Software Engineering with Objects and Components 1
Group Tutorial Project: Handout 1
October 2003

This project should be undertaken in *teams of 3–5 people.* The aim of the project is to give a small-scale case study within which to gain experience Software Engineering using UML (and Java) in collaborative software development.

The remaining sections in this handout outline the project's organisation, overview and deliverables. Subsequent handouts will provide details on deliverables and assessment. You should read *all* of this handout before commencing any activity on the project.

Organisation

This section describes the organisation of the project. You should read it carefully and ask if you do not understand anything.

Communication

Notices relating to the project will be distributed to the class via the tutorials and archived on the course web page. Messages relating to the project will be posted to the eduni.inf.ug3 news group. *It is your responsibility to keep in touch by attending tutorials and reading the newsgroup and course web-page.*

Deliverables

The work you are required to do is packaged into two deliverables. Each contributes to your final assessment. A deliverable consists of: (i) A piece of work done by the team; together with (ii) An assessment of the effort contributed by each member of the team (this is expected to be 100% each.)

How you decide to partition the work which contributes to a deliverable is a matter for the team. At the end of each deliverable report there is a section which allows the team to assess the level of contribution of each team member to the deliverable. If any team member is dissatisfied with the working of the team they are entitled to submit a "minority report" on a separate copy of the deliverable report form for that deliverable. Assessments of contributions and "minority reports" will be used in determining the individual mark allocated to a team member. The phasing of the deliverables is:

Deliverable 1: Final deadline 5 pm Monday 10th November.

Deliverable 2: Final deadline 5 pm Monday 8th December.

Submit your work to the ITO, JCMB, Room 1502.

Part of the work of this project is deciding on the phasing of work such that you can meet the above deadlines. The precise content of the deliverables should be decided in consultation with your tutor. You should keep in mind that you have only limited time to devote to the project.

This document, together with the supporting details on deliverables, provides enough information that you can decide how to go about tackling the problem. It may well be that you should plan to submit work some time before the deadlines to allow for slippage and for other work to dominate close to the end of term.

Overview

The overall aim of the project is to give you experience of collaborative activities of the kind typically undertaken in software development and the rôle of modelling languages such as UML in those activities.

The School of Informatics is in the process of developing a Secure Coursework Submission System (SCSS). This has been underway for some time and a partial requirements analysis and design has been completed (see the links on the practical page). Your tutorial group is one of four (competing) software companies who are bidding to win this lucrative development contract. As a tutorial group, you will be asked to develop the specifications for SCSS. Bearing in mind the limited resources you have to devote to this project, your group will split into three teams, each of whom will produce the specification for a part of the overall system. This will give you some awareness of the issues involved in the development of systems by structured teams, and of the use of UML in expressing and communicating models of systems.

Deliverable 1 concentrates on constructing a suitable requirements document (incorporating Use Case Models) for SCSS, providing the specification (incorporating Class Models) for an initial development iteration of the system, and providing supporting evidence validating that specification (e.g. using CRC cards). This contributes 50% of the final coursework mark.

Deliverable 2 concentrates on the design and test of a small "proof of concept" prototype of SCSS. This contributes 50% of the final coursework mark.

Tutorial Work

This section describes the provisional timetable for tutorial activities. The deliverable deadlines are firm, although the other tutorial tasks may be subject to change in exceptional circumstances.

First Tutorial: forming teams

At the tutorial meeting on Wednesday 15th October you will be split into three teams: named S, L, A. Your tutor will tell you who is in each team. For the remainder of the term, the teams will each concentrate on different facets of the system development. These are: the *S*tudent-centred aspects of the system; the *L*ecturer-centred aspects of the system; and the *A*dministration-centred aspects of the system.

You should use the first tutorial to plan your team's development process for the remainder of the term, and to organise how you will work together as a team. You will also be asked to propose a name for your software company (i.e. your tutorial group).

Team Presentations

In weeks 4,5,7 and 8 the three teams will each give presentations to the rest of the group. In these weeks teams will each have 20 minutes for a presentation, followed by a question and answer session. The timetable for tutorial tasks is given below. In the second of the two weeks there will be time to reflect on all the presentations.

The presentations have two purposes: (i) for the presenting team to have their specifications reviewed by other group members; (ii) for the other teams to learn how the presented specification satisfies (or fails to satisfy) the requirements of their own sub-system.

The presenting team should spend 10-15 minutes presenting, with the rest of their alloted time available for questions. The other teams should prepare for tutorials by drawing up a list of questions which they need answered for their own specifications.

In the second week the group should also consider interactions between their presentations and work out how to resolve any conflicts or inconsistencies.

Timetable

Week	Date	Tutorial Task
2	15th Oct	Team formation and class model exercises
3	22nd Oct	Use case & class model presentations: *S* and *L* teams
4	29th Oct	Use case & class model presentations: *A* team plus review of the presentations
5	5th Nov	Requirements review: all teams
6	10th Nov	**Deliverable 1 deadline: Monday, 5pm**
6	12th Nov	Consolidation of requirements and preliminary design:all teams
7	19th Nov	Activity diagram presentations: *S* and *A* teams. Java development: all teams
8	26th Nov	Activity diagram presentations: *P* and *M* teams. Java development: all teams
9	3rd Dec	Java prototype review: all teams
10	8th Dec	**Deliverable 2 deadline: Monday, 5pm**

Problems and Grievances

There are many potential risks in this project. For example: you may find UML difficult; preparing for tutorials may take more time than you can afford; you may have difficulties working with certain teams in your group, or even with certain members of your own team.

If you are finding difficulties with the work, speak firstly to the other members of your team to see if you can solve the problem. If this does not work, speak next to other teams in your group. Finally, if this does not solve the problem then your team/group should present their problem to the tutor.

Your tutors' job is to ensure that the tutorials run (relatively) smoothly and that they cover the necessary work. The preparation for the tutorials is your responsibility. You should expect to spend several hours a week on preparation for tutorials (including deliverables). If you find that you are spending considerably longer than this, then you are probably going into too much detail in your specification. If this happens, consider taking a more abstract view of the system, or restricting to a smaller part of the system.

Finally, if you find that you are having irreconcilable difficulties with other members of your team, you should (politely) inform your tutor as soon as possible.

Presentation Skills

Most of you will probably have little or no experience of giving presentations. Fear not, there is plenty of advice available: try searching the library catalogues or the world-wide-web with the keywords "presentation skills". A few specific suggestions are:

Be brief: you only have a limited amount of time, so make your presentation focused and to the point; however you must also...

Be clear: and bear in mind that you necessarily understand what you are talking about far better than the audience – so take them through the material nice and carefully.

Be structured: a good idea is to follow the "tell 3 times" rule. When explaining something to an audience you should:

1. tell them what you intend to explain (*Introduction*)
2. tell it to them (*Content*)
3. tell them what you have just told them (*Summary*)

You should make good use of equipment (such as OHPs and blackboards). Having legible OHP slides prepared beforehand saves time and gives your presentation a structure to work from. OHP slides should be brief and to the point: as a rule of thumb, expect each slide to have 3-6 bullet points of 4-8 words each, and no more.

You may want to prepare handouts for your audience to examine and take away: these might include UML diagrams; more detailed written explanations; and lists of issues/questions you wish to resolve.

While you may have little experience of giving presentations, you do have substantial experience of attending presentations (i.e. lectures). Think about your experiences on SEOC1 and other courses – of lectures, overhead slides and handouts. Think about not only examples of *good practice* (things to emulate), but also of *bad practice* (things to avoid).

Above all, remember that your fellow tutees are all in the same position. So do try and learn from those who give particularly good presentations, and do try to give *positive* criticism to those whose presentations are not so clear.

Stuart Anderson
October 13, 2003