Requirements w/Rapid Prototyping—Use case Analysis

1. What are Use cases?
2. A Use case Example
3. Where are they used?
4. Use case FAQ.
5. Relationship to next lecture.

Definition of Use cases

"A Use case is a collection of possible sequences of *interactions* between the system under discussion and its external actors, related to a particular goal"

—*Alistair Cockburn/Ivar Jacobson*

Important:

- **Interaction**: Connects one actor's goal to the system's or another actors' responsibilities.
- **Scope**: External, or Internal
- **Hierarchical**: Each interaction can summarize a lower level Use case.
- **Complete**: They go on until success or abandonment.

An Example: Airline Ticket Purchase

Goal	Buy a ticket
Scope	External, UI
Pre-Conditions	UAL Res System Up, Customer has acct.
Success State	Purchased valid tkt, +1 reservation on flight.
Failed State	No Effect
Primary Actor	Customer
Secondary Actor	None
Start Action	Customer login to United Connections
Description	1. Customer Logs into Website. 2. Customer Enters dates & travel endpoints 3. System checks availability & price 4. System shows availaibility to customer 5. Customer selects flights 6. System reserves seats 7. System presents charge form to customer 8. Customer enters charge information 9. System charges the cost of ticket 9. System sends receipt to customer 10. Customer logs out
Variations	1. No seats/flights available 2. Customer rejects flights/prices 3. Charge card invalid.
Extensions	1. System may offer free upgrades to FF 2. Payment may be with FF miles

Explanation of Use case Template

Goal	The Primary Actor's Goal
Scope	External to System, or Internal
Pre-Conditions	When this Use case is applicable
Success State	The state of the world when Use case ends.
Failed State	. . . and if it fails.
Primary Actor	Whose goal is of concern?
Secondary Actor	Who might help the system?
Start Action	the trigger action.
Description	1. Interactions.. 2 . . .
Variations	1. Exceptional conditions 2. . . .
Extensions	1. Additions to basic function 2. . . .

Using Use cases

- **In Requirements:**
  1. Help identify exceptional conditions
  2. Help locate extesions.
  3. Can be used in other Use cases (modular reuse) or extended (inheritance type reuse).

- **In addition to requirements**:
  1. To identify common functionality for software design.
     – Use cases are the key step in OO Analysis methods.
  2. For validating software designs.
  3. For testing & validation implementations.
  4. For creating manuals.
  5. For designing UI interfaces

Use case FAQ

1. How many Use cases?
   - One for each main goal of each type of user.
2. How many variations?
   - One for each "deviation" that affects user's goal in the particular Use case.
3. What should NOT be in a Use case
   - Internal system details that don't directly affect the user's goals.
   - Any event/interaction that doesn't relate to a specific goal.
   - User interface design: doesn't matter if buttons, menus, etc.
4. When is a Use case complete?
   - When goal has been acheived, or there is a failure.
5. How to use it with Rabid Prototyping?
   - Implement each Use case so that variations & extensions can be handled easily later.

Ask these questions

1. Is the Use case complete? Are there any details that need to be added? (Actors, Goals, Success state, Failure State, Variation, Extension?)

2. Do I feel confident that the actor's goal is going to be properly met?

3. Can the Use case be simplified? By changing terminology/function etc?

4. Are there any additional Actors that should participate in this Use case?

5. Are there any additional Goals of the Actor that are not addressed? (*)

6. Have I considered all the Actors? (*)

(*) Questions applicable to entire set of Use cases.

Thinking about Requirements Methods

- What are the common goals of requirements methods (the A7E method, and the Use case method?)
- How are they different?
- Applicability (review).
- Is this an exact science? (Can we prove these methods work?)
- The big players: Michael Jackson, Pamela Zave, Ivar Jacobson, David Parnas, Sol Greenspan, Alex Borgida....

Next Lecture–User Interface Design

How to design UI to make system functional, easy to learn, and easy to use.

Set of about a dozen design principles.

Think about applying them in the context of each Use case.

Assignment 2: Should be documented in terms of Use cases. Make it as set of web pages for demonstrating it at interviews...