

CS 578 – Software Architectures Spring 2011

Homework Assignment #3

Due: *Monday, April 4, 2011*

– *see course websites for submission details* –

Introduction

In this assignment, you will continue your study of the architecture of OODT by identifying instances of *architectural bad smells* in the OODT source code. An architectural smell is defined as a commonly (although not always intentionally) used architectural decision that negatively impacts system lifecycle qualities. System quality tradeoffs arise from either keeping each smell or removing it. Architectural smells are acceptable in cases where the removal of an instance of a smell would result in an unacceptable reduction of other system qualities. This paper explains smells in detail and provides some examples of them:

Joshua Garcia, Daniel Popescu, George Edwards and Nenad Medvidovic, *Toward a Catalogue of Architectural Bad Smells, Proceedings of the 5th International Conference on the Quality of Software Architectures (QoSA): Architectures for Adaptive Software Systems*, June 2009.

You are asked to identify 4 instances of different architectural smell types. Two of these instances must be different types of architectural smells from the paper above. The other 2 smells need not be different types of smells. For example, you can identify one instance of an Ambiguous Interface, one instance of Connector Envy, and two more Ambiguous Interface instances. Besides identifying smells in OODT, you will also be asked to describe how the particular smell instance could be refactored or restructured, how maintainability is affected by each smell instance, what other non-functional properties may be affected by refactoring or restructuring the smell instance (e.g., performance or scalability), and whether you think the smell instance in OODT should be refactored or left alone.

Deliverable

The deliverable of your assignment is a written assessment report. The report should be written using the font Times New Roman in font size 12 and single line spacing for each paragraph. Please check your spelling and use proper grammar.

Please adhere to the following format when you write your report on architectural smells.

- 1) Identification of the architectural smell in OODT. Discussion of where in the source code the architectural smell is located and what architectural elements are involved.
- 2) How does this particular smell negatively affect maintainability in this specific instance?
- 3) How could this smell be refactored or restructured so that OODT would no longer contain this smell instance?

- a. How does this refactoring or restructuring exhibit approximately the same functional behavior as before? (Remember that a refactoring is not supposed to change the externally observable behavior of the software elements involved).
 - b. What other non-functional properties may be affected by this refactoring or restructuring?
- 4) Would you recommend refactoring this particular smell instance or leaving it alone given your discussion above of the effects of refactoring?