Software Engineering – Use case scenarios

Adam Krechowicz

1 Use case scenarios

Use cases that was identified on previous laboratory means what functionalities will be executed int he system. However, it does not indicate how they will be executed.

To determine the logic that needs to be executed during each use case we use *Use Case Scenarios*. This is a very formal way of use case description (step by step) how this functionality will be performed. It is common that one use case can be executed in many was. We need to think about the cases when the use case is executed correctly as well as when it is executed with some problems. To create possible conditions we need to think from system logic perspective. For example for "Borrow Book" we need to think about the case that Reader exceed the number of books that he can borrow, he does not give back previous book in time, there is no that book in the library etc. At this level we do not need to think about some technical issues like system failures. Because of that, we do not need to think about things like: there is no communication with the server, the database is not working etc.

2 Structure of the use case scenarios

Use case scenarios should be created with use of the strict structure. This structure contains following elements:

- Use case name;
- Preconditions – conditions that needs to be satisfied in the system (system state) to start execution of this use case
- Postconditions – conditions that will be satisfied in the system (system state) after execution of this use case. Usually we are interested in a state after the correct execution of use case.
- Actors – we need to describe all actors that can execute this use case
- Main flow – description (step by step) of the stages that will be performed to execute this use case

- Alternate flows – descriptions of the alternate stages that can be executed and are different from the main flow.

It is important to know that Main flow should represent the most optimistic situation. We need to assume that everything will be executed without any problems. Because of that it does not contains any conditions. Main flow should consists of ordered items that are executed sequentially. First item typically represents some action that is performed by the actor. Following items typically are executed alternately by the system and actor.

All the situations that can be different from the main flow are described in the alternate flows. We need to consider all possible alternate situations. Alternate flow typically have the following structure:

- Reference to the item in the main flow
- Condition that is satisfied
- Reaction to this condition

Reaction is typically a GOTO to other item in the main flow or it can be END EXECUTION of this use case.

3 Example: Borrow book

- Name: Borrow book
- Preconditions:
  - Reader needs to be logged in the system
  - There must be at least one book in the library
- Postconditions:
  - Book state is changed to "borrowed"
  - Book is assigned to Reader
- Actors: Reader
- Main flow:
  1. Reader picks option Find Book
  2. System shows finding form
  3. Reader fills the finding form
  4. System shows list of books that match finding query
  5. Reader picks book from the list
  6. System shows detailed information of the book

7. Reader picks option borrow book
8. System checks the availability of the book
9. System checks if Reader can borrow
10. System marks the book as ”borrowed”
11. System assigns the book to the reader
12. System creates the date of final giving back
13. System shows the confirmation

- Alternate flows
  - 4. There is no book that match finding query. GOTO 2.
  - 8. Book is not available. GOTO 6.
  - 9. Reader reached limit of borrowed books. POWRÓT DO 6.
  - 9. Reader does not give back book in time. POWRÓT DO 6.

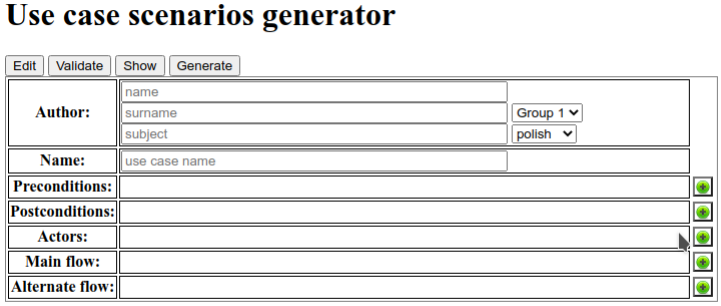
4 Tasks to complete

1. Create the use case scenarios (results put in element <article id=”use-case-scenarios-english”>)

Use case scenarios should be created using prepared generator that is available on http://kronos.tu.kielce.pl/scenarios.html.

5 Using generator

Generator allows to create use case scenarios in a structural form. Below is the look of the generator:



The form needs to be filled with informations such as Name, Surname, Group, System subject and the language of scenarios. In case of the preconditions, postconditions, actors, and flows the items should be added by using button.

Created item can be deleted with button. Button and allows to change order of items.

Validate button allows to check for scenarios completeness and errors. The *Show* button allows to view final scenarios. The *Generate* button allows to generate the content of the scenarios that needs to be included in the report. The *Edit* button allows to continue editing.

After creation of the scenario its completeness should be checked by *Validate*. In case of some errors it have to be corrected. After that no items in the validation sections should be visible. After validation the *Generate* button allows to generate html code that needs to be copied to final report.