Prof. Dr. Peter Thiemann
Sergio Feo-Arenis
Sergiy Bogomolov

Summer Term 2014

Software Engineering

http://proglang.informatik.uni-freiburg.de/teaching/swt/2014/

Exercise Sheet 1

Use Cases and User Stories

We have extracted use cases from various development projects. Read them carefully and discuss the following:

- Does the information provided correspond with what is expected for a use case?
- What quality criteria are not satisfied according to the guidelines given in the lecture?

Please also have a look at the user stories we provide and check whether they are acceptable according to the lecture. If possible, provide an improved version.

1.1. Fire alarm system A company wishes to develop a wireless fire alarm system, controlled by a central unit which can connected to the internet. Systems have different component types: repeaters (for range extension) and sensors. The central unit can collect information about the system's components and store it on a cloud service.

Use Case:	Operating parameters on the server
Goal:	Storage of updated system's operating parameters on the server
Category:	Data transfer
Precondition:	Current operating parameters are available on the central unit
Postcondition/success:	The operating parameters stored on the server are consistent with the current operating parameters of the system's components.
Postcondition/failure:	The operating parameters stored on the server are not consistent with the current operating parameters of the system's components.
Actors:	A fire alarm system and a server
Trigger:	Time-triggered or by user interaction
Description:	A system component sends over cascaded repeaters its operating parameters to the central unit, where they are stored as `int` variables. The central unit sends the data periodically to the server. The server stores the received data.
Extensions:	None
Alternatives:	None

User stories:

1. As a user I want to read off the battery status of the sensors on the PC.

2. As a user I want to have 24h support from the manufacturer.

Solution

1. Use case name is not very meaningful. A proper name would be "Parameter transmission to the server".
2. Category should be either primary, secondary or optional.
3. The term "operating parameters" is not defined for pre-/postconditions. The use case should explicitly refer to the battery voltage, signal strength, etc.
4. The post condition is too permissive. A better option would be to state that only the values of the parameters which were transferred at the last attempt might be inconsistent. The rest of the parameters stored on the server should not change their values.
5. There is no trigger "user interaction" in the description.
6. On the use case level we should treat the system as a black-box. In particular, we should avoid specifying how the parameters are transmitted and furthermore stored internally (do not mention the `int` type).
7. The first user story is ambiguous because it is not clear whether we refer to "sensors on the PC" or "system sensors" (whereas the data is then forwarded to the PC).
8. The second user story does not refer to the system.

1.2. Game Development A game development company is working on a new real-time strategy title. We have extracted two cases from their requirements document.

Use Case: Building construction
Goal: Construct a building
Category: Game mechanics
Precondition: At least one villager is selected and enough resources for the selected building are available.
Postcondition/success: The building construction starts.
Postcondition/failure:

- The building is not built.
- The game world is not changed.
- The resources are not changed.

Actors: User
Trigger: User selects the Build command
Description:

1. The user clicks the item Construct in action menu of a villager.
2. The user selects with the mouse a building to be built.
3. The building becomes schematically visible (with a green background) and can be placed somewhere in the game world.
4. The user starts the building construction at the chosen location by the selected villager by clicking the left mouse button.
5. The construction finishes immediately.

Extensions: The user can also construct building extensions.
Alternatives: The user selects a building to be built by using a keyboard shortcut.

Solution

1. Category should be either primary, secondary or optional.
2. The concepts Build and Construct refer to the same action. Therefore, the action name should be unified.
3. The construction of building extenstions should be described in a separate use case. In the extensions we can describe how the main success scenario reacts when something goes in the way different to the main scenario (`http://blog.casecomplete.com/post/Writing-Use-Case-Extensions`). For the building extension we need, e.g., to modify the precondition (main building already constructed).
4. The current alternative is actually an extension.
5. Item 5 of description contradicts the provided postcondition.

Use Case:	Move a figure by clicking
Goal:	Game figures should be able to move
Category:	Game Interface
Precondition:	The game character must be able to move, i.e. it must no be enchanted by an immobilizing spell
Postcondition/success:	The character is on the target point OR the character is on a walkable point in the world that is as close as possible to the target point
Postcondition/failure:	
Actors:	Game figure, user
Trigger:	Click
Description:	

1. The user clicks using the left mouse button on a point in the world
2. The character moves from its current position towards the target point
3. Behavior during movement:
   - If the target point is not reachable, the character tries to reach the closest walkable to the target point
   - The character walks around obstacles in due time

Extensions:	None
Alternatives:	None

Solution

1. Category should be either primary, secondary or optional.
2. The precondition should refer to the selected game figure.
3. The trigger should be defined, e.g., as follows: The user clicks on some location in the game world.

User stories:

1. As a user I can construct buildings.
2. As a user I can enter a game menu any time in order to pause the game.
3. As a user I can move the units.
4. Units can be trained.
5. In multi-player mode no network packages should be lost.
6. As a AI-player in the multi-player mode I should first train a worker in order to explore the game world.
7. As a user, I can lose.

Solution

1. In the second story we could replace "in order to" with "so that" to match the mask provided in the lecture. In particular, you can use the following options: "As a <role>, I want <goal/desire> so that <benefit>", "As a <role>, I want <goal/desire>", "In order to <receive benefit> as a <role>, I want <goal/desire>" and "As <who><when><where>, I <what> because <why>".
2. The fourth user story is wrong as it does not identify the actor who takes the action. The right version would be: "As as a user I can train units".
3. In the fifth user story the abstraction levels of "multi-player mode" and "network packages" do not match. In particular, the network communication level is too concrete.
4. In story 6, no interaction of the user with the system is described as AI-player is not an actor. We cannot provide a user story in this case as they are limited to the interaction of the user with the system and not purely between system objects.
5. Story 7 is too unprecise as it does not provide any additional context, e.g., against which player, etc.

Submission

- Submit this sheet *before* the lecture of Thursdays.
- Late submissions will not be accepted.
- Deadline: Thursday 11:59 a.m..