

CS578: Software Architectures Spring 2006

Homework #3

**Due Date: Tuesday, February 21
before 3:30pm – *no exceptions***

1. (15%) Parnas in his 1972 paper on “On the Criteria To Be Used in Decomposing Systems into Modules” said that software components should hide likely changes

- a) what did he mean with that? Why is it useful to hide likely changes?
- b) what are likely changes?
- c) what kinds of likely changes can be hidden elegantly? List and describe three.
- d) what kinds of likely changes cannot be hidden elegantly? List and describe three.
- e) what did Parnas say about software composition? (note: the answer may be surprising and not obvious. Provide a rationale why the answer is this way!)

2. (10%) What constraints on components are imposed by

- a) Functional requirements
- b) Non-functional requirements
- c) Envisioned future evolution of the software
- d) System scale
- e) Computing environment
- f) Customers/users

Note: we are looking for constraints on how the components are affected and not their construction process.

3. (5%) What is a reasonable/adequate component granularity level? What is too fine grained? What is too coarse grained?

4. (15%) What assumptions can components make about one another?

- a) There are benefits in components knowing about one another if they are to communicate. Briefly describe three!

b) There are benefits in components not knowing about one another even though they do communicate. Briefly describe three!

c) What architectural styles support a). Which ones support b) Explain your answers!

5. (10%) How can components interact? List 10 different interactions and briefly describe them!

6. (5%) If you are to understand a piece of code (class, package, component), what information do you need to extract?

7. (5%) What roles does a connector have? Briefly describe each role!

8. (35%) You are to design a new email system for USC. The management hands you the following requirements:

1. Must have an account for each USC student, faculty and staff
2. Each account holder must get a storage space of 5GB
3. Must support standard protocols for sending and accessing email (SMTP, POP etc)
4. The main screen should flash important information about the classes the student is taking.
5. Advertisements should be shown when an email is opened. The advertising should be relevant to the content of the email
6. Each advertiser is charged proportional to the number of times his advertisement is displayed
7. All users must get an account for their lifetime (not closed when they graduate).
8. Must be built in 6 months
9. Must have a downtime of less than 1 hr in a year
10. Only authorized users must be able to use the system

a) Use your intuition to build a Level-1 architectural solution (component diagram and deployment diagram) that satisfies the above requirements (note: do this before you do the CBSP approach in b)).

b) Use CBSP to build a Level-1 architectural solution (component diagram and deployment diagram) that satisfies the above requirements. Make sure to follow the process outlined in the Gruenbacher et al paper. List intermediate results such as how requirements are mapped to CBSP artifacts or the mapping between requirements and architectural building blocks.

- c) Reason about the strengths and weaknesses of the architectural solutions
- d) Reason about the strengths and weaknesses of the process you followed to construct them
- e) Refine two Level-1 components of your choice into a Level-2 architecture – make sure the components are in fact from the same architectural solution and they are communicating with one another in the Level-1 architecture (~10 sub components).
- f) Provide the traceability from architecture to requirements (for each component *and* connector, list which requirement is responsible for it). Also, make sure to point out how the various “ilities” were addressed in the architecture.

Submission Instructions: see class web site at http://sunset.usc.edu/classes/cs578_2006/