University of Stirling
Computing Science and Mathematics
CSCU9P5 Software Engineering I **Autumn 2018**
Practical 3 (for practicals in the week starting 8 October 2018)

You are going to add use case, sequence and communication diagrams to the project that you created in the second practical class – *so if you did not finish that worksheet then you should complete it now*.

Three useful quotes about Actors and Use Cases:

- "Use case diagrams can be defined as behaviour diagrams that are used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). **Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.**" (from http://creately.com/diagram-type/article/basics-uml-usecase-diagrams)
- "An actor is a person, organization, device, or external software component **that interacts with your system."** (from http://msdn.microsoft.com/en-us/library/vstudio/dd409432.aspx)
- "*Actor* isn't really the right term; *role* would be much better. Apparently, there was a mistranslation from Swedish..." (from M Fowler, UML Distilled, 3rd ed, p100)

1. To start the **Together** system, click on the **Start** icon and type **Together** in the search box. It should open in the way that you last left it. If not, open your project **Practical2**, then choose **Library** within **Practical** in the **Model navigator** pane.

Creating a Use Case Diagram

2. You are now going to create a **Use Case Diagram** for the interactions that a **Borrower** might have with the simple library system. On the Main menu, choose **File** followed by **New** followed by **Diagram**. In the **Create Diagram** dialog box select **Use case** as the **type** and give it a suitable name (e.g. **Borrower**) and click on **Finish**. A use case diagram pane is opened.

3. To the left of the pane is a revised **Diagram Toolbar** which now shows as icons the different kinds of item that you can add to a use case diagram. Hold the mouse cursor over the **Toolbar** icons to find out what each icon does.

4. Create a **`Borrower`** actor to the left of the diagram.

5. It is helpful to highlight the diagram components that are to be implemented by the intended eventual system by enclosing them in a *system boundary box*: The UML terminology is that the intended system is the "Subject" (of the design activity). There is an icon in the **Diagram Toolbar** for drawing a Subject (a rectangle): Draw a system boundary box in the right side of the use case diagram, with the **`Borrower`** actor outside it to the left. The use cases that you add below should be drawn within the boundary. Give the system the name **Library**.

6. Add two use cases **`BorrowBook`** and **`ReturnBook`** and show that the actor communicates with each of these use cases. Communication is added to the diagram in a similar way to association in a class diagram.

7. You can move between the diagrams **Library** and **Borrower** by clicking on their tabs at the top of the **Designer** pane.

8. Create a second **Use Case Diagram**, as above. This will be for the interactions that a **Librarian** might have with the simple library system. *Improvise some details!*

9. When a use case is selected in the diagram, you can enter a *description* of that use case in the **Description** part of the **Properties** pane (text formatting is allowed). Add brief descriptions to each of your use cases. There is no way to show the descriptions in the diagrams, but when you generate HTML documentation at the end, you will see the descriptions appearing there.

Creating a Sequence Diagram

10. You are now going to create a **Sequence Diagram** showing how a **Borrower** borrowing a book will be dealt with by the system**.** On the Main menu, choose **File** followed by **New** followed by **Diagram**. In the **Create Diagram** dialog box select **Sequence** as the **type** and give it a suitable name (e.g. **Borrowing**) and click on **Finish**. A sequence diagram pane is opened. Right click on **SD Interaction** and in the drop down menu chose **Open Full Screen Sequence Diagram**.

11. To the left of the pane is a revised **Diagram Toolbar** which now shows as icons the different kinds of item that you can add to a sequence diagram. You are going to create a simple sequence diagram with an actor and two objects which will be represented as lifelines.

12. Select the **Lifeline** toolbar icon and create a new lifeline in the diagram pane towards the left side. It will have a default name **a**, initially selected and ready to change. Change its name to **`aBorrower`** and right click on it. In the drop down menu, select **Choose Type** and then Borrower (which is there as an Actor because you created it in the Use case diagram). You will see that you now have the lifeline for an actor.

13. Select the **Lifeline** toolbar icon and create a new lifeline to the right of the **`Borrower`** actor lifeline, changing its name to **`aCatalog`**. Click with the right button on **`aCatalog`** and on the drop down menu select **Choose Type** and then choose **`Catalog`** (which is there as an Class because you created it in the class diagram). In the same way, create an object **`aBook`** of class **`Book`**.

14. Draw an operation invocation (**Message** icon) between **`aBorrower`** and **`aCatalog`** – draw straight across from **`aBorrower`**'s lifeline to **`aCatalog`**'s. A pop-up window appears – chose the **`borrow`** operation. You will find that parameters have automatically been added, and that there is now a white *activation* rectangle in **`aCatalog`**'s lifeline representing an *operation execution*.

15. In the same way, create a **`borrow`** message between **`aCatalog`** and **`aBook`**. *The starting point of this message must be from the white activation rectangle in* ***`aCatalog's`*** *lifeline.*

16. The **Borrower** use case diagram has two use cases. You already have a sequence diagram for borrowing a book. Create a new sequence diagram for *returning a book*.

© University of Stirling 2018

Creating a Communication Diagram

17. You are now going to create a **Communication Diagram** showing the same use case discussed in step 6., i.e. showing how a **Borrower** borrowing a book will be dealt with by the system**.** On the Main menu, choose **File** followed by **New** followed by **Diagram**. In the **Create Diagram** dialog box select **Communication** as the **type** and give it a suitable name (e.g. **BorrowingComm**) and click on **Finish**. A sequence diagram pane is opened. Right click on **SD Interaction** and in the drop down menu chose **Open Full Screen Communication Diagram**.

18. To the left of the pane is a revised **Diagram Toolbar** which now shows as icons the different kinds of item that you can add to a **Communication diagram**. You are going to create a simple communication diagram corresponding to the simple sequence diagram with an actor and two objects as before.

19. Proceed as before (steps 12 - 15), taking care of translating all the information into the **Communication Diagram** settings. Where needed, you have to discover how specific features of these diagrams can be expressed by Together (eg., how can you get the numbering of operations right?).

20. Our project is now composed of several different diagrams. The **Model Navigator** pane (the left pane) contains a list of our diagrams and allows us to move from one diagram to another. Double clicking on a diagram name will open that diagram. Alternatively, select a diagram from the list with the **right** mouse button. A menu will appear; choose **Open**.

Sanity check.

21. You know that **Sequence** and **Communication Diagrams** are equivalent. Go back to the **Sequence Diagram** developed in steps 10 - 15 and right click on **SD Interaction** and in the drop down menu chose **Open Full Screen Communication Diagram** and your **Sequence diagram** is transformed into a **Communication** (collaboration) **diagram**.

22. Compare the Communication Diagram obtained by the automated translation with the one you developed yourself. Are they the same? Is there something you could have done differently? (Or have done better than the automated translation?)

Documentation

23. Choose **Documentation** from the **Project** menu and in the resulting menu choose **Generate HTML**. A dialog box will appear. Click on the box to the right of **Output Path** and navigate to the folder which you created in the first practical class to store the documentation. A browser will be launched to display the documentation of your system. Note that all your diagrams are saved as **gif** files. That allows them to be viewed in a browser and also gives a good format in which they can be included in a Word document such as the one you will be producing for your assignment.

SBJ Oct 2018