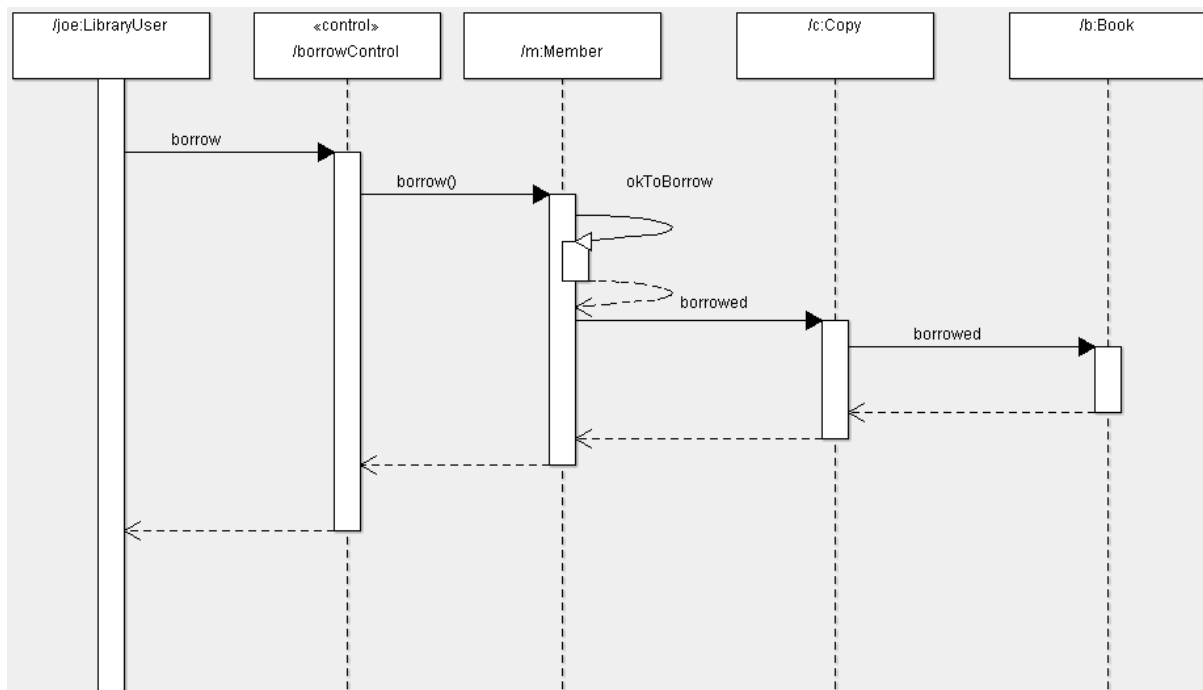


Send a **borrow** message (using New Call Action icon) from **borrowControl** to **m**.

Then get **m** to send a message to itself, this is tricky. Demoed in class.



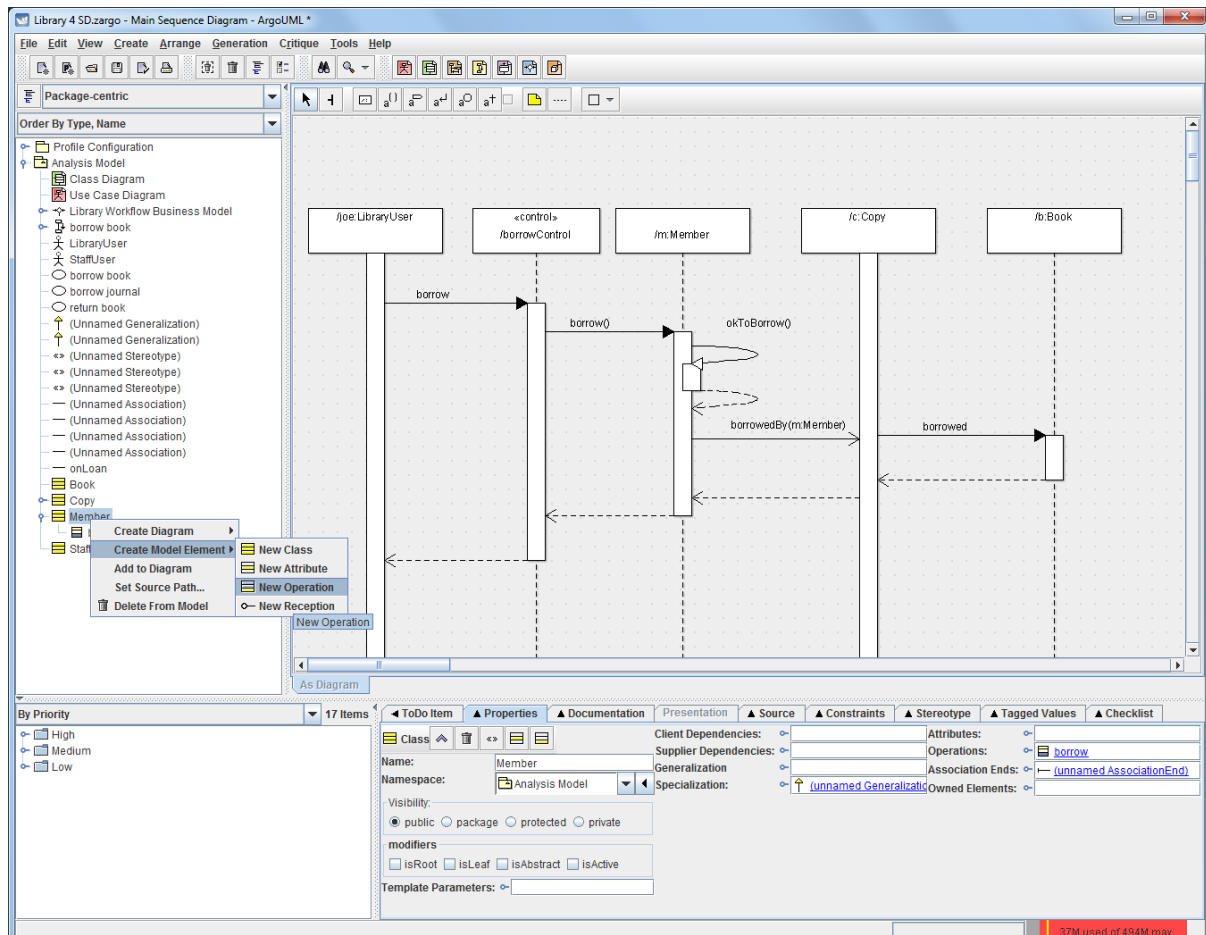
Using **New Classifier Role** icon, add two more object named **c** and **b**. Assign them the classes Copy and Book as shown next.



## Adding Operations and Data Members

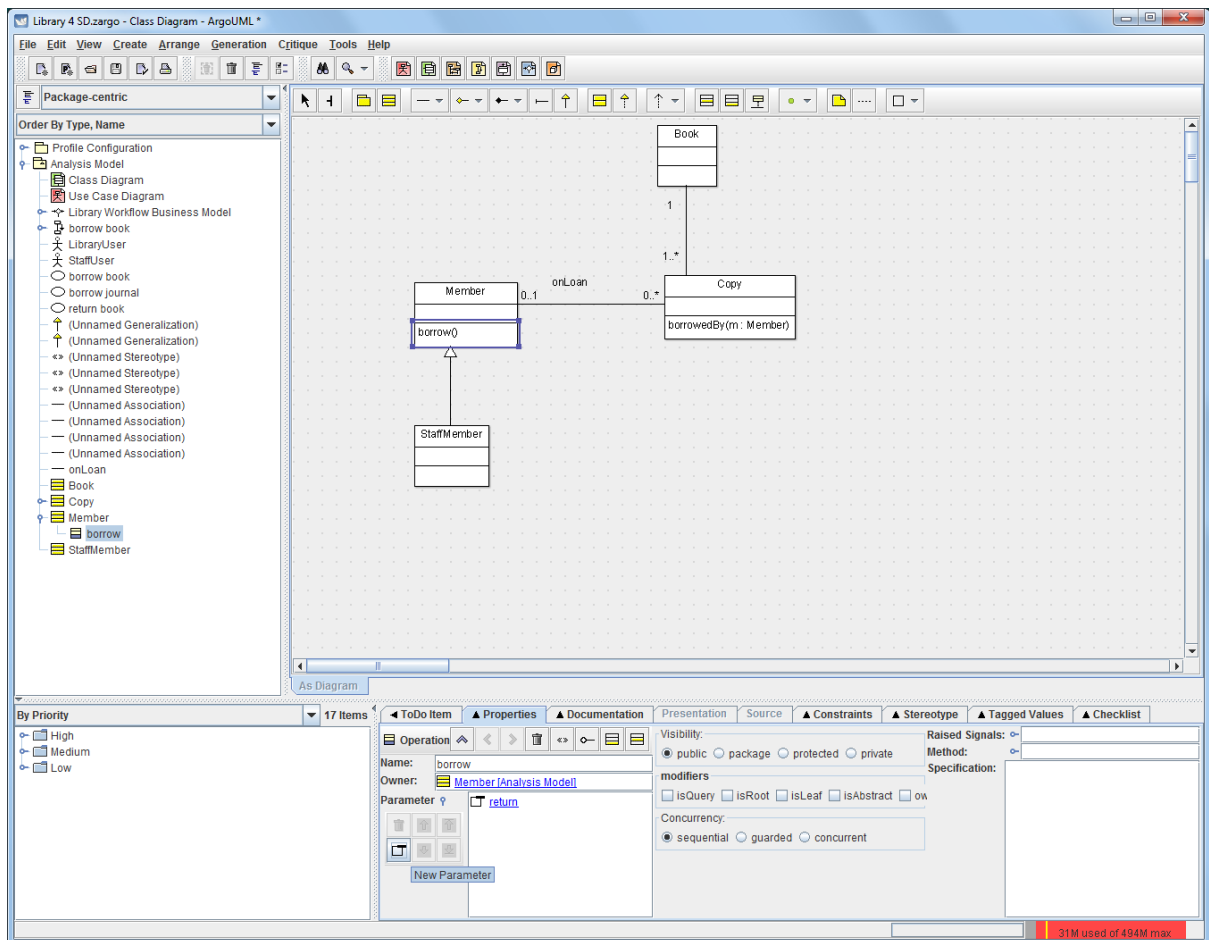
The sequence diagram gives you an idea of the messages an object might receive, the next step is to create operations or methods to respond to these messages.

We wish to add 3 methods to Member class. Select it in browser window of ArgoUML, right click on it and select **Create Model Element | New Operation** and name it borrow.

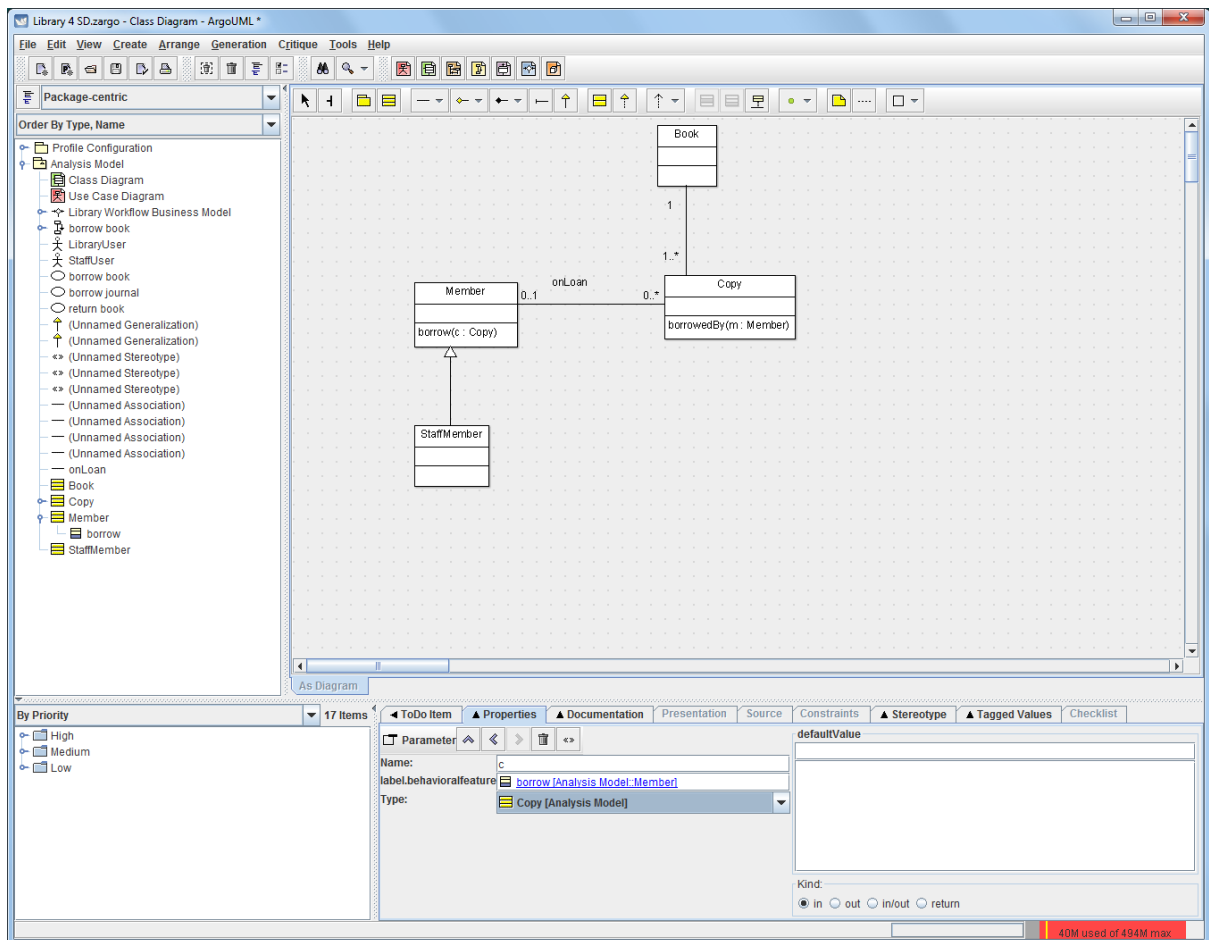


Next we wish to add a parameter c of type Copy to this method (operation). Select the operation in the browser window or in the class diagram so that the Properties window appears.

Click on Paramter icon in Properties window to get:

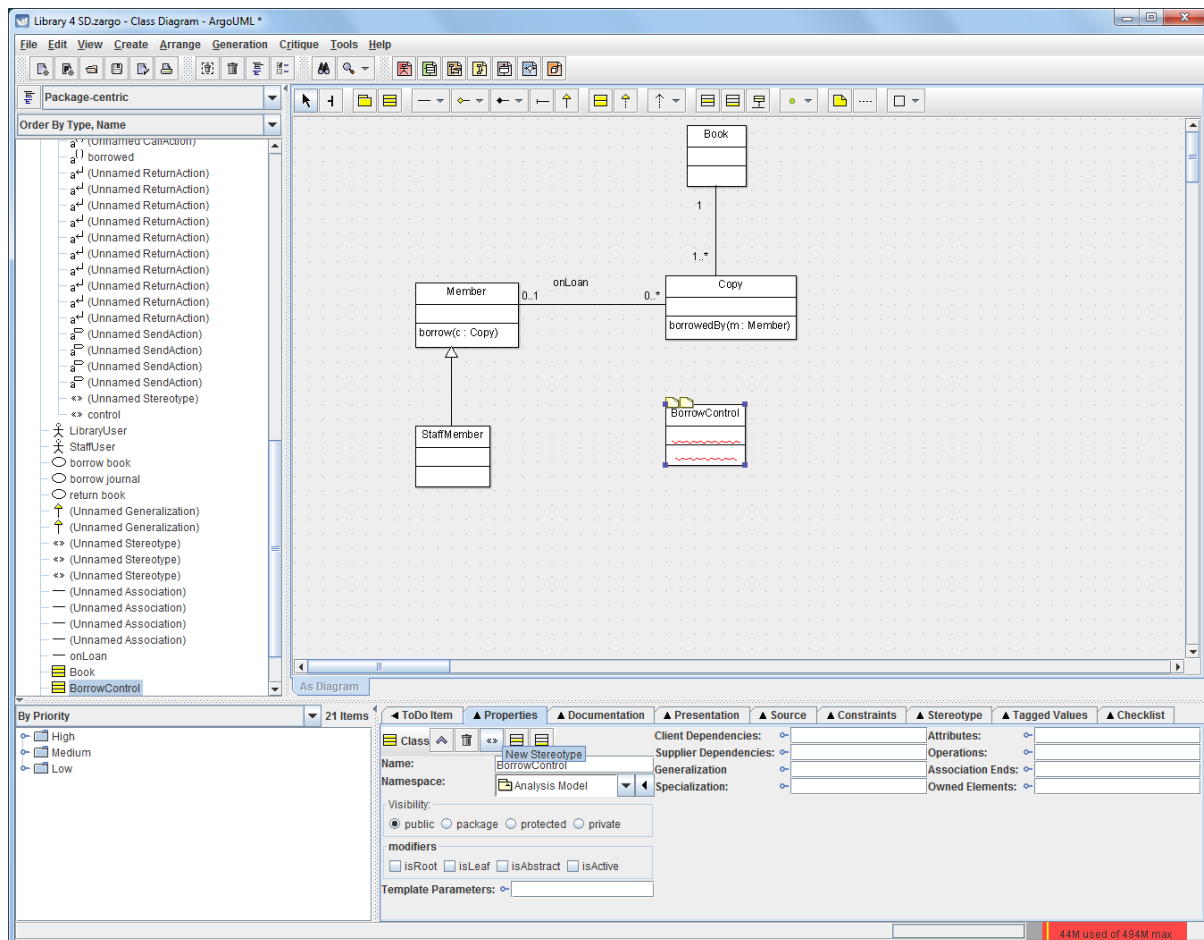


Then click on New Paramter icon, name it **c**, and select its Type attribute to be **Copy** as shown next.

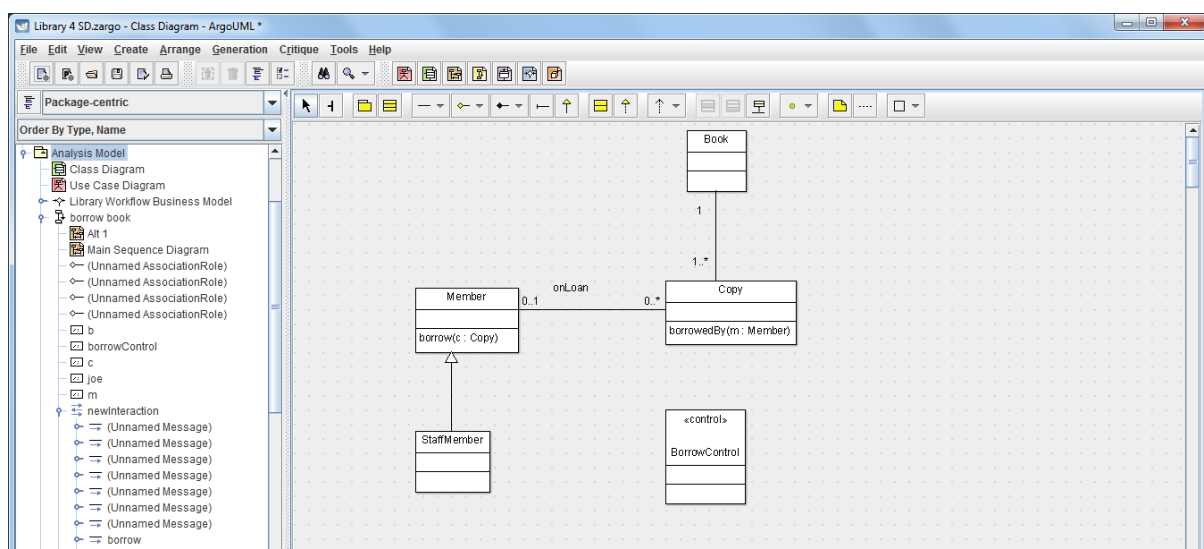




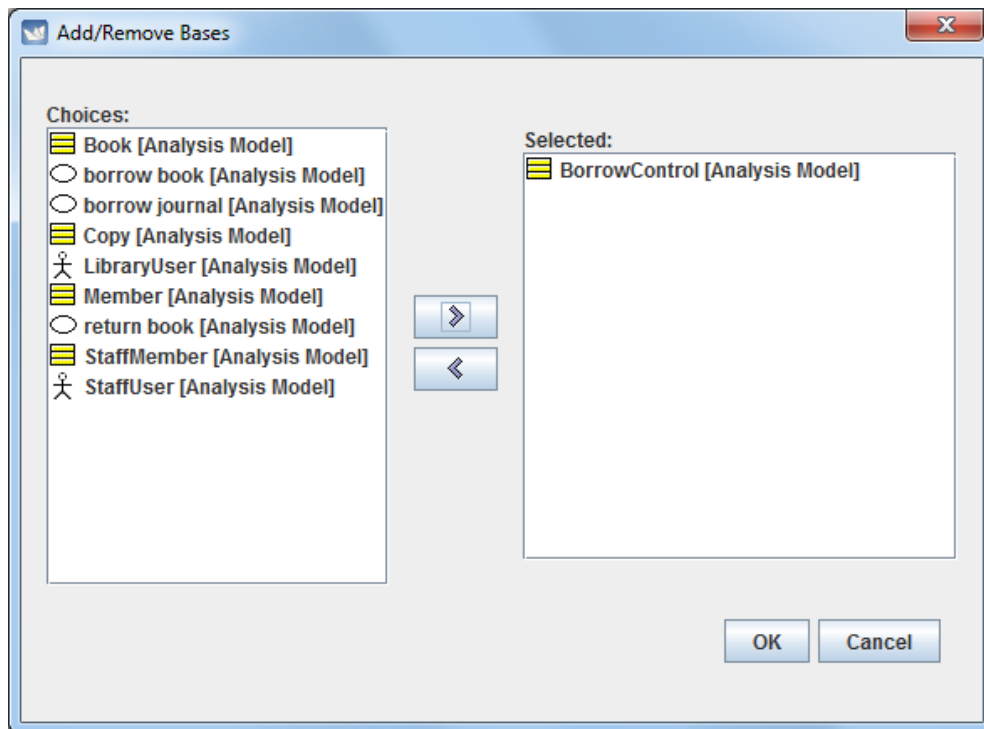
Add a new class called BorrowControl to the class diagram, click on the New Stereotype icon in its Property window:



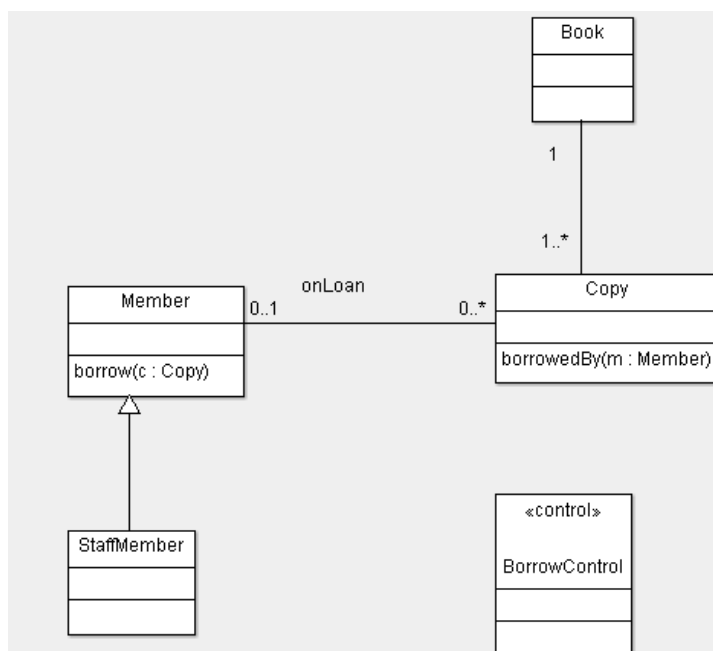
Name the stereotype control to get:



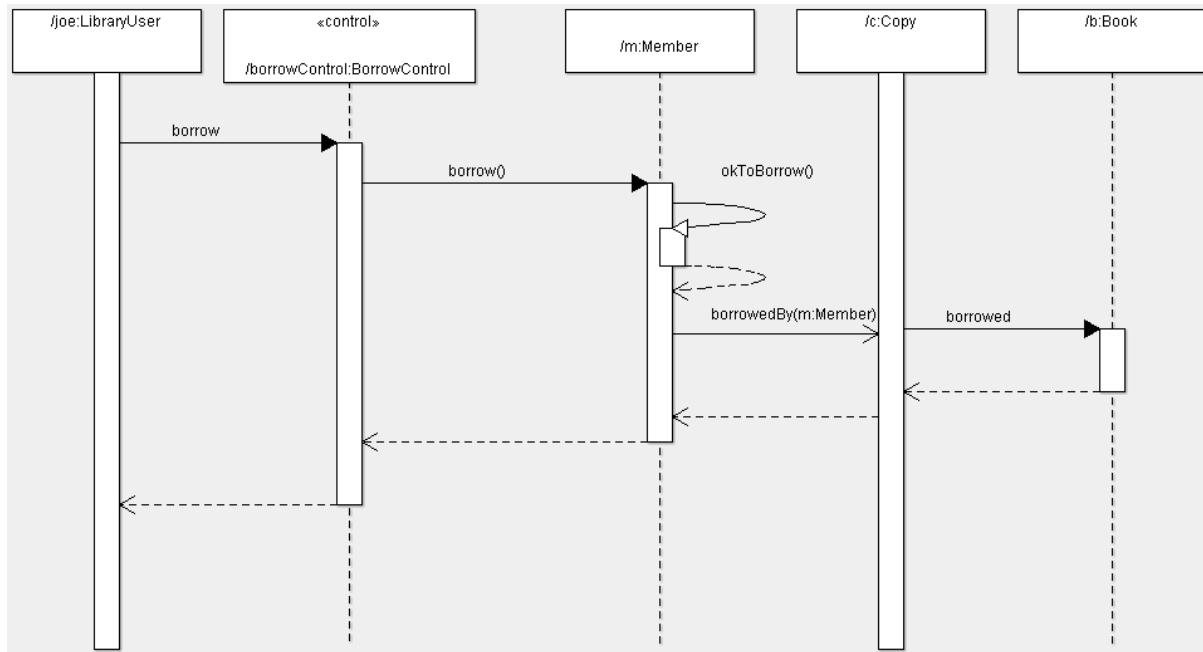
Select borrowControl object in the sequence diagram. Click Base attribute in its Properties window and add BorrowControl as its base class and click ok.



Now have

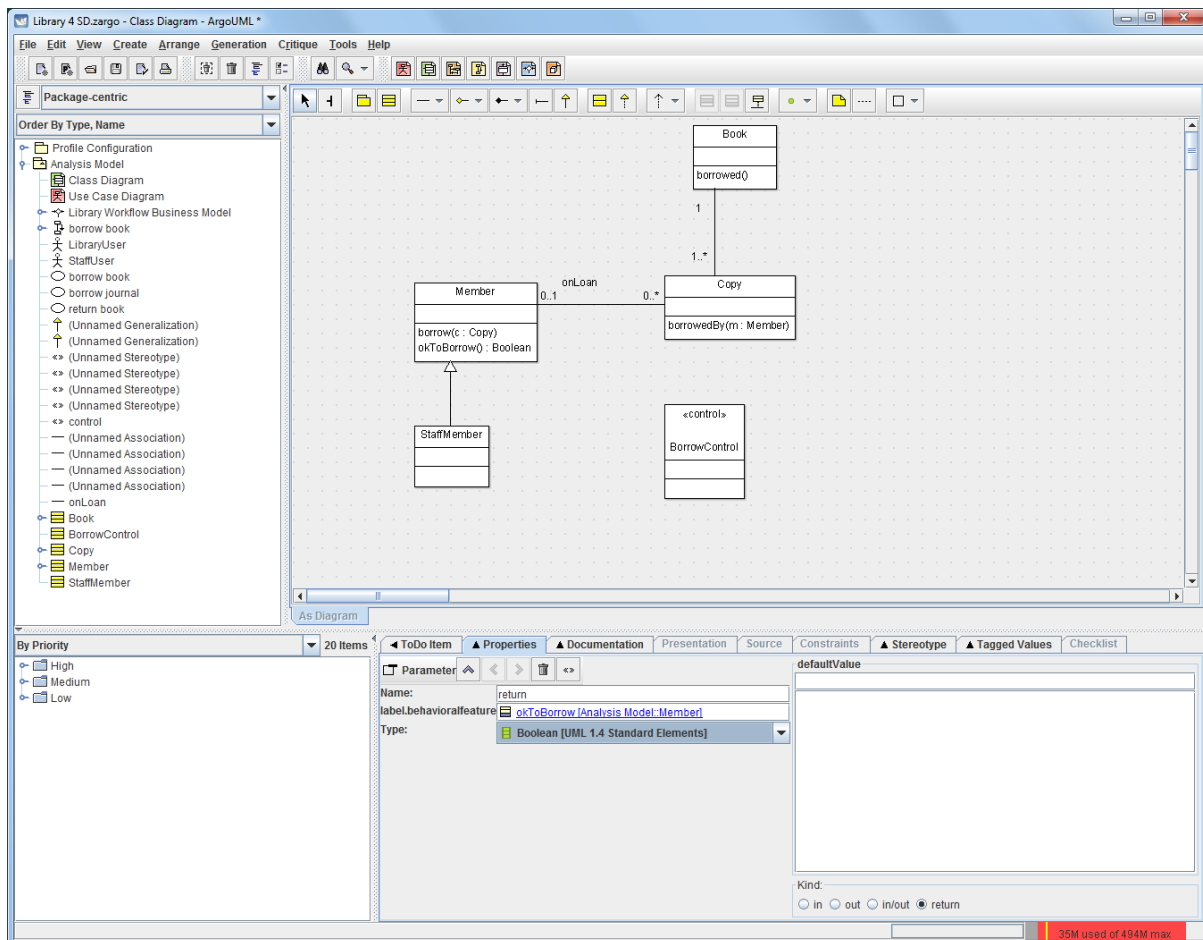


and

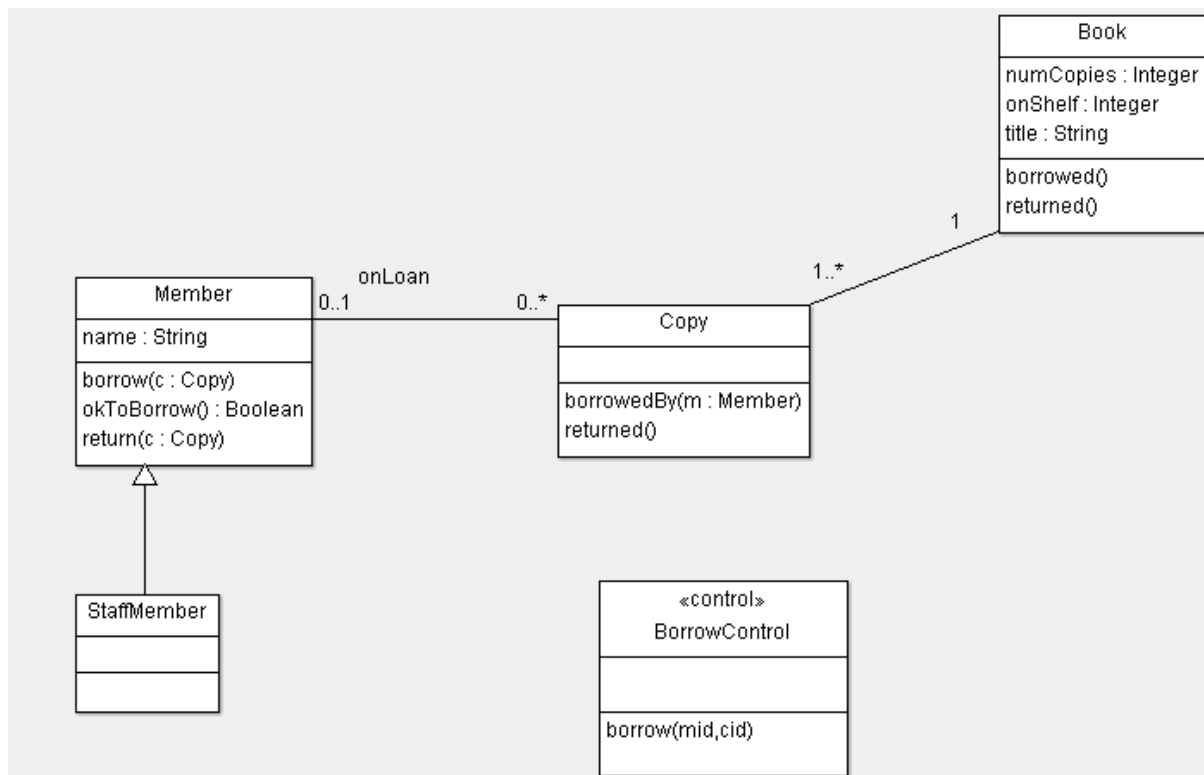


### Adding a return type to an operation

Add `okToBorrow()` operation to `Member` class. In Properties window of this operation, double click on the **return** paramter and select **Boolean** as shown below:



Next, complete adding operations and attributes to complete your class diagram along the lines of:



## Linking sequence diagram messages with operations

Select the first borrow message and right click on it, then select **Operation** and **borrow(mid, cid)**. repeat this for the other messages. Unfortunately ArgoUML doesn't show the operation signatures on the sequence diagram.

