University of Toronto
Department of Computer Science

CSC302S – Engineering Large Software Systems

March 1, 2008
Prof. Steve Easterbrook

Assignment 3: Requirements Analysis

Due Date: 11:20am, Thursday, March 20, 2008
(i.e. within 10 minutes of the start of the tutorial)

This assignment counts for 15% of the final grade

Analyze the requirements for a new feature request for the UMLet application. You will need to select one of the new feature requests listed below. Your analysis will include a goal model to explore the user's requirements, a domain model to map out the domain entities, and a set of use cases describing how the user's will use the requested feature. Finally, you will complete a robustness analysis to explore how these use cases map onto the architecture of UML. You are not required to implement the new feature for this assignment (but will be doing so for the next assignment).

The project is to be carried out in your assigned teams. Each team will submit one report.

I. Doing the Assignment

This assignment has 7 steps. They are:

1. *Read the list of new feature requests provided on the next page*. Check that you understand what each feature is. Ask your TA or the instructor if you need any user clarifications.
2. *Select one of the listed features for the assignment*. Use techniques you have learned in the course for estimating effort, assessing risk, and understanding customer priority to help you make the selection. You will be expected to defend you decision in the project write up. Note: you are strongly encouraged to investigate whether existing libraries and/or (open-source) implementations of any of the requested functionality can be re-used for your implementation.
3. *Complete a requirements analysis for the selected feature*. Your requirements analysis will identify why the user(s) want the requested feature, and will identify relevant assumptions about the problem domain. At the end of the analysis, you should have a detailed list of specific functions and quality requirements, along with a rationale for how the functions will allow the users to meet their goals. Make use of techniques such as domain modeling and goal modeling to help you complete this analysis.
4. *Generate a detailed set of use cases for the selected feature*. Create a use case diagram to give an overview of the set of use cases, and create individual descriptions for each use case, including pre- and post-conditions, exceptional behaviour, and alternative paths for each use case. Your use case descriptions should be written in a style that the users would understand, so that, for example, they could form the basis for a user manual.
5. *Complete a robustness analysis for your use cases*. Your robustness analysis should take into account the existing classes in the UMLet application, plus any new classes you need to create to implement the new feature. Use the robustness analysis to decide whether each requirement should map onto a new or an existing class.

6. *Write a report* that describes the steps you went through to select and analyze the new feature request for UMLet. Be sure to document your development process, and comment on how well the process worked for you.

7. *Document your teamwork* and complete team member evaluations and team member evaluation summary form, using the online system on the course website. Submit signed hardcopies of the online forms.

II. List of change requests

You should select one of the following list of new feature requests:

1. Add an export feature, which provides the ability to export a UML model to XMI, the OMG defined, XML-based, serialization format for UML models.

2. Add an auto-layout feature, which provides the ability to automatically lay out a UML diagram (or a subset of the elements in a diagram) in an aesthetically pleasing way, using a standard graph layout algorithm.

3. Add a code generation feature, which produces templates for Java code from UML models.

4. Add a constraint checking feature, which provides the ability to specify constraints on UML models, expressed in the Object Constraint Language (OCL), and to check that the constraints are satisfied by the models.

III. What to Hand In

Hand in your report at the start of your tutorial on the due date. *Reports not handed in within the first ten minutes of the tutorial will be treated as late.*

The report should not exceed twenty (20) pages (not counting cover pages, appendices, and teamwork forms). All detailed diagrams should be included as appendices, but they should be summarized in the main body of the report, with enough commentary for the reader to understand what you did, and what the appendices show.

Your submitted report should include the following items:

1. A brief description of the rationale for your choice of feature to implement.

2. A requirements analysis for the selected feature. Your analysis should clearly separate user requirements from specifiable functions, and indicate any assumptions needed to ensure that the functions to be implemented will indeed satisfy the requirements. Include any models (e.g. domain models, goal models) that you created to help you in the analysis.

3. A complete set of use cases for the selected feature. Your use cases should show alternative paths and exceptional behaviours, and document pre- and post-conditions for each use case.

4. A robustness analysis for the selected feature. Your analysis should clearly indicate which classes already exist in the current version of UMLet, and which will have to be added. It should also clearly show how use cases map onto these classes.

Written Presentation Requirements

Be sure to include a cover page indicating the name of your team, the names of all team members, title of work, course, date and tutor's name. Assignments will be judged on the basis of visual appearance, the grammatical correctness and quality of writing, and the visual appearance and readability of the models, as well as their contents. Please make sure that the text of your report is well-structured, using paragraphs, full sentences, and other features of a well-written presentation. Use itemized lists of points where appropriate. Text font size should be either 10 or 12 point.

IV. Marking Scheme

Your assignment will be marked by your tutor. If you have questions about a marked assignment, you should first ask your tutor before/after a tutorial. If you don't get satisfactory answers, you should talk to your instructor.

Marks for this assignment will depend on the following factors:

Description of your choice of feature to implement (20%): Did you clearly describe the criteria you used for your decision? Are the criteria appropriate? Is the choice sensible, given the criteria? Did you investigate existing resources (e.g. libraries, open source code, etc) that you could use to implement the features? Did you use appropriate tools such as estimation and risk analysis to help make the decision?

Description of your requirements analysis (20%): Did you analyze the requested feature from the user's perspective? Did you clearly distinguish between the user's goals and the functions that the software will provide? Is your list of required functions complete and appropriate? Did you set out a clear argument for why the specified functions will meet the user's goals? Did you make appropriate use of goal models and domain models to support this argument? Did you identify any domain assumptions used in these analyses?

Description of your use cases (20%): Did you identify an appropriate set of use cases for the requested feature? Did you draw a Use Case diagram? Are your use cases written from the users' perspective? Would the use cases be suitable as the basis for a user manual? Did you clearly document the normal and alternative paths for each use case? Did you identify exceptions? Did you identify pre- and post-conditions for each use case? Did you identify relationships between use cases (e.g. "uses", "extends") where appropriate?

Description of your robustness analysis (20%): Did you complete a robustness diagram for each major use case? Does your robustness diagram correctly distinguish boundary classes, control classes and domain classes? Does your robustness diagram indicate which classes will be newly added, and which already exist?

Presentation (20%): The style of your presentation, including language, grammar, clarity of the presentation, layout and legibility of the diagrams, etc. (10% - Language; 10% - Style and clarity)