
1. Introduction	1
1.1 Context	1
1.2 Assessment	1
2. What's to be done in first few lab sessions?.....	1
2.A Reverse Engineering task-Chapter 15.....	2
2.B Reverse Engineering task-Chapter 16.....	3
2.C Generate code from a state diagram.....	3
2.D Enter various diagrams etc.....	3

1. Introduction

1.1 Context

The main purpose of the first few lab sessions is to form an overview of the Unified Modeling Language (UML) and to start familiarizing with the SW tool WithClass¹ (from Microgold). In later lab sessions, it is intended that students should work on their major continuous assessment project and that tutors will be available as support.

1.2 Assessment

Overall marks for the module are split between 70% for final exam and 30% for continuous assessment.

The continuous assessment 30% will be for a fairly substantial project, going from specification of a problem through to system testing and [optionally] demonstration.

2. What's to be done in first few lab sessions?

Normally, UML is applied in a “forward” fashion. Thus, proceeding from requirements, one forms use case and class models, and then proceeds to interaction diagrams etc, and finally to code. However, as an introductory exercise, it seems useful to perform a **reverse engineering** process whereby one starts with Java code and applies WithClass to generate a corresponding class diagram. The Java code will be taken from the web-site of our text book (Stevens & Pooley).

¹ As indicated in lectures it is acceptable for students to use an alternative tool (such as Rational Rose) for their main continuous assessment project.

However, tutor tool support will be primarily focused on WithClass; note that besides the help facility there is a user manual on-line.

It is expected that the main continuous assessment project will be specified by the end of the third week of the semester.

2.A Reverse Engineering task-Chapter 15

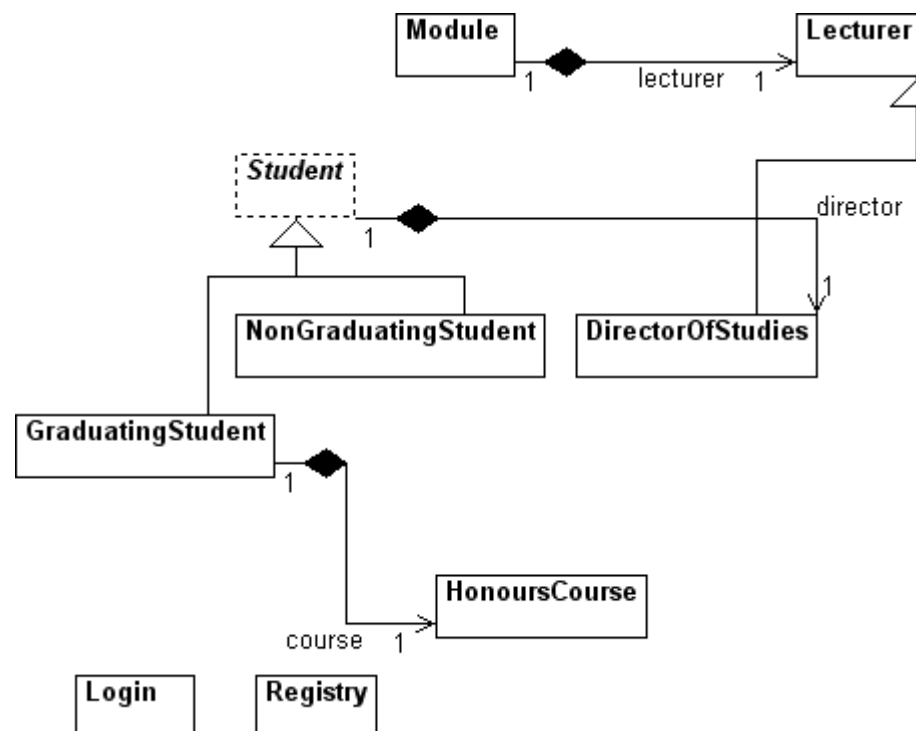
(1) Download code corresponding to Chapter 15 of Stevens & Pooley from [Using UML: CS4 example](#). For example, you could save the 9 files listed (HonoursCourse to Registry) as Java files on a floppy disk.

(2) Start a new project in WithClass and use the “Utilities/Reverse Directory” (browsing) to import each of the 9 files from step (1) into the project. You will find that an initial class diagram appears which may be rather confusing at first. This is because (a) a lot of detail may be shown and (b) the entities must be “pulled and dragged” to form a diagram that is intelligible to the human eye.

(3) You can control the level of detail in the class diagram through the icons/buttons on the left side of the class diagram. For example, the top icon will suppress all but the class names.

(4) Next, somewhat guided by Fig 15.2 of book (see my web pages), see if you can drag the diagram into a more intelligible layout. It won't be exactly like Figure 15.2, as the authors have not followed their design faithfully into the code (as they explain).

For example, I obtained



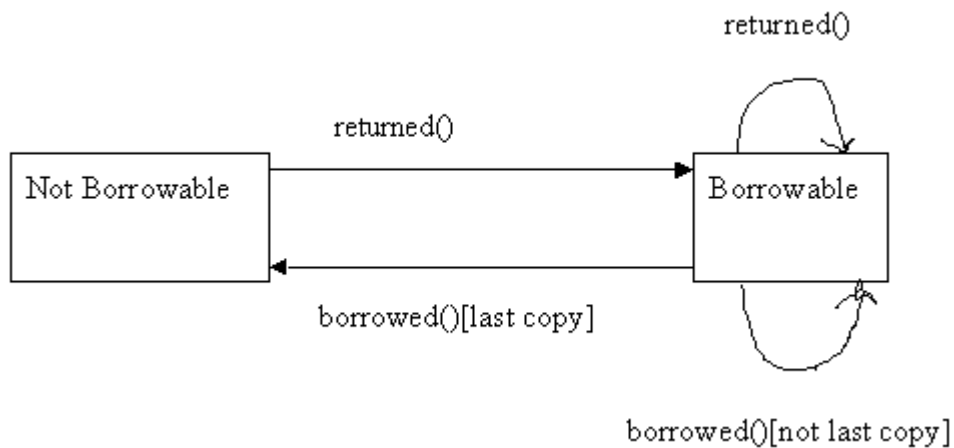
(5) Among other things, it is of interest to generate a “code report” - “Utilities/Generate Class Code/Report”. In general, have a look at the various facilities provided by WithClass.

2.B Reverse Engineering task-Chapter 16

Same as 2.A for code corresponding to Chapter 16 of book.

2.C Generate code from a state diagram

Try entering a state diagram such as



Then, generate the corresponding code (“Utilities/Generate State Code/Report”).

2.D Enter various diagrams etc

Enter diagrams from the library example (of the lecture notes) or say from the Bank Account example in Chapter 11 of the WithClass user manual or other. Then, try various manipulations such as seeing what kind of code is generated from the class diagrams or see how to generate code from sequence diagrams etc.