Lecture 5: Software requirements modelling (dynamic)

- Scenario model
- Details of scenario modelling
- Scenarios and activities
- Scenarios mapped from use cases
- Scenarios and storyboards

What do we want to do?

We want to describe the details of the use cases – system’s detailed dynamics (behaviour).

How to do it?

- write a “story” on how the system should behave (exchange information) in respect with the external objects (actors – users, other systems).
- give the story a description of the user interface (if necessary)

How do we write stories?
**Writing novels: story and the environment**

1. Lord Mark runs to aunt Marta’s house.
   - a. He talks with aunt Martha.
   - b. He talks with aunt Martha.
2. Lord Mark goes to the old hut.
3. Aunt Martha goes to Tom “the wise”.
4. Tom rides to the old hut.

**Writing (a good) film scenario**

Good scenarios are usually a good balance between the story (account of incidents) and the description of environment (people, interiors, nature, ...).
**Problems with stories for software**

System’s functionality described like a novel – with descriptions of the environment buried inside a story.

Many inconsistencies – elements of the environment with differing descriptions (synonyms, homonyms).

Problems with translating the system’s description into design and code.

SOLUTION: propose a format for stories that would allow for coherence and precision and enable easier transformation into design and code.

---

**What is a scenario?**

Sequence of actions – a dialog between objects outside a system and that system.

Performed on behalf of a primary object (“primary actor”) that triggers the scenario.

The initial trigger is followed by a sequence of actions performed by the system and by the outside objects (including the primary object).

The sequence is controlled by the system and leads to a single goal of significant value to the primary object.

The sequence can fail to reach the goal – it is then a failure scenario.
Good style for writing scenarios

Simplest possible sentences!

1. **Admin** enters user data
2. **Teacher** accepts the current marks

Subject  
Verb  
Objects (1 or 2)

**System** adds the user to the user list

Scenarios as sequences of SVO sentences

**Add user**

1. **Admin** wants to add a user to the user list
2. **System** shows user data dialog
3. **Admin** enters user data
4. **System** validates user data [user data valid]
5. **System** adds the user to the user list
6. **System** shows error dialog
Use cases – collections of stories

Use case is a collection of scenarios grouped by the same triggering action and leading to the same goal for the primary actor. Reminder: Use cases can be related by “invocation” relationships.

Use case as an activity
Inserting vs. requesting use cases

The basic idea

Separate the story from the description of the problem domain.
Extending the story with domain description

The SVO-based stories lack descriptions of notions used within them.

But somewhere else, in a distant story…

Separating the notions from the story

Keep the vocabulary separate, but coherent with the stories.
Building the noun vocabulary

Vocabulary is a map of “user’s territory”. Better to make this map graphical.

1. Dean wants to add new lecture to course
2. System asks for semester
3. Dean enters the semester
4. System asks for data of the lecture
5. Dean enters the data of the lecture
6. System adds the lecture to the list of lectures
7. System assigns the lecture to the teacher

Adding verbs to the nouns

Verbs in SVO sentences can be attached to nouns that are derived from sentence objects.

1. Dean wants to add new lecture to course
2. System asks for semester
3. Dean enters the semester
4. System asks for data of the lecture
5. Dean enters the data of the lecture
6. System adds the lecture to the list of lectures
7. System assigns the lecture to the teacher
Adding storyboarding information to scenarios

When discussing scenario details it is of great benefit to determine user interface details.

A scenario with added information about the user interface is called a storyboard.

Creating a storyboard might change the scenarios slightly.

1. **Admin** chooses the add user option from the main menu
2. **System** shows user data dialog
3. **Admin** enters user data
4. **System** validates user data [user data valid]
5. **System** adds the user to the user list
6. **System** shows add user success message

Summary – how will WE do it?

Every use case from the User Requirements Model should have several scenarios.

- main course scenario (most common) – ends with a success
- alternative course scenarios – end with a failure or with a success

Scenarios should be written in clear language.

- SVO(O) sentences
- subjects found in the actor vocabulary
- objects found in the domain, configuration and UI window vocabularies
- verbs found in the operation lists of sentence objects

Scenarios should be extended into storyboards.