University of Stirling
Computing Science and Mathematics
CSCU9P5 Software Engineering I **Autumn 2018**

Practical 2 (for practicals in the week starting 1 October 2018)

This practical introduces the Together Architect CASE tool. By learning how to create a class diagram using Together you should try to understand the general principles about how Together works. When following the detailed steps, try to familiarise yourself with the main functionality supported by Together. Note: Together Architect is a customized version of the Eclipse IDE.

Setting up and Starting a new modelling project

1. First use Windows Explorer to create a new folder to hold your **new** *Eclipse Workspace* as follows: For efficiency reasons, your **Together** working materials should be kept on your **T:** drive. Click on the **Start** icon and choose **My Computer**. Under **Network Drives** double click on **T:**. Now create a folder called, say, **P5Practicals**. This folder will contain data used by Together to manage your projects, and the projects themselves.

2. To start the **Together** system, click on the **Start** icon and either type **Together** into the search box, and click on the **Together** that is found, or navigate via **All Programs** then **Borland Together** and then click on **Together**.

3. You will be asked to choose a workspace: click **Browse**, navigate to your new folder **P5Practicals** on **T:**, select it but *do not navigate into it*, and click OK. The **Borland Together** system is launched. Note that you should create and use the Together workspace on the T drive. Using a different drive (e.g. H) can lead to inefficiencies. **Also**, you should **not** try to use the same workspace folder for both Together and standard Eclipse as they are incompatible!

4. You will now create a project (and folder) for this practical: On the Together main menu bar at the top, choose **File** followed by **New** followed by **Project.** This gives a pop up window in which you **Select a wizard**. Choose **Modelling** and in the list choose **Java Modelling Project** and then click the **Next** button.

5. A new **Java Modeling Project** dialog window is now shown. At the top is the **Project name** text box. Type in the name for your new project (for example **Practical2**). Below this is the **Contents** pane; ensure that the **Create new project in workspace** radio button is checked. A new folder will be created in your workspace (in the **P5Practicals** folder) – you can see the location and name in the greyed out Directory box. Then click on **Next**.

6. You are now asked for your **Modeling Settings**. Under **Start with Diagram:**, clear the **Use Default** box and select **Class** from the list of possible diagrams. Name the diagram **Library**. Now click on **Finish**.

7. If you open your **P5Practicals** workspace folder on the desktop you can see that it now contains a **.metadata** folder (holds general IDE settings, including a well hidden catalogue of known projects), and the new project folder **Practical2** (holds various settings and subfolders for the actual design details).

8. The window is now divided into three panes:
   - On the left is the **Model Navigator pane** (where you can navigate through your project).
   - On the top right is the **Designer pane** (where you draw and display the diagrams).
   - Bottom right is the **Properties pane** (where you can edit the properties of your diagram).

Class diagram – the first class: `Publication`

9. Before starting to design classes, it is useful to adjust one of Together Architect's default settings:

   Select the **Window** menu → **Preferences** → **Modeling** → **Java**. Then *clear* the "**Recognize Java Bean Properties**" option. If you do not clear this then Together will recognise when an attribute has standard **get** and **set** operations and will hide the operations and relabel the attribute as a "property", which can be very confusing!

10. The **Designer** pane is set for you to create a class diagram. Now add a *class*. To the left of the **Designer** pane is the **Diagram Toolbar** which shows as icons the different kinds of item that we can add to a class diagram. Placing the cursor over an icon causes the name of the diagram element to be displayed. Find and click on the **Class** item in the toolbar. Now click on a suitable spot in the **Designer** pane – a class named `Class1` is created. The new class name is initially selected for editing, and you could change it immediately, but just press Enter to accept the offered name.

11. You can change the name of the class by editing the diagram. Select the name `Class1`; it will be highlighted. Now type in the name `Publication` and press Enter. The name of the class has been changed.

12. You now need to add attributes and operations to the class. Right click on the class `Publication` in the **Designer** pane. A cascading menu appears; choose **New** followed by **Attribute**. The new attribute is shown highlighted in the attribute part of the class diagram with the name and type `-field1:int` . Press Enter to accept the offered name.

13. You can change the amount of information shown on attributes and operations. Click on **Diagram** in the top menu bar and select **Preferences** from the drop down menu. In the window that opens, choose **View Management**. Click the tick box to **Enable diagram specific settings**. Now set the **Detail level** to be **Analysis** and click on **Apply**. (There are three levels: analysis, design and implementation and each shows differing amounts of detail.) The analysis level does not show the type of the attribute. Now set the detail level back to **Design**, and click OK.

14. Select `field1`, change just its name to `title` and press Enter. You could also just change its type to `String` in the same way – **but do not!** Instead, click on `title` in the `Publication` node and go to the **Properties** pane where you are given all the information about this attribute. Choose **type** in the **Properties** pane, and change `int` to `String` by typing it or by using the drop down menu (small arrow at the right). Press Enter.

15. To add further attributes, right click on the class and select the menu option **New** followed by **Attribute**. Try this by adding the attribute `catNum` of type `int` to your diagram.

16. We now want to add the constructor `Publication`. Right click on the class `Publication` and then select the menu option **New** followed by **Constructor**. The constructor **Publication** is added.

17. To add an operation, right click on the class and select the menu option **New** followed by **Operation**. An operation **method1** is created which we want to change to **`getTitle.`** By default this operation will be **`public`**, will have no parameters and will be **`void`**. Use the Properties window to change its **return type** to **`String`**.

18. To add further operations, right click on the class and select the menu option **New** followed by **Operation**. Add the operations: **`getCatNum`** that returns an **`int`**, and **`borrowPub`** and **`returnPub`** that are **`void`** methods.

The Java source code

19. The tool automatically generates outline Java code. *Double click* on the **`Publication`** class in the diagram pane and a pane appears in the middle right containing the outline Java code. You can close the pane to save screen space. Later, you can also try *editing the code* and seeing the changes reflected in the **Diagram** and **Properties** panes – this is Together's "round trip engineering".

Adding more to `Publication`

20. We want **`borrowPub`** to have a parameter **`member`** of type **`Member`**. Select the operation **`borrowPub`** in the diagram. You can now edit it in the **Properties** window by choosing **parameters** and clicking on the three dots button on the right. A window appears in which you click on the **Add** button and change **`parameter0`** to **`member`** and **`int`** to **`Member`** and click on **OK**. The parameter will appear in your code window, but not in your diagram – why? Note that the class **`Member`** has a red wavy underline in the code window since it has not been defined yet.

21. Currently, the parameter does not appear in the **Diagram Pane**. To see them, we must change our view to **Implementation**. Select the class diagram pane, click on **Diagram** in the top menu and select **Preferences** from the drop down menu. In the window that opens, choose **View Management**. Now set the **Detail level** to be **Implementation**. Click on **OK** and the parameter will now appear.

22. We want the constructor to have two parameters. Select the **`Publication`** constructor and edit its **parameters** value in the **Properties** pane to be: **`String t, int c.`**

Adding a `Catalog` class – a collection of `Publications`

23. Now create a **`Catalog`** class in the same way. It has a constructor together with the operations **`getPub`** (which has a parameter **`catNum`**, and returns a **`Publication`**), **`borrowPub`** (which is **`void`** and has parameters: **`member`** and **`catNum),`** and **`returnPub`** (which is **`void`** and has parameter: **`catNum)`**. Note that you will have to type in **`Publication`** for the return type of the **`getPub`** operation. It will not be available as a default type in the 'return value' drop down box.

24. Create an association between **`Catalog`** and **`Publication`** by selecting the *association icon* from the **Diagram Toolbar** (a diagonal line '/'), selecting **`Catalog`** by holding down the left mouse button and then dragging the mouse to **`Publication`** where the mouse button is released. (If you make a mistake, you can **delete** the association by selecting it and using **Edit** in the Main menu, or using the Delete key.) Now add the **multiplicity** of the association by selecting the association (the line between the two classes) and then select **Supplier** in the list to the left of the **Properties** pane and then set **supplier multiplicity** to be: **`0..*`** . Now select **Client** in the list on the left; you will see that **client multiplicity** is already set at 1 which is what we want. As that is the default, it is not shown in the diagram.

Adding a `Book` class – a *subclass* of `Publication`

25. Now create a **`Book`** class with attribute **`author`** of type **`String`**, a constructor and operations **`getAuthor`** (returns **`String`**) and **`borrowPub`** (parameter **`m`** of type **`Member`**).

26. Click the **Generalization** icon from the toolbar (diagonal line with triangle). Put the mouse over **`Book`**, hold down the left button and drag the mouse to **`Publication`** where the button is released. Note that the **Properties** pane has been updated to show the inheritance.

27. Now set both class **`Publication`** and its **`borrowPub`** operation to be **abstract** by changing their **abstract** value in the **Properties** pane. Note that they are now shown in italics. Change the **`title`** attribute to be **`protected`** by changing its **visibility** value in the **Properties** pane.

Adding a User Interface

28. Now that you have created a collection of *entity* classes, add a *user interface (boundary)* class. It should have operations corresponding to the user actions that you might expect, and will have associations to appropriate entity classes.

Project documentation and Help

29. You may print out your diagram by choosing **Project** from the **top** menu and choose **Documentation** followed by **Generate HTML** from the cascading menus. You can use the browser to determine where the documentation is to be put. Create a folder called **doc** (anywhere, for example in your project folder), and put your documentation there as it is best to put it in a new folder and not mix it up with your UML diagrams. A web page opens with your documentation.

30. Go to folder **doc\Practical2\doc-images** and you will see that gif images have been produced which you can later put into word-processed documents when producing assignment documentation.

31. When you are finished, it is best to *close* the project you are working on: right-click the project name in the **Model navigator** pane, and select **Close project**. You may exit Together by choosing **File** menu, **Exit**. When you next enter **Together**, you can re-open your project by right-clicking on **Practical2** in the **Model Navigator** pane and choosing **Open project**.

32. You can get lots of information on how to use Together by clicking on **Help** in the main menu bar.

That's all.

SBJ Sep 2018