**Software Engineering with Objects and Components 1**
**Seventh Tutorial: Practical Work**

**Please read this tutorial sheet before you arrive at the tutorial. You are required to do some preparation for the tutorial.**

*Before the tutorial* you should *prepare* a short presentation on OHP slides that:

1. Presents the Java definition of the main two classes that you intend to develop Java code for. ***To be notice it is that in order to realize sensible and non-trivial use cases you may need to implement other classes that collaborate with the two chosen ones.***
2. Presents the main use cases that involve those classes.
3. Presents an outline of the collaboration diagrams corresponding to those use cases.
4. Provides a summary of the test cases you propose to develop to test the classes.
5. Demonstrates how the test cases can be traced back to the relevant use cases via the collaboration diagrams. ***You may want to use (for your deliverable 2) a template similar to that one presented in the Lecture Note 14 on Software Testing.***

The aim of this tutorial is to identify implementation issues (e.g., conflicting definitions, missing information, etc.) that inhibit the collaboration between teams in order to realize high-level and complex use cases. Ideally:

1. This should be a small subset that would allow the rapid implementation of a basic system.
2. There should be agreement across the teams on the collection of selected use cases that will provide the basis for deliverable 2.
3. The outcome of the tutorial will be:
   (a) An agreed set of use cases from each team that will be used as the basis for this part of the practical.
   (b) Agreement with each team on the classes that they should provide tests and implementations for.

*After the tutorial:*

1. You should finalize the definition of the main classes that will provide the basis for deliverable 2.
2. You should begin to define the tests you need for your chosen classes.
3. You should review the choice of use cases chosen to represent the preliminary implementation and attempt to identify inconsistencies.

## Team Resources

- 1 instruction sheet (this page)
- 1 blank OHP slides
- 1 non-permanent OHP pen
- 10-15 minutes preparation time

## Instructions

***Each team brings along the slides with the two Java class definitions they are implementing for their deliverable 2.*** Each team will have 10-15 mins to present the presentation. The rest of the group will comment on the suitability of the proposed implementation of the team presenting. The aim of the comments should be to fix the inconsistencies in the class definitions and improve the quality of the testing of the team presenting their proposals.

The process you should follow is:

1. Each team presents their slide overviewing the Java class definitions, together with some preliminary test cases, they intend to fully implement. During the presentations a "secretary" to the group records potential issues to be resolved between the teams.

2. After the presentations, the "secretary" takes the group through the list of issues attempting to resolve them, this process will arrive at an agreed collection of classes that will be implemented for deliverable 2.

3. Teams consider the use cases their classes realize. Teams figure out high-level use cases combining the proposed use cases proposed. Ideally the classes should be non-trivial and should both be involved in meeting at least one of the requirements expressed in the use cases.

4. If there is time available the teams should revise potential test cases for their classes.

## Tutorial Outcomes

By the end of this tutorial your group should have:

1. Agreed a set of use cases that capture the preliminary implementation of the system

2. Ensured that each team has agreed on the Java class definition they are implementing for deliverable 2 with their tutor

3. Begun to refine what test cases are appropriate for their classes.

Massimo Felici
November 14, 2005