

Institute of Computer Science

MTAT.03.094 – Software Engineering (Tarkvaratehnika)

Final Exam – 6 January 2012

Notes:

- The exam is open-book and open-laptop. Web browsing is allowed, but you are not allowed to use e-mail clients nor Instant Messaging clients.

Context: Sangar.ee

Tartu-based shirt manufacturer Sangar (www.sangar.ee) is considering the possibility of expanding its online presence. Currently (in January 2012) Sangar's Web site offers standard e-shop functionality. The Web site allows customers to browse the catalogue of shirts sold by Sangar, to add shirts to the shopping cart and to check-out and pay online. Shirts are shipped to the designated address via postal services.

The application has received an average of 3000 visits per month since its deployment. About 20% of these visits lead to purchases. On average, each purchase is for 2 shirts and the profit margin per shirt is EUR 4.

In November 2011, Sangar started to consider the possibility of manufacturing and selling customized shirts through their e-shop. Customers would be able to design customized shirts by selecting the fabric, the type of collar, type of cuff, and the types of buttons. Customers would also be able to add a label (e.g. their full name or initials) to their shirt design. Once a customer has completed a shirt design, he/she can add one or multiple units of this shirt to their shopping cart. At each step during the customization process, the web site will display an image of the customized shirt as well as the price of the shirt. The price depends on the options that have been chosen (e.g. the type of fabric chosen). This "mass-customization" functionality is already offered by online shirt retailers such as Blank Label (<https://www.blank-label.com>).

One of the advantages of selling customized shirts is that they have a higher margin (EUR 8 per customized shirt, as opposed to EUR 4 for traditional shirts).

Sangar hired a consultancy company named RightConsultants to make a market analysis and to assess the potential benefits of selling customized shirts. The consultants concluded that if Sangar introduced into their e-shop site an option allowing customers to design their own customized shirts, the number of customers using the customization option would be about 2000 per month (in addition to the existing customer visits), and the conversion rate of customers who design customized shirts would be 50% – meaning that 50% of customers who design a customized shirt would end up buying it. Customers of customized shirts would buy on average 1.5 shirts per purchase. Each customized shirt has a profit margin of EUR 8.

Disclaimer: Although based on a real-world company, the above scenario is fictitious and does not reflect the current situation or plans of the company in question.

Part I: REQUIREMENTS SPECIFICATION

[40 points]

1) Why does this problem need to be solved?

Using language for the Knowledge Acquisition for automated Specification (KAOS)

- a) **Define a system goal model.** The initial goal model is provided in Figure 1. Ask questions “why”, “how” and “how else” to complete this goal model. Your model should result in at least 6 requirements and 3 expectations (potentially more), and illustrate at least 2 alternative goal refinements.

[10 points];

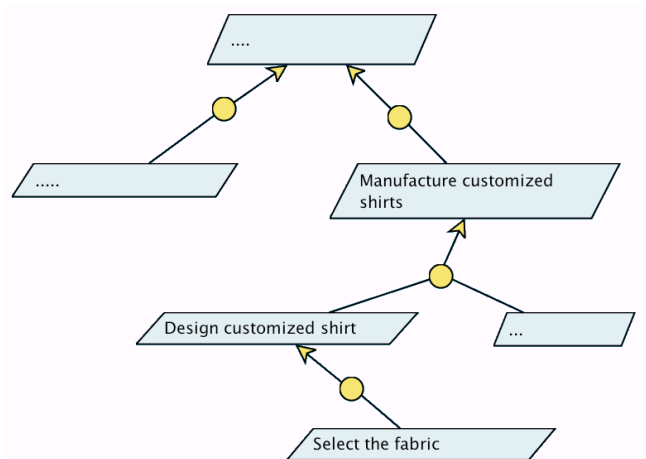


Figure 1. Initial goal model

- b) **Create an agent responsibility model.** Identify major environment and software agents, who are capable to fulfil requirements and expectations identified in your goal model (you do not need to re-draw the goal model, the agents can be assigned on the same goal model).
- c) **Prepare an operation model** for (at least) one requirement/expectation from your goal model.
- d) **Discuss, which system alternative** you would recommend to choose for the actual software system design, and **why?**

[5 points]

[5 points]

[5 points]

Answer:

Your answers will be evaluated (i) for the solution correspondence to the given problem, (ii) for the solution completeness regarding the requirements given for each task, and (iii) for the correctness of the language application (constructs and relationships).

2) What does the software system need to do?

[5 points]

Following the analysis performed in the previous task create a use case diagram. The use case diagram should include at least 5 relevant use cases, illustrate application of *includes* relationship. The *extends* relationship is optional.

You will be evaluated for *syntactic correctness* of the use case diagram, and the *consistency* between your use case diagrams and the KAOS models, as well as whether or not your use case diagram captures the *key functionality* of the envisioned system. You can start with the diagram given in Figure 2.

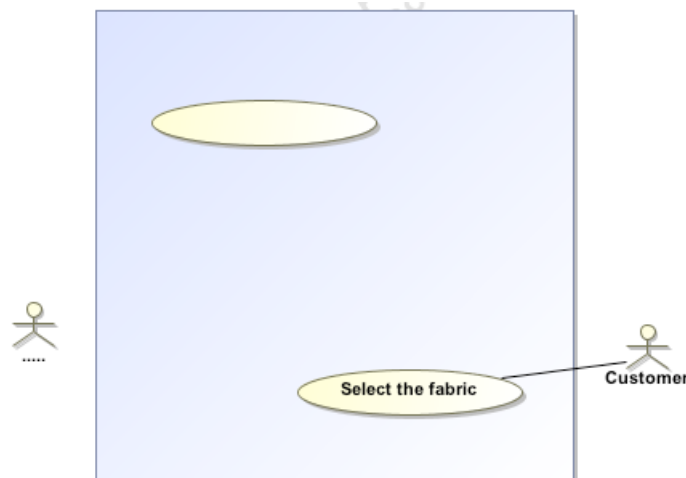


Figure 2. Initial use case model

3) How does the problem manifest itself?

[10 points]

Define 2 use cases for the envisioned system. Each of two use cases will be evaluated for 5 points. The score will be given for the *consistency* between the use case graphical and textual descriptions, for the *level of the details* filled in the template entries, and for the *correspondence* to the problem description.

Please use the “Use Case” template attached to the exam.

Part II: MODELLING

[30 points]

4) Identify the major classes and perform the static problem analysis by creating a class diagram for the given problem.

You will be evaluated for:

- Semantic relevance to the given problem:
 - Identify key classes;
 - Identify relevant attributes and operations;
 - Identify relevant relationships.
- Syntactic correctness and completeness of your diagram. The diagram should include:
 - Association relationships (with reading directions, and multiplicities);

- At least two association classes;
- At least one generalisation relationship.

You can start with diagram provided in Figure 3.

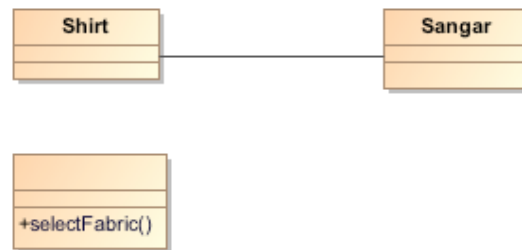


Figure 3. Initial use case model

Part III: CODE DESIGN

[30 points]

Analyse the code snippet given to you below. What are the code-smells or its design issues as to how the code can be improved?

```

01: public ArrayList doStuff(boolean jätaAegunudVälja) {
02:     // tõenäoliselt on vähemalt 25 tükki
03:     ArrayList<Contract> c = new ArrayList<Contract>(25);
04:     try {
05:         BufferedReader reader = new BufferedReader(new
06:         FileReader("lepingud.txt"));
07:         String l;
08:         while (true) {
09:             l = reader.readLine();
10:             if (l == null) break;
11:             String[] osad = l.split(";"); // semicolon is separator
12:             if ((osad.length > 10 && osad[9].equals("EX")) ||
13:             !jätaAegunudVälja) {
14:                 String agreementNr = osad[2];
15:                 if (!agreementNr.startsWith("0"))
16:                     agreementNr = "0" + agreementNr;
17:                 String clientName = osad[4];
18:                 try {
19:                     int a = Integer.parseInt(osad[5]);
20:                     Contract cc = new Contract(agreementNr, a, true, clientName, 12);
21:                     if (cc.toString(true).contains(":")) {
22:                         c.add(cc);
23:                     }
24:                 } catch (NumberFormatException e) {
25:                     // vahet pole vist
26:                 }
27:             }
28:         }
29:     } catch (IOException e) {
30:         System.out.println("viga");
31:     }
32:     return c;
33: }
  
```

You will be evaluated for number of problems/improvements (both general ones and line by line cases) identified.

Use case ID	ID-1		
Use case name	Select fabric		
Created by	Author	Last updated by	Author
Date created	06.01.2012	Date last updated	06.01.2012
Actor	Customer		
Description			
Trigger			
Pre-condition			
Post-condition			
Normal flow			
Alternative flows			
Exceptions			
Includes			
Priority			
Frequency of use			
Business rules			
Special requirements			
Assumptions			

Use case ID	ID-2		
Use case name	Show current price		
Created by	Author	Last updated by	Authors
Date created	06.01.2012	Date last updated	06.01.2012
Actor			
Description			
Trigger			
Pre-condition			
Post-condition			
Normal flow			
Alternative flows			
Exceptions			
Includes			
Priority			
Frequency of use			
Business rules			
Special requirements			
Assumptions			