Homework 4 (2007-09-28)

Submission deadline: 2007-10-04

Question 1: Consider the classification of Design goals (given in Section 6.4.2). Assign on or more categories of design goals to each of following goals.

1. User must be given feedback within one second after they issue any command.
2. The ticketdistributor must be able to issue train tickets, even in the event of network failure.
3. The housing of ticketdistributor must allow for new buttons to be installed in case the number of fares increases.
4. The atm must withstand dictionary attacks (i.e. users attempting to discover an identification number by systematic trials).

Question 2: Consider closed and open architectures (Fig 6-10 and Fig 6-11).

1. Compare design goals which are achieved and which could be difficult to meet in each (use bullets, first analysis design goals for Open and then similarly discuss the same for closed architecture).
2. Under which scenario (set of design goals) you will consider developing an Open architecture and under which developing a Closed architecture.

Question 3: Consider Repository, Model/View/Controller, Client-Server, Peer-to-Peer, n-Tier, and 'Pipe and filter' architectural styles (Section 6.3.5). Discuss how these architectures affect the following design goals. I) Extensibility II) Response Time III) Modifiability IV) Access Control V) Maintenance cost VI) Adaptability.

1. Consider a concrete example first (preferably a single example) and then try to decompose into above mentioned architectural styles. And then discuss how architecture affects the mentioned design goals. (Use bullets against each design goal, better create a table comparing architectures with respect to design goals)
2. Make sets of design goals (not just limited to above mentioned design goals, consult section 6.4.2) for each architectural style that could serve as guideline for choosing certain system architecture.

Question 4: Consider eBay auction. Customers browse and view products and make an offer before end-date of given auction or buy the article on 'But-it now price'. Users are also registered with system and create profiles etc. etc. ….. System registers buyers, displays them the items active in auction etc. etc. …. You can check eBay's process. Formalize the general usage of eBay in terms of actors you can identify. And give a very brief usage scenario for each identified actor. (These will form assumptions for your answer)

Your task is to design an access control policy or discuss already in-use access control policy on eBay.

Question 5: List the factors due to which you can't describe boundary use-cases in Requirements elicitation or Requirement Analysis phase?

Question 6: Tetris field comprises of a collection of stones, some representing empty fields, and some representing parts of a particular Tetris block, whenever a Tetris block lands, it dissembles into its individual stones, which are considered independent from then on. It makes sense, therefore, to represent each stone as an individual object in an object-oriented Tetris application. Doing so naively results in a large amount of objects being allocated, deleted, and moved all the time. What design pattern can be applied for optimization, taking advantage of the fact that the stones are really very similar?

Question 7: What design pattern(s) will you use if you are supposed to realize following commands in your own developed UNIX shell:

```
$ cat apple.txt | wc | mail -s "The count" nobody@december.com
```

Question 8: List at least 3 general dangers and benefits of using design patterns.

Question 9: Develop a recursive composite design pattern and represent it using UML Class diagram. (Bonus Point 02)

Question 10: Based on the problem description for "Advert Consultancy Inc" (Introduced in Question 5 of Homework 2), and you results from homework 2 and homework 3, and the examples starting on pages 288, 337 and 380 of the textbook, **Develop a system design and object design document** which includes:

1. Subsystem decomposition structure (using class diagrams).
2. Mapping subsystems to processors and components - hardware/software mapping (using UML deployment diagrams).
3. Persistent storage solution for the problem.
4. Access control, global control flow and boundary conditions.
5. Applying design patterns to designing object model for the problem (using class diagrams).
6. Writing contracts for noteworthy classes

Question 10 is Important, please do it carefully and completely

Question 11: Consider a simple intersection with two crossing roads and four traffic lights. Assume a simple algorithm for switching lights, so that the traffic on one road can proceed while the traffic on the other road is stopped. Model each traffic light as an instance of a TrafficLight class with a state attribute that can be *red, yellow, or green*. Write invariants in OCL on the state attribute of the TrafficLight class that guarantee that the traffic cannot proceed on both roads simultaneously. Add associations to the model to navigate the system, if necessary. Note that OCL constraints are written on classes (as opposed to instances).

Question 12: Consider a sorted binary tree structure for storing integers. Write invariants in OCL denoting that

- All nodes in the left subtree of any node contain integers that are less than or equal to the current node, or the subtree is empty.
- All nodes in the right subtree of any node contain integers that are greater than the current tree, or the subtree is empty.
- The tree is balanced.