

Software Engineering with Objects and Components 1

Fourth Tutorial: Practical Work

The aims of this tutorial are:

1. To discuss any feedback for your Deliverable 1
2. To prioritise use cases
3. To identify conflicts (in terms of classes) among your class diagrams
4. To solve conflicting class diagrams
5. To identify responsibilities for the implementation of shared classes.

Team Resources

- 1 instruction sheet (this page)
- 1 blank OHP slide
- 1 non-permanent OHP pen
- your first Deliverables

Instructions

Phase I. Your tutor will give some feedback on your first deliverables. The aim is to discuss the main issues with your deliverables. This feedback will let you revise your deliverable 1 in order to address your deliverable 2. This phase should take up to 15 minutes.

Phase II. Each team will have 10 minutes to inspect the first deliverable of the other teams. In particular you should identify the shared classes between teams. By the end of the tutorial you have to agree on a common definition for the shared classes and to allocate the responsibilities for their implementation. For instance, if a **Student** class appears in the deliverables of different teams, you have to agree on the definition (in terms of, attributes, methods, visibilities, etc.) of the **Student** class. Moreover, you have to allocate the responsibility for its implementation to one of the teams. To be notice it is that all teams will share classes with one another.

Priorities, Inconsistencies and Responsibilities. The activity is structured in the following way (note that this is very tightly timed – your tutor will enforce these so the activity fits in the available time):

Preliminaries: Get into your teams.

5 mins *Each team* inspects/reviews one of the deliverables of another team: identify the shared classes and write down its specification.

5 mins *Each team* inspects/reviews the deliverable of the other team: identify the shared classes and write down its specification.

10 mins *Each team* discusses/identifies:

1. the use cases' priorities. To be notice it is that you will be required to implement the most important use cases (high priority). Among these high priority use cases are those involving shared classes. These use cases require you to implement and use shared classes.

2. the identified inconsistencies in the deliverables of the other teams
3. proposals for solving the class inconsistencies
4. proposals for allocating the implementation responsibilities.

Each team constructs one OHP slide with the above information, i.e., 1, 2, 3 and 4.

5 mins *Each team in turn* is given 5 mins to present their information on use cases' priorities, classes' inconsistencies and responsibilities.

Finally: At the end of the tutorial you have to agree on the design of the shared classes and the responsibilities for implementing them.

Tutorial Outcomes

By the end of this tutorial your group should

1. Have some feedback on your first deliverable
2. Have identified priorities for your use cases
3. Have identified changes on your class diagrams in order to accommodate other team viewpoints
4. Have identified the responsibilities for the implementation of the shared classes.

MASSIMO FELICI
OCTOBER 29, 2004