CS3 Software Engineering with Objects and Components 1
Group Tutorial Project: Deliverable 2

NC-Detergents: Initial Implementation Iteration

Your software company (tutorial group) has now submitted its first project deliverable (use case and class models) to *NC-Detergents:*. Your second project deliverable is a "proof of concept" Java implementation and supporting documentation (revised use case model, activity diagrams and test suite).

Deadline for Deliverable: 5pm Mon 10 Dec 2001

Documents to submit

Your team should submit: a completed cover sheet; a revised use case model and accompanying specifications; a prototype Java implementation of a *subset* of these use cases; documentation for this Java code in the form of activity diagrams; a test suite for your prototype consisting of a test specification for each implemented use case. As for the first deliverable, the cover sheet records your tutorial group, team details and contribution given by each team member.

Your tutorial group as a whole should also submit, by email to your tutor, the Java code and test suite. Further details on this part of the submission will be issued shortly.

Revised Use Case Model

You should provide a (potentially) revised version of the use case model from your first deliverable. You should expect this model to contain 12–20 use cases in total. If your original deliverable contained less than 12 use cases, these were probably too abstract and you *must* sub-divide some of your original use cases to produce at least 12. If your original deliverable contained more than 20 use cases, these were probably too detailed and *if you wish to* you may compose some of your original use cases to produce no more than 20. As with the first deliverable, your use case diagrams should be accompanied by brief informal descriptions of the tasks they specify.

You must prioritise your use cases as being either *alpha* or *beta*. Those use cases designated *alpha* should comprise between one third and one half of the total number, and should consist of:

1. for each other team in your tutorial group, at least one use case in which (some part of) that team is an actor. Note: some of these may be very simple, but at least one must involve a minimum of four of classes from your subsystem;

2. at least one use case in which the actor is outwith *NC-Detergents* (e.g. The HSE, external Auditors, Shareholders, ...);

3. at least two use cases in which the actor(s) are internal to your subsystem (i.e. a human actor in your department, or some internal non-human part of your subsystem)

Java Prototype

You must provide Java classes for sufficient of the classes from your first deliverable's class model to realise the *alpha* use cases. You must also ensure that you have provided sufficient Java classes to realise any relevant use cases prototyped by other teams in your tutorial group. These Java classes should have consistent attributes and methods/messages.

In this initial prototype you are *not* required to provide code implementing each classes methods. The implementation of methods should simulate the passing of messages to other prototyped Java classes (whether of your subsystem or of other teams subsystems), and should pass valid arguments with these messages *but* no other computation in response to these messages need be implemented (so arguments may simply carry valid default values). This "proof of concept" code is to demonstrate that the messages passed between these Java classes are *potentially sufficient* to realise the *alpha* use cases.

In the event that a valid Java class implementation requires a design decision based upon information which is not available to you, you are free to make any reasonable decisions concerning this detail of the design, provided that these decisions are recorded and submitted as part of your deliverable.

Test Suite

Further informataion regarding the test suite which be issued shortly.

Documentation: Activity Diagrams

You must submit activity diagrams which document how your Java classes interact with the classes prototyped by the other teams in your group. These activity diagrams should cover all of your *alpha* use cases.

Assessment

Marks will be awarded by team, with individual team members receiving a proportion of marks in relation to their contribution as for deliverable 1. Marks for this deliverable constitute 50% of the practical marks for this course. That is, 12.5% of the overall course mark. The revised use case is worth 20% of this mark, the activity diagrams 30%, and the Java protoype plus test suite is worth 50%.

While your reports are assessed as individual teams, it is expected that all four of the groups' reports will be consistent (i.e. wrt terminology, shared use cases, Java classes, etc.) Marks will be awarded by team on the basis of that teams Java subsystem. However, bonus marks are available to the tutorial group as a whole if the entire system demonstrates successful interfacing between the 4 subsystems.