

CS554 - Designs for Software and Systems

Student name: _____

(You may answer either in English or in Korean.)

[Questions 1-7] Answer the following questions. (10 points each)

[1] The famous Brook's Law states that "Adding manpower to a late software project makes it later". Discuss why men and months cannot simply be interchangeable in a software project.

[2] When comparing the old "vertical" structure of computer industry with the modern "horizontal" structure of computer industry driven by PC revolution, what kind of software engineering problems do we need to address nowadays? Discuss your opinion.

[3] What is an architectural style? Give three examples and explain them briefly..

[4] How can the activity of a filter in the pipe-and-filter architecture be triggered?

[5] Suppose someone told you they were using implicit invocation, except that by convention there is always exactly one recipient of each event. And further, the announcer of the event expects a return event from the receiver carrying a result. Is this really implicit invocation? Why?

[6] Discuss the basic approach to problem solving in the blackboard framework.

[7] Why is there a common restriction against a lower layer accessing a higher layer? What kinds of problems could this cause?

[Questions 8-10] As for software quality attributes (or non-functional requirements), answer the following questions. (5 points each)

[8] What are the runtime quality attributes of good software? List at least four quality attributes and explain them.

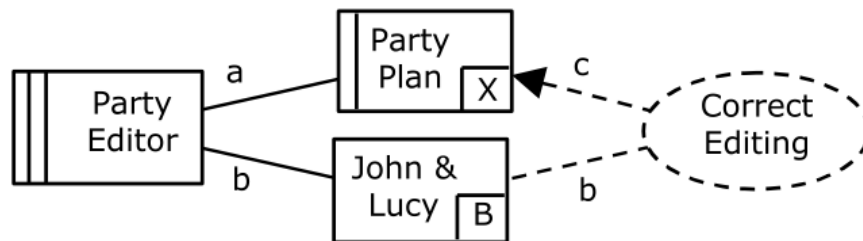
[9] What are the non-runtime quality attributes of good software? List at least four quality attributes and explain them.

[10] Systems often fail to meet user needs when designers narrowly focus on meeting some requirements without considering the impact on other requirements. List two examples for such impact or conflict among requirements for one system.

[Question 11] As for Michael Jackson’s problem frames, answer the following question. (15 points)

[11] The figure below is a problem frame specification for a “Party Plan Editing” problem described as follows:

- Lucy and John give many parties, to which they invite many guests. They want a simple editor to maintain the information, which they call their Party Plan.
- The Party Plan is a list of parties, a list of guests, and a note of who's invited to each party.
- The editor will accept command-line text input, in a (very old-fashioned) DOS or Unix style.
- We are not at all concerned with presenting or printing the information - just with creating and editing it.



a: PE!{PlanOperations}
PP!{PlanStates}

b: JL!{Commands}
c: {PlanEffects}

Explain everything (i.e., every component appeared in the figure) about this problem frame.

(The End)