Softwaretechnik/Software Engineering

`http://swt.informatik.uni-freiburg.de/teaching/SS2019/swtvl`

Exercise Sheet 2

Early subm. (Ex. 2+3): Wed, 2019-05-22, 12:00   Regular subm. (Ex. 2+3): Thu, 2019-05-23, 12:00

Exercise 1 – Requirements Elicitation   (8/20 Points + 5 Bonus)

Recall the video game project from Exercise Sheet 1. One of the requirements from the customer was the following:

> "The game must include at least 10 different actions (incl. run, capabilities, etc.)."

> ("Das Spiel muss mindestens 10 verschiedene Aktionen (inkl. Laufen, Fähigkeiten, usw.) enthalten.")

(i) Clarify the meaning of the terms that the requirement depends on, as intended by the customer[1]. Present the results in form of a dictionary.

For at least 4 terms in your dictionary, give one plausible interpretation that is accepted by the customer, and one plausible interpretation that is not accepted.   (3)

Real-world example: if the new flatmate says "Oh, could you fetch some milk from the supermarket later?", an RE-person would start to ask "What do you mean by 'milk'? Cow, sheep, camel, anything else…?"

(ii) Clarify with the customer[1] whether and which exact constraints there are on the considered terms in order to accept the delivered video game.   (3)

If cow-milk is it and 'some' means between 1 and 3 litres, the RE-person would continue to ask "Fat-degree? UHT or fresh or … ? Bottle or … ? Brand? Any bio-label?". And flatmate may add "Oh, if they have the 0.5l bottles of Breisgau-Milch, bring two of those; I'd like to use these bottles as flower vase in my room."

(iii) Summarise your findings from the previous tasks in a comprehensible specification which describes your understanding of the set of all (!) solutions accepted by the customer.   (2)

[1]*For the course of this exercise, you need to interact with a real customer from the domain of student-developed video games. The task is to analyse what this customer accepts or does not accept. Since the customer wants to stay anonymous during this early stage of contract negotiations, you can only communicate with the customer via your software engineering tutor. That is, any enquiries you send to your tutor by mail (forum enquiries will not be considered) will be forwarded to the customer and you will receive the customer's reply. Note that, as a true customer, the person may actually not yet exactly know what is accepted, may misunderstand your questions, may give answers which you do not understand, may get annoyed by impolite questions, etc. In that case, just continue to send further questions: Do not give up easily. Note that, in the end, it is your job to clarify the requirements, not the customer's. Please be aware that answering questions may take some time. Expect a worst-case response time of three working days outside the holiday period. For the sake of traceability, quote your questions and examples, and responses you got from the customer in an appropriate way for further clarity and so that your tutor can comprehend how you came to your conclusions.*

> Fine print: This task is not related in any way to the running *Softwarepraktikum*, similarities are purely coincidental. Since the customer who you communicate with is anonymous, it may be a person from the *Softwarepraktikum* team or somebody completely unrelated. Thus exactly the answers from your software engineering tutor (and by nobody else) are authoritative exactly for this exercise (and for nothing else).
>
> This Exercise 1 is to be submitted by 6th of June only. That is, early submission is 2019-06-05, 12:00, regular submission is 2019-06-06, 12:00.
>
> Your tutor will inform you when the customer is ready for interactions.

Preparing the Second Meeting   **(5 Bonus)**

The goal of requirements analysis is usually to propose a specification that is a good compromise between creating a good game (to make your customer happy and to raise your company's reputation) and keeping the project costs in line with the budget (to make your developers and stakeholders happy). For this purpose, it is crucial to have an idea of the set of acceptable solutions. In the previous task, the requirements engineer has created a specification which captures his or her understanding of the requirements so far. This specification is to be validated in an upcoming meeting.

(iv) Prepare this meeting by sketching the *relevant aspects* of

- one example game which satisfies the specification but for which you argue that this example is far too expensive to be realised in the given schedule by the given team (cf. Exercise Sheet 2), thus a game which you'd prefer not to deliver, and

- one example game which satisfies the requirements and which you would suggest to the customer as a good compromise (possibly in favour of the developers).

Exercise 2 – Analysis of Decision Tables   **(6/20 Points + 2 Bonus)**

Consider the decision tables shown in Figure 1.

(i) Are decision tables "DT2" and "DT3" *complete*?
Which of these decision tables are complete *without* considering the conflict axiom?   (2)

(ii) Are decision tables "DT2" and "DT3" *deterministic without* considering the conflict axiom?
                                                                         (2)

(iii) Do decision tables "DT1" and "DT2" have *useless rules without* considering the conflict axiom?   (1)

(iv) Is decision table "DT3" *consistent* with respect to *conflicting actions*?   (1)

Justify your answers with proofs or counterexamples.

Hint: One possible method for showing properties of a decision table is by creating a truth table *with all possible combinations of conditions and determining which rules and actions are applied to each combination. Another possible method is by deriving the formulas represented by the table and using the rules of propositional logic to show their validity, satisfiability, etc.*

(v) Extending on Task (iii), which of the decision tables "DT1" and "DT2" have useless rules (now considering the conflict axiom)?   **(2 Bonus)**

Note: In the lecture we have not defined when a decision table has *useless rules considering the conflict axiom*. Write down the definition that you used.

DT1	R1	R2	R3	R4
C1	×	×	−	*
C2	−	*	×	−
C3	×	−	*	×
A1	×	−	−	×
A2	×	−	×	×
$\neg [C1 \wedge C2 \wedge C3]$				

(a) Decision table DT1.

DT2	R1	R2	R3
C1	*	*	×
C2	−	*	×
C3	×	−	×
A1	×	×	−
A2	−	−	−
$\neg [C2 \wedge C3]$			

(b) Decision table DT2.

DT3	R1	R2	R3	R4
C1	×	×	×	−
C2	*	×	−	*
C3	−	*	×	*
A1	×	−	−	×
A2	×	−	×	−
$\neg [C1 \wedge C2 \wedge \neg C3]$				

Conflicting actions: A1 $\notmid$ A2

(c) Decision table DT3.

Figure 1: Decision tables.

Exercise 3 – Creation of Decision Tables   (6/20 Points)

Consider the following transcription of the interview with the customer who is looking for a software system to manage the lending management of a crowd-sourced book rental company.

- The company is in possession of a number of books. Some of them are owned by the company, and some are owned by clients.

- The business model is to lend the books to clients. If the book is owned by another client, this client receives $0{,}50 \in$ per lending transaction.

- The current primary goal of the company is market growth in the number of clients. To be attractive for more clients, the secondary goal is to increase the number of books offered.

- Clients are classified into three different groups, based on the number of books they offer: *newcomer (0–5)*, *supporter (6–50)*, and *professional (>50)*.

- With each order, a *supporter* receives a $3\,\%$ discount, and a *professional* receives a $5\,\%$ discount.

- Clients can only borrow one book per transaction.

- Clients can buy a *premium* membership. Orders are distributed to premium members with priority. A premium membership costs $2 \in$ per month.

- Clients are rated whenever they return a book with a new defect (e.g., a page is ripped out). Clients with a bad rating are not entitled to any discounts, even if they are premium members.

- After receiving three bad ratings, the client is not allowed to borrow new books anymore.

- The customer hopes to get a new client base among students. Hence every student receives an extra $5\,\%$ discount with the first five orders.

- Every book has a base price assigned, but this price may change (e.g., due to demands, promotional campaigns, etc.).

Here are two examples of a client dealing with the company:

- Charlotte is a student who owns 38 of the books possessed by the company. So far she does not have any bad ratings. She wants to borrow "Das Kapital" as her fifth book. Altogether, she should get a $3\,\%$ discount plus another $5\,\%$ discount on top.

- After one month, Charlotte has added another 15 books to the company's possession. Now she wants to borrow the book "Animal farm." This time she should get a $5\,\%$ discount.

Formalise the informal requirements above on borrowing refusal and discount computation using a decision table (in the standard (i.e., not in the collecting) semantics).

(i) Create conditions and actions as necessary and make appropriate use of environment assumptions and conflict axioms. Specify the rules necessary to tell if a transaction must be refused, and, if not, to indicate the discount factor.   (5)

(ii) Consider again the example with Charlotte from above. After another month, Charlotte has received two bad ratings. Now she wants to borrow "The mythical man-month."

What is the output according to your decision table in this case?   (1)