University of Toronto
Department of Electrical & Computer Engineering

CSC444F – Software Engineering I

Instructor: Matt Medland
September 22, 2015

Assignment 1: Reverse Engineering & Design Recovery

Due Date: Tuesday, October 13, 2015 at the start of lecture

This assignment counts for 7.5% of the final grade.

Reverse-engineer a set of models from the source code you have been given, to explain its design. The goal is to use UML, and other modeling techniques of your choosing, to highlight the structure and behavior of the code. The assignment requires you to use your judgment about which aspects of the design to model, how much to abstract away from the code-base, and which parts of UML to use. There is no "correct" answer – your models will be judged for how well they explain the most interesting and important aspects of the design. The project is to be carried out in your assigned teams. Each team will submit one report.

Doing the Assignment:

This assignment has seven (7) steps. They are:

1. Familiarize yourself with the existing codebase. Check it out of the repository, attempt to build it (optional, but recommended), and browse the directory structure.

2. Familiarize yourself with the tools available to help you with this assignment. You will need a UML drawing tool of some kind (simple lightweight tools will work fine for this), and, optionally, a reverse engineering tool. Identifying and evaluating suitable tools is part of the exercise.

3. Create UML package and class diagrams representing as many of the important packages, classes, subclass relationships, and associations in the source code as is feasible. You can either do this by hand, or find a suitable reverse engineering tool to do it for you. If you use a reverse engineering tool, note that you will still need to edit the generated model to capture information missed by the tool, to remove unnecessary detail, etc.

4. Draw a higher-level diagram to show the overall architecture of the system. Use any appropriate UML notation (e.g. packages, components, interfaces, etc.) Be sure to show clearly where each package belongs in this architecture, and what external packages the system interacts with.

5. Identify at least two (2) different design patterns used in the system, and show how each is implemented using UML diagrams. Be sure to illustrate the pattern with both structural views (e.g. class or object diagrams) and behavioral views (e.g. sequence, activity, etc. diagrams).

6. Write a report that describes the steps you went through to reverse engineer the design and to produce the required models.

7. Document your teamwork by completing team member evaluation forms, which are available on the course website.

What to Submit:

Hand in your report by emailing it to your instructor in PDF format, as well as submitting a hard-copy at the beginning of lecture on the due date. The report should not exceed fifteen (15) pages; not counting cover pages, appendices, and teamwork forms. It should include the following items:

1. A brief description of the reverse engineering process(es) you used, including the tools you used (if any) and the abstraction steps you took.

2. A commentary on the architecture of the system, highlighting any interesting aspects of the design (e.g. architectural style, degree of coupling, etc.), discussing the quality of the architecture used in the system, and suggesting possible improvements where appropriate. Use UML diagrams to illustrate the points you wish to make.

3. A description of each of the design patterns you identified, along with both structural and behavioral UML models (suggestion: limit yourself to no more than 2 pages per pattern).

4. Other UML models you generated, as appropriate, included in appendices. It is impractical to include enough views to show all classes in the entire project. Higher-level package diagrams should show all major components. A deeper dive into one or more of the packages should show all classes within, at least down to class names.

Be sure to include a cover page indicating the name of your team, the names and student numbers of all team members, title of work, course, and date. Assignments will be judged on the basis of visual appearance, the grammatical correctness, quality of writing, 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.

Marking Scheme:

This assignment will be marked by your TAs. If you have questions about a marked assignment, you should first ask the TAs by email. 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 process (10%): Did you identify and evaluate suitable tools? Did you make good (sensible) choices? Did you describe the decisions you took in abstracting higher-level models from the code? Did you critique your process, and reflect on what you learned?

- Description of system architecture (20%): Did you clearly show the overall architecture? Is your architecture an accurate representation of the design of the system? Did you critique the architecture, identifying both strengths and weaknesses? Did you make good use of UML sketches to make your points?

- Description of Design Patterns (20%): Did you identify at least two (2) design patterns? Are these patterns genuine? Did you clearly show how the pattern is implemented in the code? Do your UML diagrams correspond accurately to the code? Did you show both structure and behavior for these patterns?

- Remaining UML models (25%): Did you provide UML diagrams with enough detail to see the design of the entire system at least down to package/module names? Have you eliminated other detail to ensure these diagrams are still readable? Did you make good use of different UML modeling constructs to describe the system? Did you produce a full class diagram for at least one of the major modules?

- Presentation (15%): The style of your presentation, including language, grammar, clarity of the presentation, layout and legibility of the models, etc.

- Interview (10%): Did you present your work well during the interview? Did you work effectively as a team? Did you demonstrate active involvement of all team members in the project? If your teams had problems, did you address them effectively and in a timely manner?