15-413: Introduction to Software Engineering

Jonathan Aldrich

Assignment 7: Design

Due: Monday, October 24, 11:30am (hardcopy at beginning of class)
50 points

This assignment is a group assignment. Each project group should turn in one response to each part, with all the names of the group members.

Parts of this assignment depend on interaction with clients. Occasionally, you may not be able to meet with clients for reasons outside your control, e.g. they are traveling. If this is the case, contact the instructor or TA to get an extension on the assignment. However, you are expected to contact your client promptly so as to avoid any possible delays.

1. Information Hiding (30 points)

a) Describe the decomposition of your system into modules. You may choose the level of granularity to work at, and you may choose to focus on only a part of your system. The part of the system you describe should have at least 5 modules. For each module, explain (i) What the purpose of the module is, (ii) Describe its interface, and (iii) Describe the design decision that is hidden inside the module. Your answer should total about a paragraph for each module.

b) Describe at least 4 scenarios for likely changes in your system (2-3 sentences for each scenario). Discuss this with your clients if possible, to make the exercise realistic. Try to think of challenging scenarios, so that some of them involve changes to interfaces in your EDSM below.

c) Draw an environment and design structure matrix (EDSM) for your module decomposition, with rows and columns for each of the modules, module interfaces, and change scenarios described in (a) and (b) above.

d) Discuss (i) the effect of each of the change scenarios on your EDSM. Also, be sure to explain entries in your matrix where (ii) one module depends directly on another module rather than on the module's interface, or (iii) a module's interface is affected by environmental change. For (ii) and (iii), discuss if there are alternative designs that might avoid these dependences, and what might be the tradeoffs in adopting them (e.g. they might be less resilient to other changes, or more inefficient, or difficult to realize because of legacy code).

Note: this assignment will be graded on a basis of how clearly you analyze your design, ***not*** on how good your design is. So criticizing problems in the design is a good thing. This is because (1) we recognize that aspects of the design may not be under your control; they may be specified by clients or embedded in legacy code, and (2) ideally the assignment will help you to improve your design.

EDSM references:

K. Sullivan, W. G. Griswold, Y. Cai, B. Hallen, "The Structure and Value of Modularity in Design", ESEC/FSE 2001, September 2001.

- http://www.cs.virginia.edu/~sullivan/publications/ESEC_FSE_2001.PDF
- This paper introduced EDSMs. It includes a very nice explanation of DSMs (which are EDSMs without the change scenarios) and motivates the addition of environments. Their change scenarios are more abstract than the ones used in class; I find it easier to analyze concrete change scenarios and you should be concrete as well. For this assignment, you can skip or scan the sections about computing numerical values for different designs; the qualitative value is what's important.

K. Sullivan, W. G. Griswold, Y. Song, Y. Cai, M. Shonle, N. Tewari, and H. Rajan "Information Hiding Interfaces for Aspect-Oriented Design", ESEC/FSE'05, September 2005.

- http://portal.acm.org/citation.cfm?id=1081706.1081734
  - Accessible from any computer with a CMU IP address
- This paper is slightly more practical than the one above, but assumes some knowledge of aspect-oriented programming; it applies EDSMs to analyze alternative aspect-oriented programming modular decompositions.

2. Design Patterns (20 points)

The purpose of this part of the assignment is to identify and analyze design patterns in an application. If you are developing your project in an object-oriented language and style, find design patterns in your project. Otherwise, use the source code to some other application (and describe the application in a paragraph or so, including where you got it).

Find 3 instantiations of different design patterns in the project. For each instantiation, describe (a) which design pattern is instantiated, (b) participants of the pattern in the system, (c) classes and code snippets relevant to the design pattern and their locations. For example, if you found an instantiation of the Strategy pattern, the answer should be (1) Strategy, (2) Context corresponds to MyAppContext, ConcreteStrategy corresponds to MyAppConcreteStrategy, and Strategy corresponds to MyAppStrategy.

If you are certain that a particular design pattern appears in the code, but cannot tie code snippets to the participants described in the GOF book, give us your justification. If you

cannot find enough design patterns in the code, discuss places where design patterns could have been applied in a useful way.

Design Patterns References

- Design Patterns, Gamma, et al.; Addison-Wesley, 1995; ISBN 0-201-63361-2; CD version ISBN 0-201-63498-8
    - This is the definitive reference on design patterns. It's definitely worth buying.
    - I'll have at least one copy of the book that one of the groups can borrow next week.
- http://www.research.ibm.com/designpatterns/pubs/dp-tutorial.pdf
- http://www.cs.unc.edu/~vivek/patterns/doc/