






## Practical 5: Adding a State Diagram to a Together project

You are going to add a simple state diagram to the library project from practicals 2 and 3, or you could use these instructions with any existing UML project in Together.

1. Launch **Together** in the usual way and open your library project (possibly **Practical2**).
2. You are now going to create a **State Diagram** showing the possible situations in which a **Book** object might find itself and the relationship between those situations (we say "*the states in which the Book might find itself, and the transitions between those states*"). On the main menu bar, choose **File** followed by **New** followed by **Diagram**. In the **Create Diagram** dialog box select **State Machine** as the **type** and give it a suitable name and click on **Finish**. A state diagram pane is created.
3. To the left of the pane is a revised **Diagram Toolbar** which now shows a palette of icons for the different kinds of item that you can add to a state diagram. By selecting the **State Machine** icon  on the toolbar, create a **state machine**. *In this box*, you will draw the state diagram (or you can right-click in the box's title bar and select **Open in Full Screen**).
4. By selecting the **Initial** icon  in the toolbar, then clicking in the diagram pane (perhaps towards the left side), create a **start state** (black dot)
5. Then select the **State** icon  in the toolbar, and click in the diagram pane (perhaps around the middle). The state name is initially highlighted ready to enter a new name: name it **onShelf**. Similarly, create state **onLoan** towards the right.
6. Now select the **Transition** icon  and draw a transition from state **onShelf** to state **onLoan**.
7. The easiest way to give this transition the name **borrow** (and to add guard conditions and actions) is: Click on the transition arrow and look at the **Properties** pane (or right-click on the arrow and select **Properties**). Then edit the value of the **label** property to be **borrow** and press **Enter**. The **label** property can contain any free text, so you can type a guard and actions as well. Once the label is shown in the diagram, it can also be selected and edited there (click twice with a pause between, like renaming a file in a folder window).
8. The "correct" way to name a transition is to select it, and enter the desired name into the **trigger** property in the **Properties** pane. Similarly, a guard should be entered on the **guard** tab (at the left of the Properties pane) in the **body** property – just the Boolean condition, the [ ] are added automatically in the diagram view. If using this approach, a simple way to enter actions is to select the trigger/guard text shown on the diagram, and type / followed by the actions, then Enter (the action text is held as a **Custom** property called **\_effect**).

9. The "correct" way to add an action to a transition is a more awkward: It is necessary to build an **Activity diagram**, in which named activities can be given a detailed description in terms of flows of **actions**, and a transition in a state diagram can then be linked to a named **activity** by clicking in its **effect** property and choosing from the offered options in the  dialogue box. You can try it!
10. Make the two state icons larger by clicking in their title bars and dragging on the edges. Now add a transition named **return** from **onLoan** to **onShelf**.
11. Draw a transition from the **start state** to **onShelf** and give it a suitable name.
12. Add *actions* to the transitions between **onShelf** and **onLoan**.
13. Consider that a book may be *renewed*: extend your state diagram to represent this.
14. Consider that a book may become *overdue* after a certain time has elapsed: a fine must be paid when it is returned, and it may not be renewed once overdue. Add suitable states/transitions to model this.
15. Consider that a book may be *requested* by another library member: If it is not currently on loan, then it may only be subsequently be borrowed by that member. If it **is** currently on loan, then it may **not** be renewed. Add suitable states/transitions to model this. Maybe requests can be *cancelled* too!
16. The last step for the library is to add a **stop** (or **final**) state. Consider from where there should be a transition(s) to the **stop state**, add it/them and give them suitable names.

End of worksheet