KTH Microelectronics
and Information Technology

Exam in ID2207 Modern Methods in Software Engineering, 2017-10-26, 08:00-12:00

Rules
This exam is “closed book” and you are not allowed to bring any material or equipment (such as laptops, PDAs, or mobile phones) with you. The only exceptions are English to “your favorite language” dictionary and pencils.

Instructions
- Please read the entire exam first!
- Write clearly
- Each sheet of paper must contain your name, ”personnummer”, Problem number and a unique sheet number
- Write only on one page of a sheet. Do not use the back side
- Sort your sheets according to the problem’s numbering
- Only one Problem must be reported on each sheet
- If more than one sheet is needed the continuation should be clearly noted on the beginning of each sheet and the sheet numbers used should be consecutive
- Always motivate your answers. Lack of clearly stated motivation can lead to a reduction in the number of points given
- The tasks are not necessarily sorted in order of difficulty. If you get stuck it might be a good idea to go on to the next task.

Grading
If n is amount of your exam points and m is your bonus points earned in autumn 2014 then:

$n+m < 50$ fail (F)
$50 \leq n+m < 60$ grade E
$60 \leq n+m < 70$ grade D
$70 \leq n+m < 80$ grade C
$80 \leq n+m < 90$ grade B
$90 \leq n+m$ grade A

GOOD LUCK!

Mihhail Matskin, mobile 0704614269

Problem I. General questions

a) What is difference between the problem domain and the solution domain?

嗯 problem domain代表各种用户问题包括什么物理环境啊什么的 对于开发者很难去理解 然后solution就是一个给系统建模的过程吧应该是 代表系统设计和对象设计 比problem更易于变化 因为系统设计总是越来越详细 一些新型的技术和对项目越来越深的了解也会影响solution

(4p)

b) A programming language is a notation for representing algorithms and data structures. List two advantages and two disadvantages of using a programming language as sole notation throughout the development process.

Advantages:

- Developers need only learn one notation for all development activities.
- Traceability among models and between models and code is made easier since they are written in the same notation.

Disadvantages:

- A programming language is a low level notation which is difficult to use for representing user requirements, for example.
- A programming language enables and encourages developers to represent implementation details too early.

4p)

Problem II. Software Life Cycle

a) Specify which of the following decisions were made during requirements or system design:

- ***"The ticket distributor is composed of a user interface subsystem, a subsystem for computing tariff, and a network subsystem managing communication with the central computer."***
- ***"The ticket distributor will use PowerPC processor chips."***
- ***"The ticket distributor provides the traveler with an on-line help."***

(4p)

b) What is difference between activities and tasks? What are main Software Engineering activities?

(4p)

c) Explain inductive program synthesis. What is/are its advantage(s) and what is/are its drawback(s)?

(4p)

Problem III. UML and OOP

a) Organize the following set of classes into hierarchies: Circle, Point, Rectangle, Matrix, Ellipse, Line and Plane. You can add any number of additional classes into your hierarchy.

(5p)

b) Can the system under consideration be represented as an actor? Justify your answer.

The system under consideration is not external to the system and shouldn't be represented as an actor. There are a few cases, however, when representing the system as an actor may clarify the use case model. These include situations where the system initiates uses cases, for example, as time passes (Check for Outdated Articles, Send Daily Newsletter).

(3p)

c) Show how condition and iteration can be presented in a sequence diagram.

(4p)

Problem IV. Requirements Elicitation

a) We identified the following types of scenario: As-Is, Visionary, Evaluation and Training scenario. Briefly explain each of them and give examples.

(4p)

Multiple-choice questionnaires provide possible answers to the questions that are asked. This introduces two problems: first, the analyst must be familiar enough with the application domain to offer a good set of answers for each question. Second, the user is constrained to select these pre-designed answers. It is then preferable that the analyst use other elicitation methods, such as task analysis or unstructured interviews, to gather sufficient knowledge of the application domain and to discover implicit knowledge that the user assumes everybody has. Subsequently, the analyst can use multiple choice questionnaires to confirm a hypothesis or to prioritize certain functionality

b) Explain why multiple-choice questionnaires, as a primary means of extracting information from the user, is not effective for eliciting requirements.

(4p)

c) Specify which of these requirements are verifiable and which are not:

- ***"The system must be usable."***
- ***"The system must provide visual feedback to the user within 1 second of issuing a command."***
- ***"The availability of the system must be above 95%."***
- ***"The user interface of the new system should be similar enough to the old system such that users familiar with the old system can be easily trained to use the new system."***

(4p)

Problem V. Requirements Analysis

a) Both sequence diagrams and statecharts diagrams can be used in analysis. When each of them is more appropriate?

(4p)

b) Let us consider a scenario consisting of selecting a file on a memory stick, dragging it to Folder representing a file system on your computer and releasing the mouse. Identify and define at least one control object associated with this scenario.

(4p)

c) Consider a traffic light system at a four-way crossroads (e.g., two roads intersecting at right angles). Assume the simplest algorithm for cycling through the lights (e.g., all traffic on one road is allowed to go through the crossroad while the other traffic is stopped). Identify the states of this system and draw a statechart describing them. Remember that each individual traffic light has three states (i.e. green, yellow, and red).

(5p)

Problem VI. System Design

a) What are criteria for choosing file system or database as a data management solution?

(5p)

b) What are advantages and disadvantages of open and closed layered architectures? Which of them is more suitable in which cases?

(4p)

Problem VII. Object Design - Reuse

a) What is Observer pattern? Where is it applicable? Give an example

(5p)

b) Indicate which occurrences of the following inheritance relationships are specification inheritance and which are implementation inheritance:

• A Rectangle class inherits from a Polygon class.

• A Set class inherits from a BinaryTree class.

• A Set class inherits from a Bag class (a Bag is defined as an unordered collection).

• A Player class inherits from a User class.

• A Window class inherits from a Polygon class.

(5p)

Problem VIII. Object Design – Interface design

a) Explain meanings of invariant, pre-condition and post-condition. Give their examples.

(4p)

b) Assume that you have the following 3 classes: parallelogram, rhombus and square. Assume that rhombus inherits from parallelogram and square inherits from rhombus. Write invariants in OCL for each of these classes taking into account the inheritance of contracts.

(5p)

Problem IX. Moving to Code

a) Explain realization of a bidirectional, one-to-many association in source code.

(5p)

Problem X. Testing.

a) What are pro and cons for black-box and white-box testing?

(5p)

b) Compare advantages and disadvantages of bottom-up and top-down integration testing strategies.

(5p)

----------End of Exam----------