Exercise for the Lecture Requirements Engineering (Elite)
Summer Semester 2022
Prof. Dr. Alexander Pretschner, Markus Schnappinger, Lena Gregor
Tutorial on June 28th, 2022

Technische Universität München
Fakultät für Informatik

Tutorial 7: Functional Requirements

This exercise sheet covers the contents of the **7th lecture**.

Exercise 1 Performance, Security, and Safety: non-functional vs functional? (Discussion)

Consider the following statement:

Temporal behaviour and performance are non-functional properties.
Safety and security are functional properties.

Do you agree with the statements or do you disagree? Support your opinion by three arguments, including examples, that either refute or support the thesis.

Exercise 2 Cockburn's Use Case Template (Modelling)

Your boss wants you to write down some *use cases* for the procedure of reserving an e-scooter with the app. In our case an user can reserve any scooter for free for up to 15 minutes. Luckily, you heard about *Cockburn's use case template* in the Requirements Engineering lecture and recall that it is a good method for eliciting and breaking down use cases.

a) Start with the template given below: Formulate the goal of the use case as a short active verb phrase.

b) Formulate the "goal in context", "scope", "level", "preconditions", "success end condition", "failed end condition", "primary actor", and "the triggering event".

c) Come up with a meaningful *Description* of the main success scenario of at least 6 steps, some *Extensions* and one *Sub-Variation*.

d) Write down the "priority", the "performance target", the "frequency" and one "open issue".

e) Now transfer the documentation from the template into a corresponding model. To do this, create a UML activity diagram that describes the set of scenarios specified in the use case.

You can use the following table template for writing down the use case:

USE CASE 1	
Goal in Context	
Scope	
Level	
Preconditions	
Success End Condition	
Failed End Condition	

Primary Actor	
The Triggering Event	
MAIN SUCCESS SCENARIO	
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
EXTENSIONS	
2.	
7a.	
7a1.	
7a2.	
9a.	
9a1.	
10a.	
10a1.	
10a2.	
SUB-VARIATIONS	
1.	
7.	
Priority	
Performance Target	
Frequency	
Open Issues	

Exercise 3 Function Hierarchies (Modeling)

As you already know from the lecture, function hierarchies are used to structure the functionality of the system interface considered in the RE-task.

1. Create a function hierarchy by extending or refining the individual system functionalities from the use case *withdraw money* of an ATM system. Relate the elaborated collection of functions to each other using decomposition and dependency.
2. Specify input and output channels for your leaf-functions.
3. Identify additional relationships between the functions, e.g., *"excludes/XOR", "interrupts", "as alternative to", "precedes", "enables/disables", "follows", "data (flow) dependency"*.
4. Provide an important reason for creating and using a function hierarchy. How is a function hierarchy different from the functions in the use case?

Exercise 4 System models, artefact orientation, and their problems (Understanding, Discussion)

One major challenge for model-based software engineering or requirements engineering is the synchronization of various models. Name and describe two ways to achieve consistency among all used models!

Remember our discussions about artefact-orientation and activity-orientation. Discuss reasons why consistency management may or may not be a problem when using frameworks like AMDIRE.

Exercise 5 Executable Models (Discussion)

Discuss the following hypothesis:

The goal of requirements engineering should always be to specify business processes using executable languages (e.g. BPEL). These allow their immediate execution by means of workflow engines.