

**Software Patterns (INSP-06)**  
**Examination (Tue. April 24, 2007)**

**You are allowed to use books and course notes during the examination**

**Question 1 (5 points)**

You are the architect responsible for designing a system similar to eBay. The following (incomplete) requirements have been assembled by the requirement engineer:

1. The system will support online auctions.
  - a. There are several possible scenarios for how auctions are run, so have flexibility in setting up and running auctions.
2. The system will support searching through active and past auctions.
  - a. There will be several possible search modes, such as search by general category of merchandise, keyword in item title, by seller, by location of item (e.g., search only auctions in a particular country.)
  - b. It is expected that this database will grow large over time, and as you can imagine, it is very dynamic. Auctions are starting and finishing all the time.
3. There will be user accounts.
  - a. They allow users to set preferences.
  - b. They allow buyers to see which auctions they are involved in.
  - c. They allow sellers to see their active auctions.
  - d. They allow people to rank satisfaction with sellers and buyers.
  - e. A person may be both a buyer and a seller.
4. The system handles payment by handing the transaction off to a third-party payment system.
5. The system has strong security needs. Personal information must remain safe, and auctions must not be compromised. Transactions (e.g., bids) must be secure.
6. The system must be up 99.99% of the time.
7. The system must handle transactions reliably. For example, since networked communication is vulnerable to disruption, partial transactions must be handled or prevented.
8. Performance must be consistent. In particular, an auction must end exactly when it says it will, and bids must be handled promptly and consistently.

Assume that you can propose changes to these requirements if you come up with convincing reasons or if your architectural decisions result in relevant tradeoffs. Of course as an architect you can prioritize the requirements according to your judgment. Your tasks are the following:

- a. Make a preliminary architectural design of the system by applying at least four patterns. The patterns can be architectural, enterprise, design, or other kinds. If you use patterns that were not covered in the lectures (i.e. from sources other than POSA, GOF and Fowler) briefly explain them. You do **not** have to tackle **all** of the above requirements. For each applied pattern elaborate on:
  - o the requirement(s) it tackles,

- the specific variant of the pattern that you use and why,
  - which other alternative patterns could be applied for the same purpose and why you selected the particular one.
- b. Explain how the patterns are combined in the system design (for example a pattern delegates part of its solution to another pattern) and if certain components undertake responsibilities from more than one pattern.
  - c. Evaluate the impact of the combined patterns to the system's quality attributes. Consider at least 3 Quality Attributes, explain what tradeoffs have been made and justify them.

### **Question 2 (2 Points)**

By giving an example (you may refer to the eBay system of Question 1 if you want), answer the following:

- a. The benefits and liabilities of patterns are explicit but usually generic and qualitative. How does an architect reason about the consequences of an individual pattern when applying it into the system's design? (1)
- b. An architect usually needs to combine a number of patterns. How does this combination affect the overall system and especially the quality attributes? (0.5)
- c. If many patterns are combined in a system design then an element may undertake responsibilities from different patterns. How does this affect the understandability of the design? (0.5)

### **Question 3 (3 Points)**

- a. Why do the descriptions of patterns contain variability in the solution? Give an example of two variants in a pattern (do not use the Pipes and Filters) and explain their difference in benefits and/or liabilities. (1)
- b. Pattern languages can tackle complex problems by combining patterns, but they are not 'silver bullets'. What are the disadvantages or limitations of pattern languages? Give an example for every disadvantage/limitation. (1)
- c. Do you consider "Exception Handling" as a pattern? Justify your answer. (0.5)
- d. Provide a case and an example where the application of a pattern can degrade the quality of a software system. (0.5)