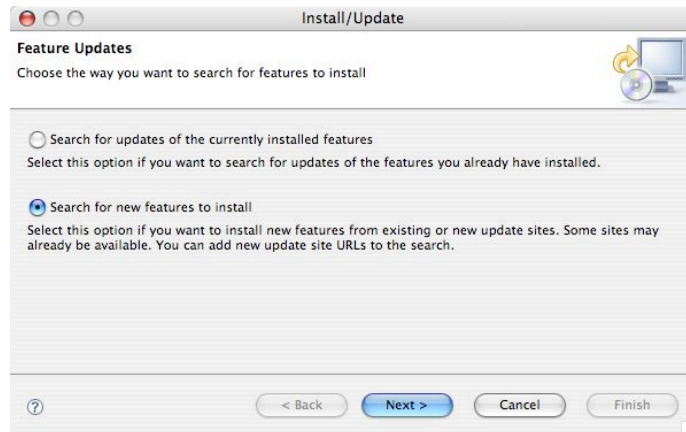


Lab Session 1:

AspectJ Development Tools

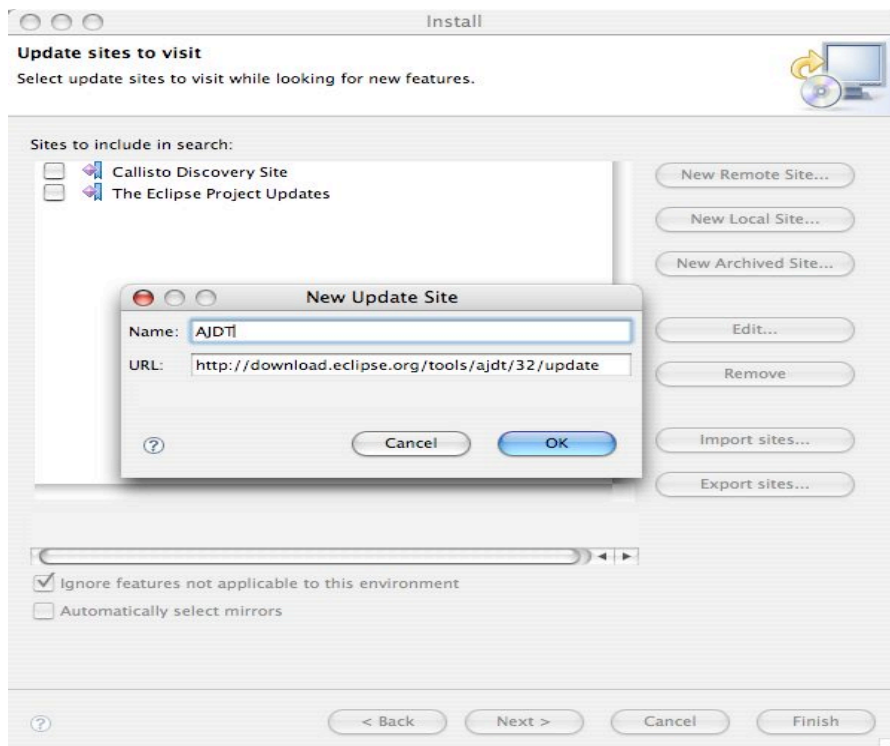
1. Installing AJDT

We will first install the most recent AJDT tools in our Eclipse environment. In the Eclipse main menu, open the Update Manager by navigating through **Help > Software Updates > Find and Install**. The following screen will appear:

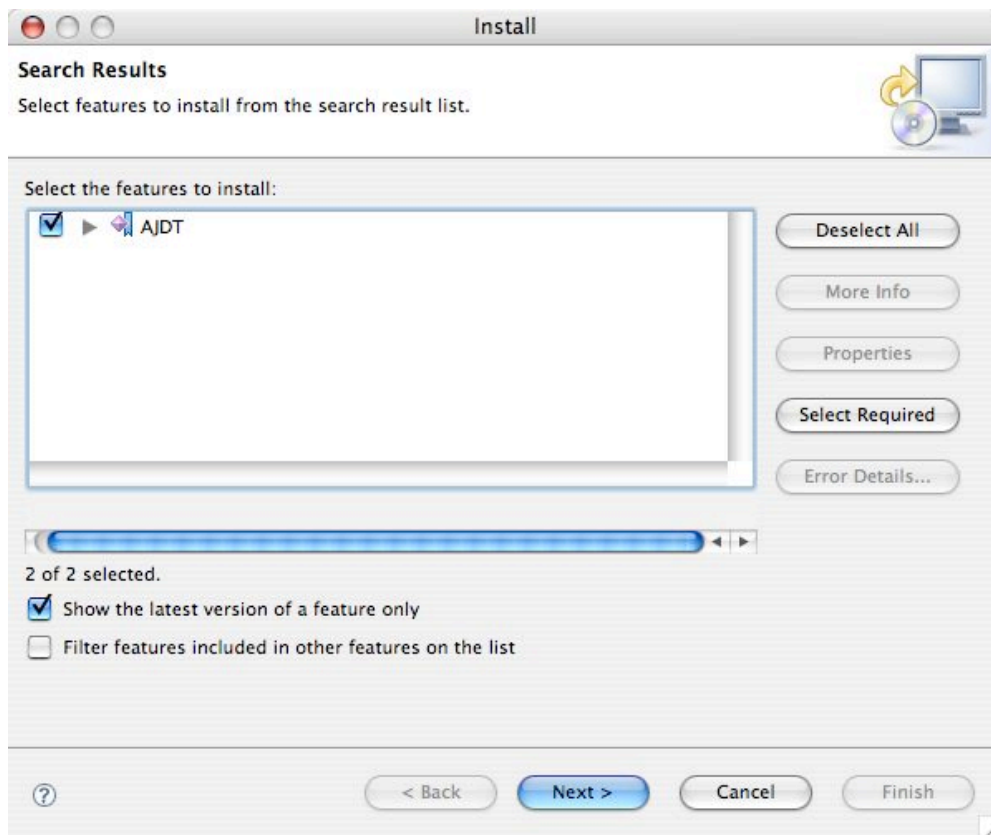


Select '**Search for new features to install**' and press the **Next** button. In the subsequent window, press the '**New Remote Site...**' button and fill in the information as it appears in the following screenshot. Depending on the Eclipse version, you need to enter a different URL:

- Eclipse 3.0.x: <http://download.eclipse.org/tools/ajdt/30/update>
- Eclipse 3.1.x: <http://download.eclipse.org/tools/ajdt/31/update>
- Eclipse 3.2.x: <http://download.eclipse.org/tools/ajdt/32/update>



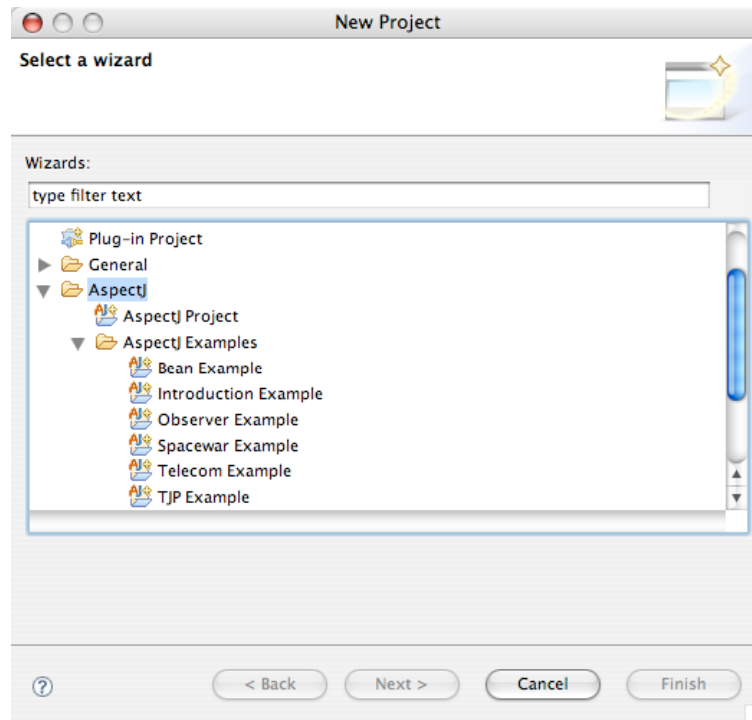
After pressing the 'OK' button, the AJDT site will be automatically selected. Press 'Finish' to continue. In the following panel (see below), you need to select the freshly created AJDT feature to install it:



Press the 'Next' button (and accept the license agreement). After that, you need to select the installation directory for the feature. Select a directory where you have write access. After that, the feature will start downloading and installing.

2. Exploring AJDT

After installing AJDT correctly, we can start exploring its features in the context of a simple AspectJ project. For this purpose, import the '*Telecom*' example which was packed with the AspectJ AJDT installation. In the '**New Project Wizard**', select the *Telecom* example in the *AspectJ* folder:



After completing the steps above, a new project named '*Telecom example*' should have been created in your projects view. We will investigate the structure of this AspectJ application through the use of the AJDT.

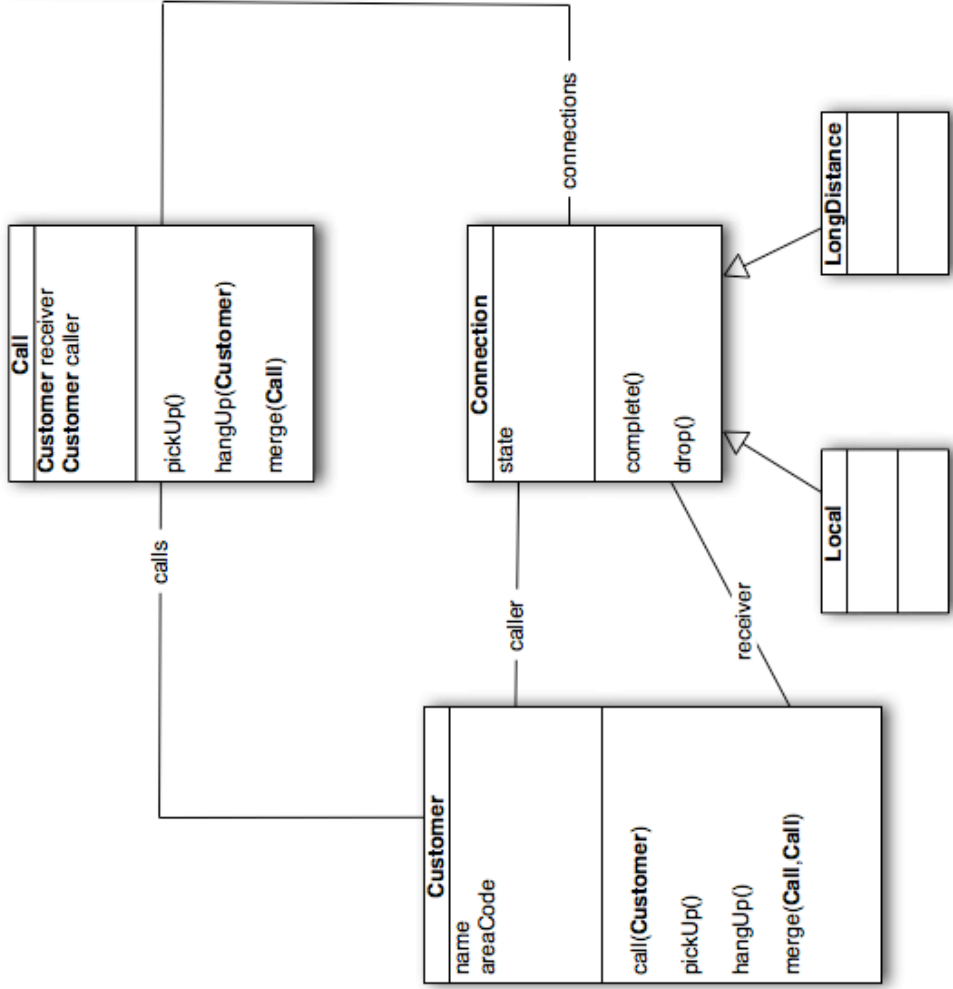
Exercise 2.1: Complete the diagram on the following page to indicate how the aspects intervene in the application. Use the **Outline**, **Editor**, **Aspect Visualisation** and **Cross-References** views to indicate which aspects *advise* which methods (or calls to methods) in the *Telecom* application. Indicate also which aspects *declare* which members in which classes. We do not use a particular notation in this exercise, the intention is to explore the AJDT and use it to comprehend the *Timing*, *TimerLog* and *Billing* aspects in the *Telecom* application.

Timing Aspect

TimerLog Aspect

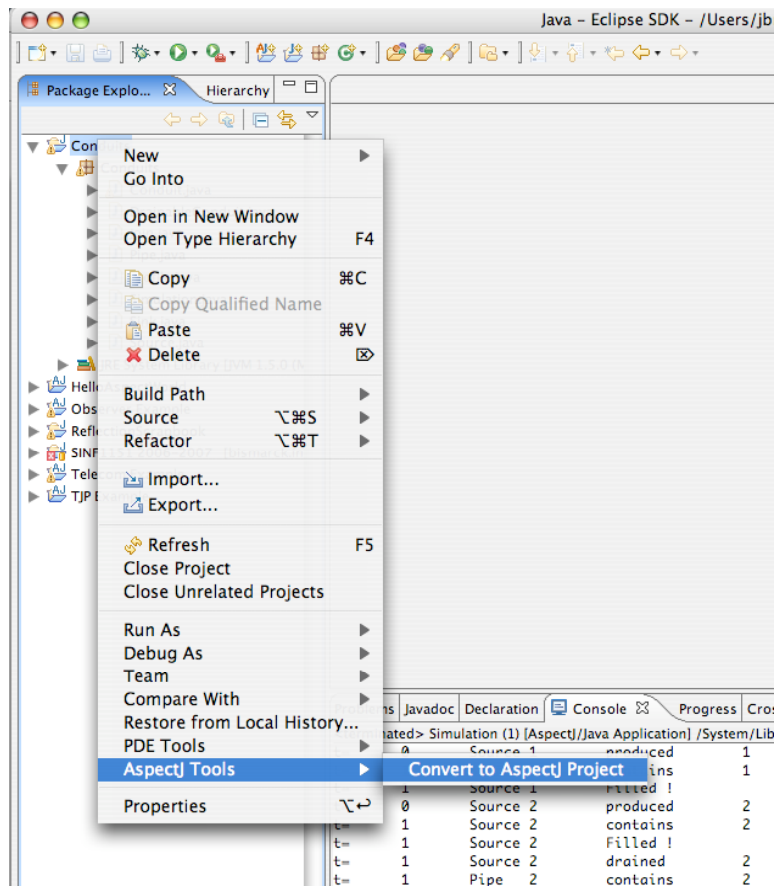
Billing Aspect

Timer
start()
stop()

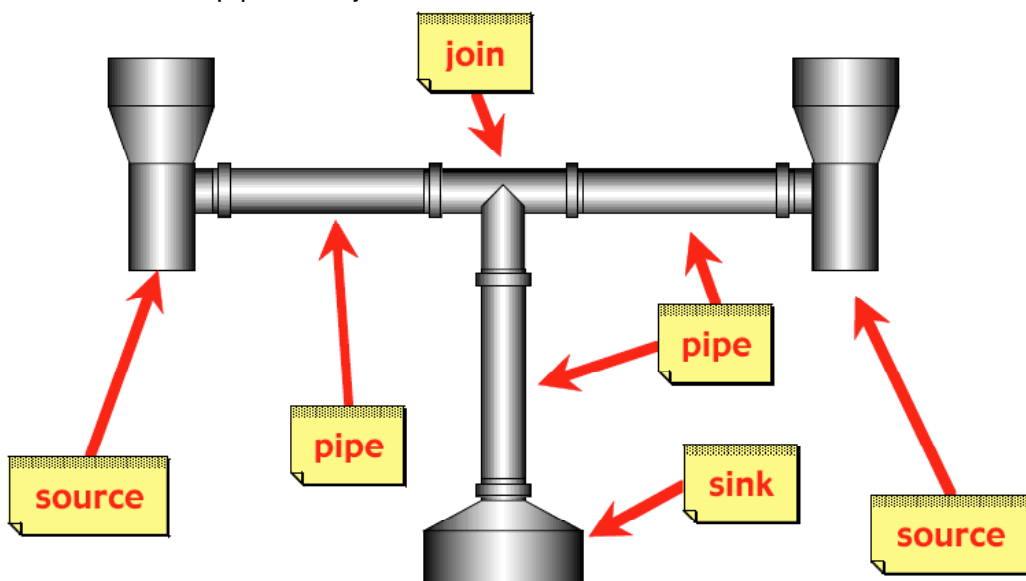


3. The Conduits AspectJ-Project

Load the *Conduits* Java application and convert it into an AspectJ project using the context menu of the Conduits project:



This application is a simulation of interconnected conduits that transport water from sources to sinks via pipes and joins:



Exercise 3.1: Investigate the code of the application, understand its workings and find some crosscutting concerns. What crosscutting concerns can you find and why do you think they are crosscutting concerns? Use the following questions to identify potential crosscutting concerns:

- What application functionality is distributed across multiple classes and methods?
- What application functionality cannot be provided to other applications as a reusable module?
- What application functionality requires the developer to modify multiple files/classes/methods in order to change its actual behavior?

Exercise 3.2: Describe how to refactor the code of the application such that you can write an AspectJ aspect to modularise the crosscutting concerns identified in exercise 3.1. Design the aspects, their pointcuts and their advices. If you have some remaining time, start implementing the aspects. We will finish their implementation in the next lab session.