# Introduction to USE

USE is a system for the specification of information systems. It is based on a subset of the Unified Modelling Language (UML)

A USE specification contains a textual description of a model using features found in UMLclass diagrams (classes, associations, etc.).

Expressions written in the Object Constraint Language (OCL) are used to specify additional integrity constraints on the model.

A model can be animated to validate the specification against non-formal requirements. System states (snapshots of a running system) can be created and manipulated during an animation. This is done with SOIL For each snapshot the OCL constraints are automatically checked. Information about a system state is given by graphical views.

OCL expressions can be entered and evaluated to query detailed information about a system state.

# The USE Environment

The use package is a Java package that opens a USE command line window and creates a Graphical User Interface.

Use Specifications are textual and can be entered in the command window. These textual entries can then be displayed in the GUI.

Once a USE specification file, \*.use, has been created it can be modified using notepad.

Warning There is no error checking of \*.use files

## USE Specification File Error checking

There is none.

If there is an error introduced by direct editing of the USE specification file, \*.use, the model will not load and no error messages are generated.

This can be avoided by entering USE specifications at the USE command window. The session can then be exported, saved, to a USE specification file, \*.use.

Extraneous or unwanted items can be commented out after the session is over by adding “—“ in front of the unwanted items in the USE specification file, \*.use

# Install and Start USE

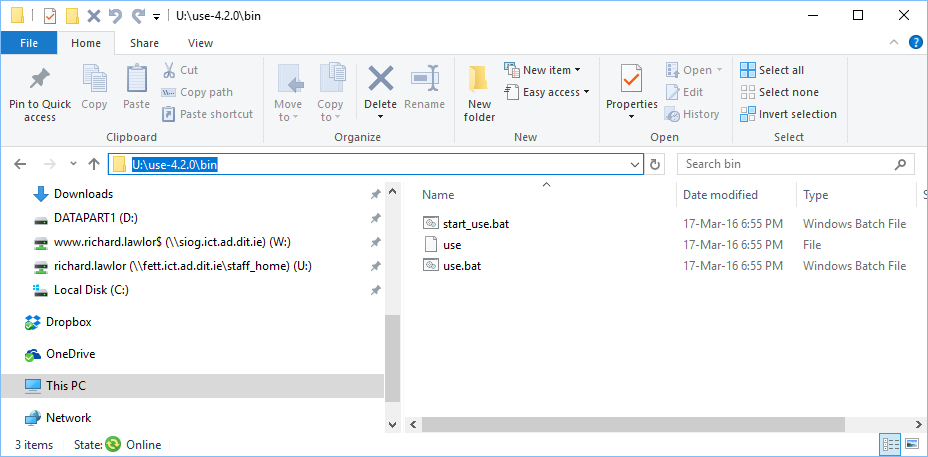
USE4.2.0 can be downloaded from the web.

The zip file needs to be extracted in the root directory of your machine in order for the batch files to run correctly. Follow the read me directions.

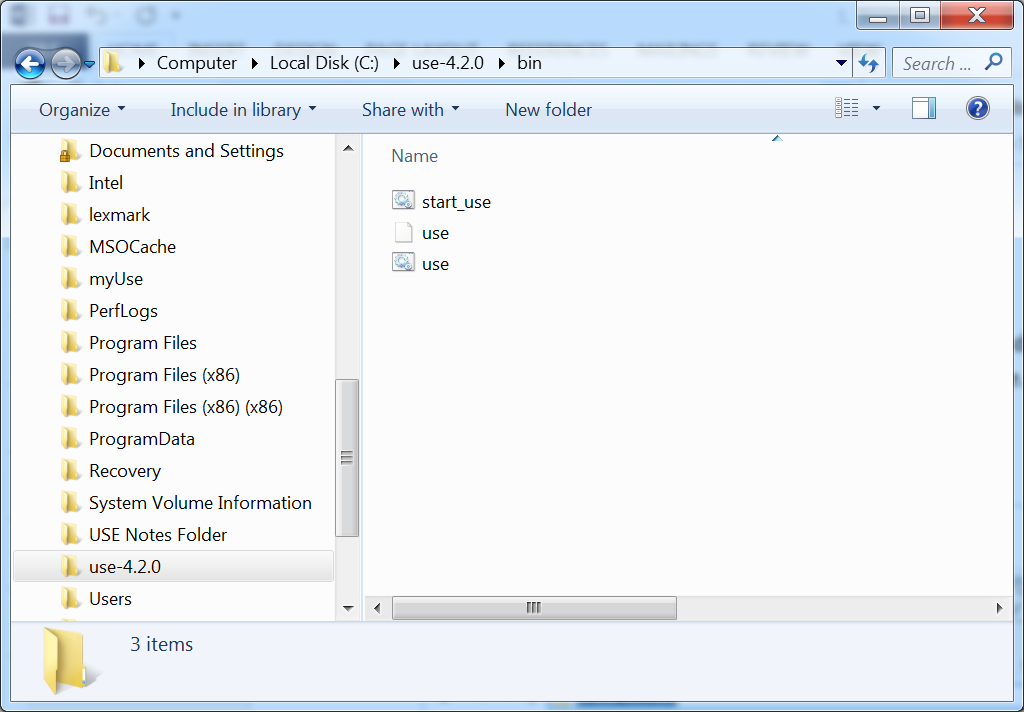
The path to Java has to be included in the system path.

Once you have USE extracted from the zip file, use Windows **File Explorer** to view your U: drive. Then navigate to the bin subfolder of the USE folder, i.e. to U:\use-4.2.0\bin for this particular version of USE.

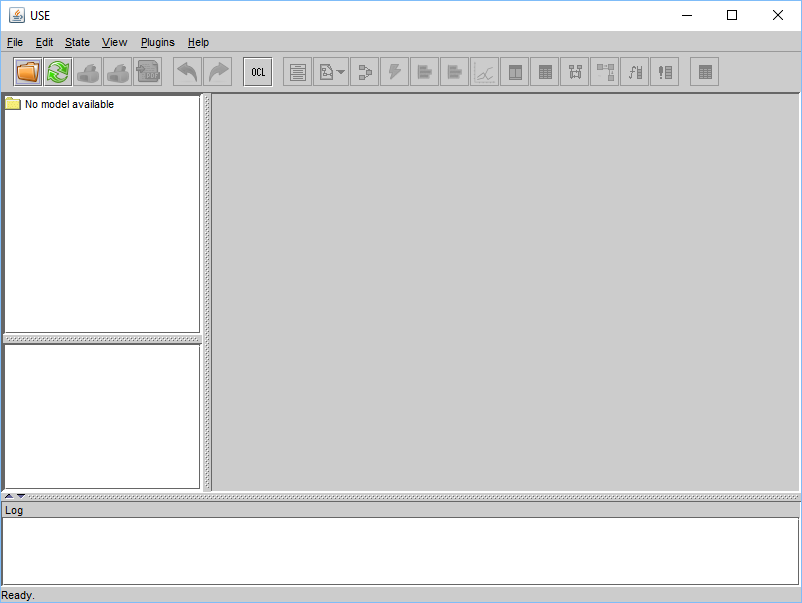
You will see something like:



Or in windows 7



Double click on **start\_use.bat**. For this to work, the Java bin folder must be listed on the **Path** environment variable. USE GUI looks initially like:

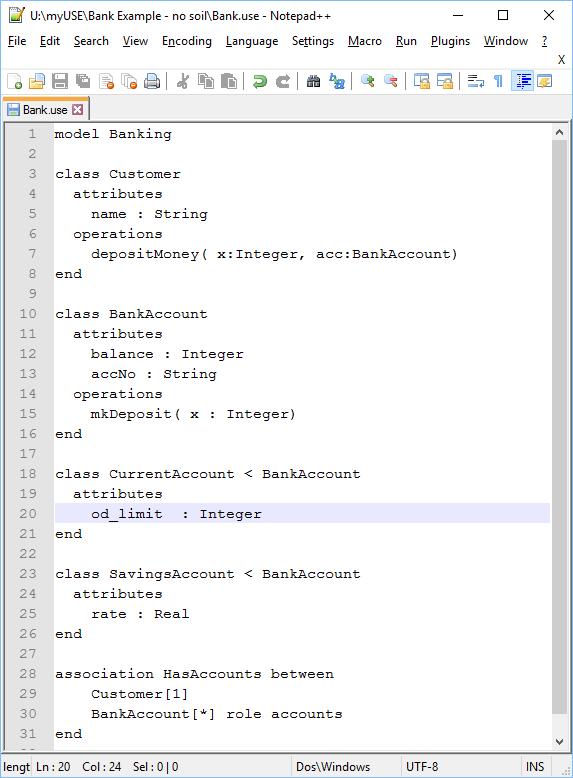


## A USE Specification

Next step is to write a USE specification.

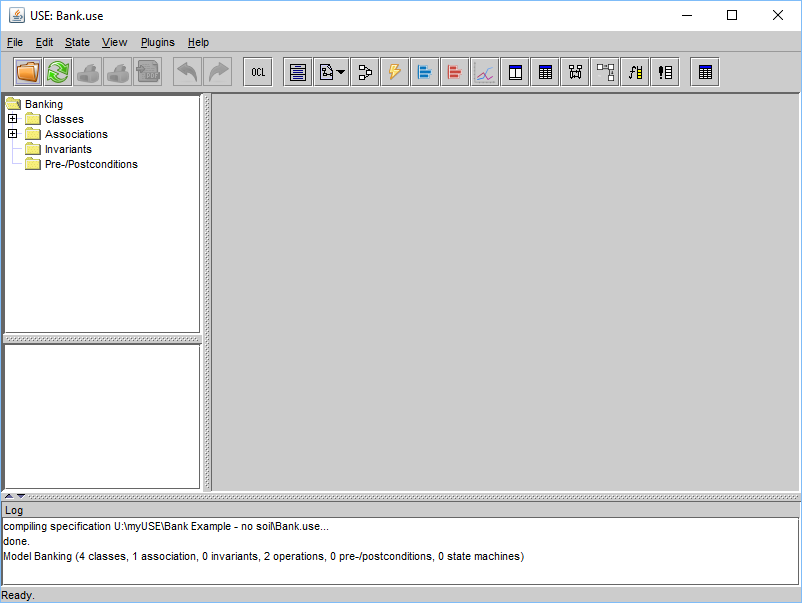
Create a folder called **myUse** on your **U:** drive and a subfolder in this called **Bank Example**.

Then use Notepad++ to write the USE code shown below and save it to this folder under the name **bank.use**.

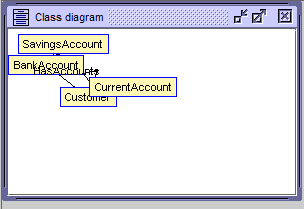


## Load Specification into USE

Use the USE menu **File | Open Specification** to load your USE code. If successful, USE GUI will look like below. If not you have errors in your code.

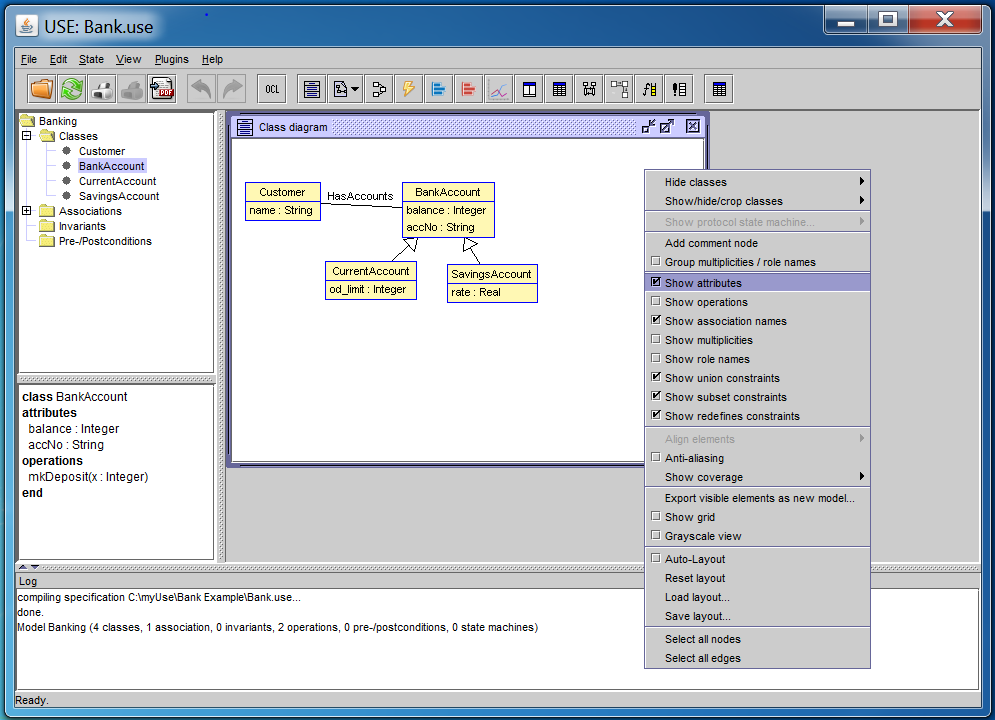


To see your class diagram in USE, click on the **Create class diagram view** icon . Class diagram like blow will then appear.

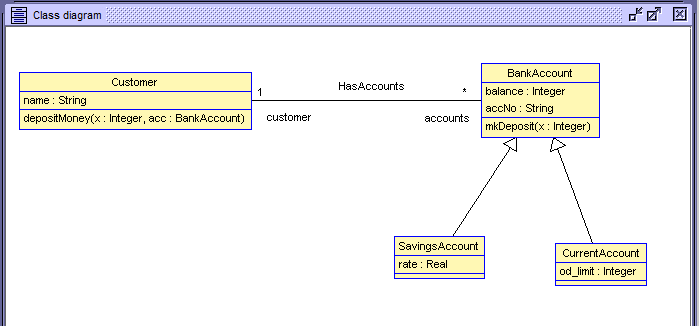


Resize this window to make it bigger and right-click on it to see context sensitive menu:

Select a number of the show options so that attributes, operation, multiplicities and role names are visible.



Then reorganise your class diagram layout to something like:



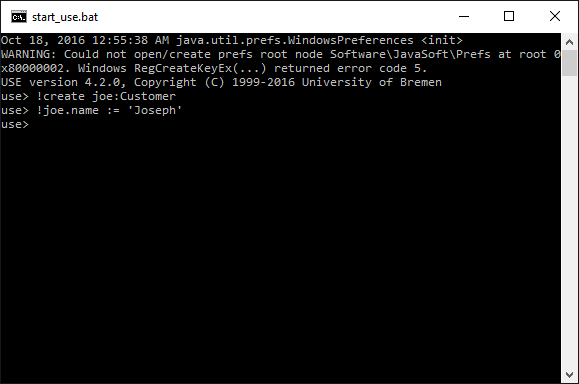
Then save this layout by right-clicking and selecting **Save layout**. Save it with the name bank. You can reload this layout in future lab sessions.

## Create Objects to Animate your Specification

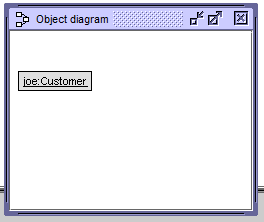
On the **Command window** that starts with use type in the commands

!create joe:Customer

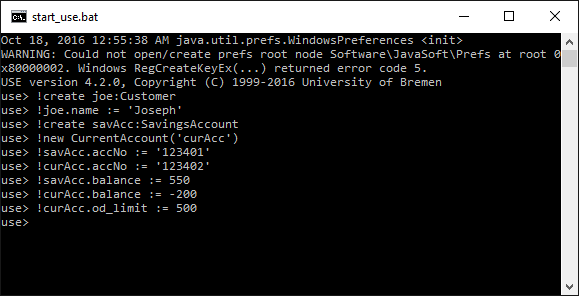
!joe.name := ‘Joseph’



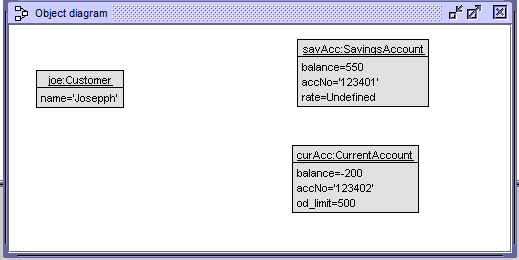
In Use click on the **Create object diagram view** icon  to get an object diagram like:



Enter more USE command line commands as shown next:



Then get the object diagram to show attributes and reorganise its layout. Right-click and select **Save layout** and save the layout with the name bank.

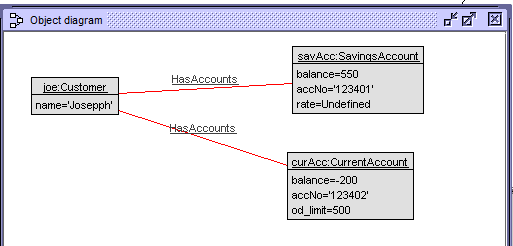


Finally we wish to link these bank account objects with joe. Use the command line to do this with:

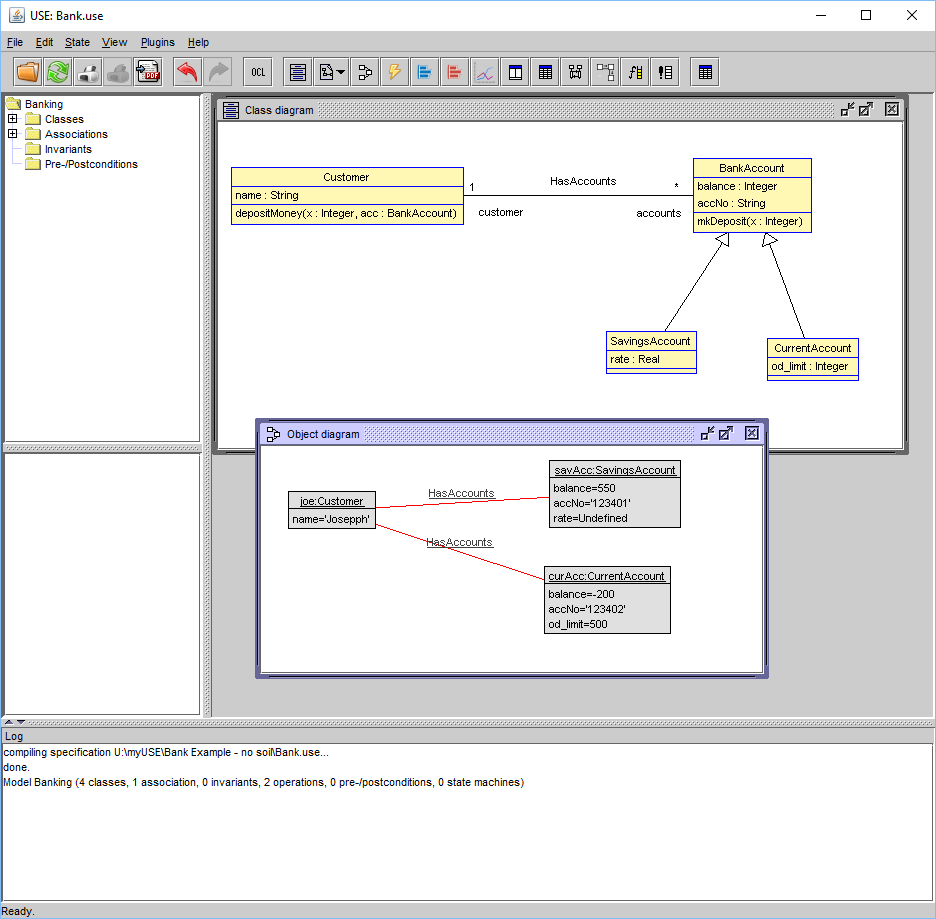
!insert (joe, savAcc) into HasAccounts

!insert (joe, curAcc) into HasAccounts

Object diagram will now look like:

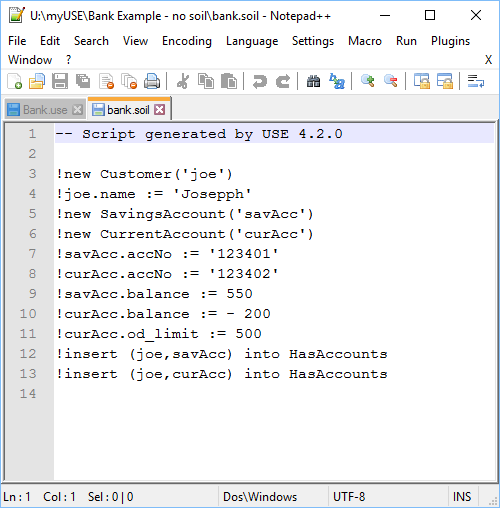


USE itself may look like:



Next use the menu **File | Save script (.soil)** to save your object creation and modification commands. Save them to a file called **bank.soil**. T

his will allow you to reload your object at a later date. Using Notepad++ have a look inside this .soil file. Should be like:



## Exercise

Create a customer called anne and a savings account for her with a balance of 1100. Link the 2 objects.