Use Cases (EE5525)

Prof Peter R Hobson

What is a Use Case?

- A contract between the project *stakeholders* about the behaviour of the system.
- Involves *actors* who play a role external to the system.
- Actors have goals which they wish to fulfil by interacting with the system.
- Use cases exist primarily as *prose text*.

What are they used for?

- They describe the externally visible behaviour of the system under discussion (SuD)
- They describe how the SuD responds to a request from a stakeholder to fulfil a goal.

Therefore

They are very useful as a means of finding out what goals are to be achieved.

A way of getting a view of what type of messages and responses might be involved.

They are an excellent way to find out what is really required from a system.

HOWEVER – they are not *all* of the requirements (but an important fraction nevertheless)

Jargon 1: Stakeholders

- A Stakeholder has a vested interest in the behaviour of a system.
- Some stakeholders directly interact with the system (*actors*).
- **But** some stakeholders may not (e.g. the owner, a regulatory body, the state)
- Use cases protect the interests of the stakeholders.

Jargon 2: Actor

- An actor is some *thing* outside the system boundary that interacts with the system to achieve a goal.
  - Often the actor is also the *trigger* of the use case.
- An actor plays a role.
  - Many different roles may be played by the same entity.
  - Many distinct entities may map onto a single actor.
- An actor (role player) may be
  - A human being
  - A computer
  - Another system

An example – Withdraw cash from an ATM

- Actor = Customer (Account Holder)

1. Customer puts card into system
2. System reads card and validates customer
3. Customer enters PIN
4. System validates PIN
5. Customer selects amount of cash to withdraw
6. System notifies bank and receives new balance
    1. System returns card to customer
    2. System delivers cash to customer
    3. System logs the transaction

The example deconstructed

Withdraw cash from an ATM

- Actor has a clear goal (to get some cash)
- The Use Case is essentially a dialogue
- If capturing requirements for a new system then treat the use case as a black-box (*what* not *how*)
- Avoid a UI centric approach (see bullet above)
- Don’t go into micro details, capture the essentials.
- Look to the expected, successful outcome *first*

Some useful resources

These are adapted from the book *Writing Effective Use Cases*, Alistair Cockburn, Addison-Wesley, 2001

Also see his web page at http://alistair.cockburn.us/index.php/Resources_for_writing_use_cases

Especially this resource

http://alistair.cockburn.us/index.php/Use_cases_in_theory_and_practice_180

Reminders

- A Use Case is an essay – keep your eyes on the text and *not* on the UML diagrams for Use Cases.
- Make the Use Case easy to read.
  - Use active verbs in short direct sentences.
  - Say what is really a requirement and eliminate things which are not.
  - Use names for use cases which are verb phrases
  - Start from the trigger and continue until the successful end (i.e. the goal is achieved).

Use this form of sentence

- Present tense
- Active verbs
- Describe a goal being achieved that moves the whole process forward.

Some examples

Customer enters credit-card and PIN

System validates customer

Customer specifies shipping destination

From what point of view?

- Don’t write from the inside of the SuD looking outwards
- Write from the point of view of a bird flying overhead looking down.
- Perhaps the analogy of reporting on a match (rugby/soccer for e.g.) between two teams will help you.

Keep the User Interface (UI) out!

Makes text longer and more complicated.

Danger of tying you down to a particular implementation – remember this is the requirements stage of the project.

Remember the Stakeholders

- They need guarantees
- System enforces a contractual agreement between stakeholders. The Use Case is written from the user perspective and describes how this contract will be fulfilled.

Preconditions

A precondition *must* be true *before* a Use Case can be executed.

A pre-condition implies the **successful** outcome (*post-condition)* of another Use Case (so look for it/write it).

Remember this if nothing else!

An actor is outside the system boundary