Project Directions

CS371S –Fall 2006

1. Project Teams

Projects will be executed by teams. A team will be two to four participants with members with three class members as the target. The team will select a coordinator for each team. The coordinator will be the person who communicates with the instructor concerning the project.

At the end of the project each team member will be asked to evaluate the contribution of each of the team members to the project. Individual grades on the project are determined by multiplying the average of the grades for the project as a whole by the rankings of contributions of each individual by the team. Complete directions for the ratings will be provided.

2. Development Environments

There are several possible development environments. This year everyone will use Objectbench for debugging and code generation. Objectbench regrettably does not support standard notation. The two other tools that support all of standard notation and some form of debugging and code generation (Bridgepoint and iUML) have both caused problems with installation. There are several free capture tools available which offer full capture of diagrams in standard form. Some also generate code skeletons. These include Poiseidon, MagicDraw (from Bevoware) and Eclipse. You can use any of these for capture support if you want.

3. Project Documentation

Each step of the development process yields one or more work products. These work products will be the means of evaluation of each step of the project.

3.1 Directions for Submitting Project Materials

A directory for project workproducts for each group will be provided under /projects/cs371ss2006. The project work products should be placed in that directory for evaluation.

3.2 Content for Reports

The first step should be to review the requirements specification you have been given. Create a new requirements specification document if needed. Please feel free to address questions concerning your requirements specification to me by email or to make an appointment to discuss the requirements specification

(a) Class Diagram - Please turn in the derivation of the class diagram and the files for the class diagram by 5PM on 10/3/2006. Derivation of the class diagram should be pursued by both the linguistic analysis method and the Object Behavior Analysis method. You will get feedback on the first versions of the class diagrams and there will be a later final submission.

The first element of the reports should be an updated and clarified requirements specification. (If any assumptions or clarifications are needed.)
The report for the linguistic analysis method should include the following:

(i) A list of nouns from the requirements analysis with an analysis of why they are or why they are not objects.

(ii) A list of the adjectives associated with each noun selected as an object and why each is or is not an attribute for that object.

(iii) A list of verb phrases and an analysis of why each defines a relationship or why it does not lead to a relationship.

Use the tabular format as in the examples.

The report for the Object Behavior Analysis method should include:

i) One or more scripts based on the requirements analysis.

ii) The Parties, Services and Attributes Glossaries derived from the scripts

iii) Object cards for the derived objects omitting the "Provided Services"

iv) A few paragraphs reasoning about instances of abstraction and/or specialization.

The tables and object cards can be submitted as a pdf, ps or text file but the class diagram should be the diagram from the tool.

(b) Derivation of the State Models without actions.

Construct the state models including the event lists using both the OBA and Use Case approaches.

Submit the state diagrams in the format of the tool you are using in the context of the current version of the class diagram. State transition tables can be from the tools when these are supported or as Word or Word equivalent tables converted to pdf.

Please submit the first draft of the state model workproducts by 10/17/2006.

(c) Class Diagram - Complete Class Diagram including state models and action language– Add the action language programs to the state models. Please submit the revised class diagram (including state modelswith action language) by 10/31/2006 at 5PM

d) Test Plan for the Model – Please submit a complete test plan by 11/7/2006. The test plan can be specified as an English narrative listing the scenarios you plan to execute. There will be additional specifications for the work products of this step given in the lectures.

e) Validated Models – Please submit the current version of your validated model by 11/21/2006 at 5PM including the test scenarios.

f) The completed and validated C++ system for your project including the code, the input and output files for the test scenarios and directions for using the system should be available for inspection by the instructor and the TA by 11/30/2006.

4. Class Presentations - Each team will make two presentations. Each member of the team must participate in at least one presentation. The schedule for the presentations is given in the lecture schedule and repeated following. The transparencies for the presentation must be available on the Group's project directory for review by the instructor at least 24 hours prior to the time of the presentation.

a) Class Diagram Presentations – The presentations will be on 10/10/2006 and 10/12/2006. The presentations will be in order of group number. The coverage of the class diagram should include:

i) The analysis of the requirements specification leading to the object class diagram.
ii) The graphical representation of the class diagram
iii) The work products for the class diagram

b) State Model Presentation The state model presentations will be 11/14/06, 11/16/06. The presentations will be in reverse order of group number. Hopefully all of these presentations will be at least partially in the form of demonstrations of the tool executing the model.

i) The analysis of attributes and objects which leads to the definition of the states for each object.
ii) The state transition diagram for each object including the events and action language programs
iii) The action language for each state model. Hopefully all of the state models will be running and can be demonstrated using the simulators

5. Demonstrations

a) I will want a demonstration of each model when it is validated. The dates for the demonstrations will be set later in the semester. Demonstrations consist of the instructor, the TA and the team jointly executing the model. The instructor will typically create some weird set of inputs to test the system.

b) Validated Code - The validated code should be available for demonstration to the instructor and the TA. The instructor will typically create some weird set of inputs to test the system.