605.707 – Final Project

The final project is a paper presenting an analysis of the use of patterns or potential for pattern use in a piece of open source software. The software must meet the following requirements:

1. It must be open source. I should be able to find the source online myself.
2. It must not be written by any of the following people: yourself, friends, co-workers, fellow students, the instructor.
3. The software must be written in a language that the instructor can understand: C, C++, C#, Java, Ada, csh, Executable UML. I'll consider others like LISP or Scheme but prefer not.
4. The software must be sufficiently complex to support a rich set of patterns. What usually works best is to find a medium size program and to focus on one part of it.

The paper will cover the following topics:

- The architecture of the software (UML diagrams are encouraged). This is a description of the overall purpose of the software, the high-level structure, and where in that structure is the software containing the patterns that you will discuss. (10% of the content)
- Patterns encountered in the software or that should be applied to the software. This should include patterns at all levels from architecture to idiom. It is preferable to concentrate on 3-5 patterns and treat them in depth than to gloss over many. (50%) Discussion should include:
    - Implementation topics discussed in class such as the boundary conditions of the pattern (e.g., how is a Strategy initialized or when is a Singleton destroyed)
    - Differences from the canonical implementation
    - If the structure is similar to another pattern, why it is one vs. the other
    - Which of the various types of coupling and cohesion are realized by the pattern's implementation
    - How modularity and flexibility are improved by the pattern

- Appropriateness of the patterns, i.e., was the pattern used appropriately in the software, matching the intent given by the GoF. Or if you are discussing the application of patterns to the software, justify their use. (40%)

Special consideration will be given to any new patterns found in the software, though due diligence must be exercised to show that the pattern has not been previously documented.

The paper should be 10 to 20 pages. I favor shorter papers and quality over length. It is due by email before the beginning of the final module. It is not necessary to reproduce the GoF structure diagrams. Diagrams should be of the code that implements the pattern, rather than the ideal pattern itself.

There are two interim deliveries for the final project.

- A decision on the project topic must be provided six weeks into the semester. This needs to be a few sentences in an email.
- A draft of the project is due three weeks after that. The draft should be a page or so in length, and identify 2-3 of the patterns with brief descriptions. This is intended as a check that the chosen program has patterns or pattern opportunities to find, and as an opportunity to provide feedback.