

Software Engineering I CSC-382



Lecture 5

Software Engineering I CS-382

- Lecture 5
- What we will cover: (Requirements Engineering)
 - Chapter 8, Section 8.5 (Use Cases) in Pressman
 - Goal of the next 2 lectures is to understand the critical tasks associated with deriving the software requirements of a system

Recall Some Potential Tools for Eliciting Requirements II

- Object Oriented Analysis-based Techniques
 - Scenario-based with **use cases** to define key actors of system
 - Actors then help to define the classes
 - Things they do specify the functional requirements
 - Can also use **Swim Lanes** and **Activation Diagrams** to further clarify the use cases.

3

Use-Case Based Elicitation

“[Use-cases] are simply an aid to defining what exists outside the system (actors) and what should be performed by the system (use-cases).” Ivar Jacobson

- **Actors** represent roles people or devices play as the system functions
- Actors have two common features:
 1. External to the application
 2. They take initiative, stimulate and interact with our system

4

Use-Case Based Elicitation II

- Three types of actors:
 1. Users
 2. Administrators
 3. External Programs and Devices

5

Other Details of Use Cases

- Use Cases are one means for describing scenarios of operation of a system
- A scenario that describes a “thread of usage” for a system
 - It describes the specific path that a user will take to perform a given task
 - There can be multiple scenarios running in a single Use Case

6

Other Details of Use Cases II

- Actors can play a number of different roles for a given scenario
- Two more common formats for Use Cases
 1. Use Case **Diagram**
 2. Use Case **Text** Template (most useful method)

7

Use-Cases Answer the Following:

- Who is the primary actor, the secondary actor (s)?
- What are the actor's goals?
- What preconditions should exist before the story begins?
- What main tasks or functions are performed by the actor?
- What extensions might be considered as the story is described?
- What variations in the actor's interaction are possible?

8

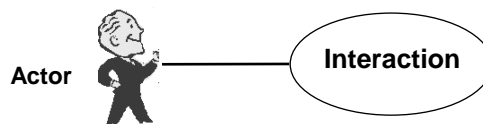
Use-Cases Answer the Following: II

- What system information will the actor acquire, produce, or change?
- Will the actor have to inform the system about changes in the external environment?
- What information does the actor desire from the system?
- Does the actor wish to be informed about unexpected changes?

9

Format of the Use-Case

- Basic format of the Use Case Diagram:



- The Use Case diagram defines the Cases to define in more detail:

- Define using the Use Case descriptions:

| | |
|----------------|----------------|
| Use Case Name: | Primary Actor: |
| Data: | |
| Stimulus: | |
| Response: | |
| Comments: | |

10

Recipe for Developing Use-Cases

1. Identify how is going to be using the system
2. Choose one of those actors
3. Define what that actor wants to do with the system
 - Each thing they want to do becomes a Use Case
4. For each Use Case determine the most usual course
 - i.e. what happens normally
5. Describe that basic course, in prose as:
 - “actor does something, system does something, actor does something else ...”

11

Recipe for Developing Use-Cases II

6. Now consider alternate paths and add those as **Extending** Use Cases
 - Name these by the condition under which they occur
 - E.g. “student lacks prerequisites”
 - These are never part of the basic course of operation
7. Review all the Use Case descriptions against each other and look for **Commonality**
 - These common elements can become “**uses**” (or include) Use Cases
 - These are always part of the basic course of action
8. Repeat Steps 2-7 for every actor

12

More Guidelines to Writing Use Cases

1. Use Simple Grammar
 - Subject...verb...direct object...prepositional phrase
 - E.g. “the system deducts amount from the user account balance
2. Show Clearly ‘Who Has the Ball’
 - Picture a ping-pong game with the action going back and forth
 - The ball is the message or data that is being passed around the system

Cockburn 13

More Guidelines to Writing Use Cases II

3. Write from a ‘Bird’s Eye View’
 - Rather than write get ATM card and Pin...
 - The customer inserts the ATM card and enters PIN
4. Show the Process Moving Forward
 - Amount of progress relates to how abstract we are making the Use Case
 - Initially in the design we should stay at a high level to keep the length of the Use Case manageable (I like no more than 10 steps)

Cockburn 14

More Guidelines to Writing Use Cases III

5. Show the Actors Intent not the Movements
 - Do not describe the interface details
 - Stay at the level of stating what data to enter, rather than specifics such as “user hits enter”..

More Guidelines to Writing Use Cases IV

6. Include a ‘Reasonable’ Set of Actions
 - Jacobson views each step Use Cases as a Transaction, which entails:
 1. Actor sends Request
 2. System validates Request
 3. System alters its internal state
 4. System responds to actor with result
 - This is a lot to put on 1 line (hard to read), but remembering these sub-steps allows us to write better Use Cases (we won’t miss steps)

More Guidelines to Writing Use Cases V

7. “Validate” inputs do no “Check” them
 - This is a little subtle but important
 - If you have steps in the Use Case that are if...then sequences then you probably have a check statement
 - “check” means you have to follow it with the cases
 - Validate subsumes the operations of the if..then within it
 - Makes the Use Case less Procedural and also sticks to the important flow of the process

Cockburn 17

More Guidelines to Writing Use Cases VI

8. It is Optional to Mention Timing
 - Timing is implicit in the flow of the Use Case
 - Usually there is no need to specifically mention it
9. Use Idiom “ User Has System A Kick System B”
 - Useful in cases where the system is an intermediary to some other system (e.g. web pages going to databases, etc.)
 - State:
 - User has the system fetch the user profile data from the Accounts DB

Cockburn 18

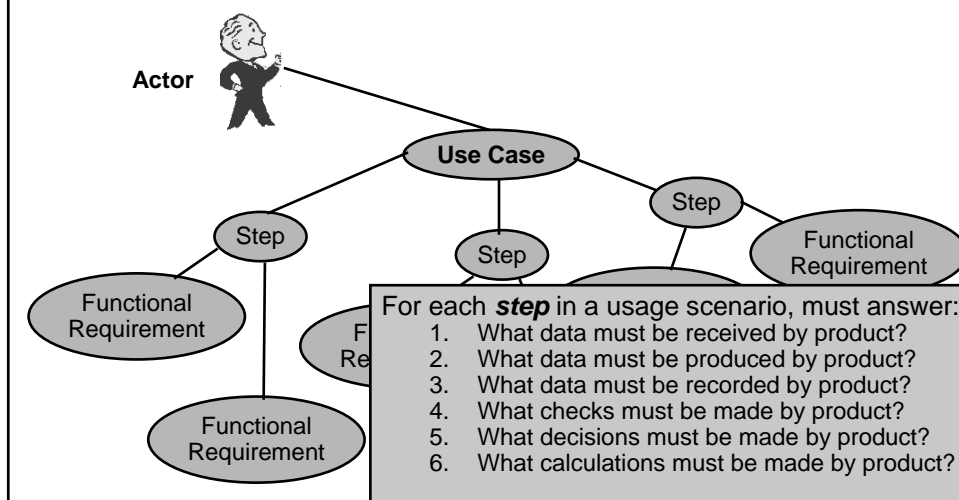
More Guidelines to Writing Use Cases VII

10. Use Idiom: Do Steps X-Y Until Some Condition
 - Can have line in Use Case: “User repeats steps 2-6 until she indicates she is done”

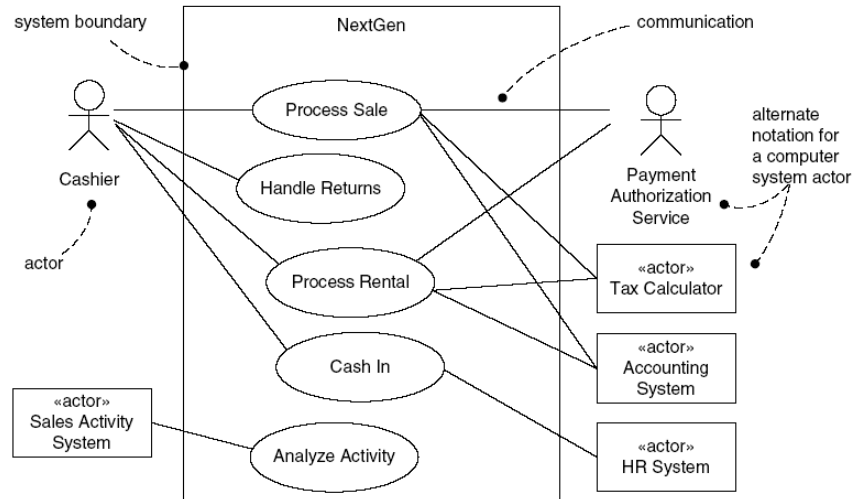
Cockburn

19

How to Use “Use Cases” for Requirements Elicitation



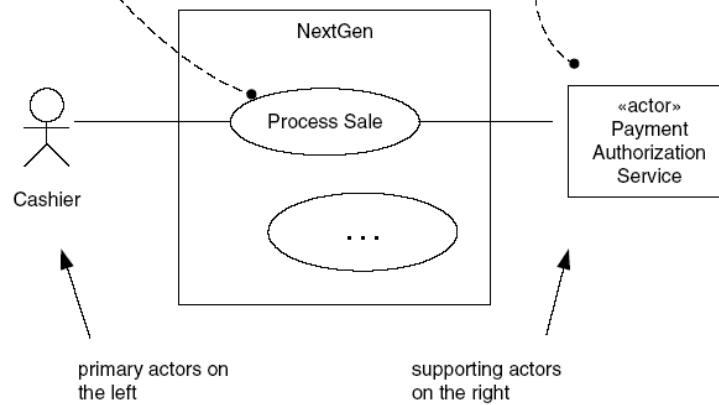
Example System Use Case Diagram



Helpful Use Case Formatting Hints

For a use case context diagram, limit the use cases to user-goal level use cases.

Show computer system actors with an alternate notation to human actors.



In-class Requirements Engineering Exercise: SafeHome



23

Recall: SafeHome from Users Perspective



How a Project Starts

The scene: Meeting room at CPI Corporation, a (fictional) company that makes consumer products for home and commercial use.

The players: Mal Golden, senior manager, product development; Lisa Perez, marketing manager; Lee Warren, engineering manager; Joe Camalleri, executive VP, business development.

The conversation:

Joe: Okay, Lee, what's this I hear about your folks developing a what? A generic universal wireless box?

Lee: It's pretty cool, about the size of a small matchbook. We can attach it to sensors of all kinds, a digital camera, just about anything. Using the 802.11b wireless protocol. It allows us to access the device's output without wires. We think it'll lead to a whole new generation of products.

Joe: You agree, Mal?

Mal: I do. In fact, with sales as flat as they've been this year, we need something new. Lisa and I have been doing a little market research, and we think we've got a line of products that could be big.

Joe: How big. . . , bottom-line big?

Mal: (avoiding a direct commitment): Tell him about our idea, Lisa.

Lisa: It's a whole new generation of what we call "home management products." We call 'em *SafeHome*. They use the new wireless interface, provide

homeowners or small business people with a system that's controlled by their PC—home security, home surveillance, appliance and device control. You know, turn down the home air conditioner while you're driving home, that sort of thing.

Lee: (jumping in) Engineering's done a technical feasibility study of this idea, Joe. It's doable at low manufacturing cost. Most hardware is off the shelf. Software is an issue, but it's nothing that we can't do.

Joe: Interesting. Now, I asked about the bottom line.

Mal: PCs have penetrated 60 percent of all households in the USA. If we could price this thing right, it could be a killer-App. Nobody else has our wireless box—it's proprietary. We'll have a two-year jump on the competition. Revenue? Maybe as much as \$30–40 million in the second year.

Joe (smiling): Let's take this to the next level. I'm interested.

24

Can we Find Some Actors in This ?

Recall only going to focus on home security first

Home security functions:

- Standard window/door/motion sensor **monitoring** for unauthorized access (break-ins).
- **Monitoring** for fire, smoke, and CO levels.
- **Monitoring** for water levels in basement (e.g., flood or broken water heater).
- **Monitoring** for outside movement.
- **Change security setting** via the Internet.

25

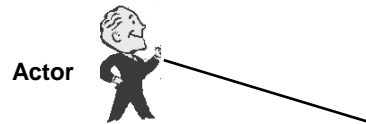
Can we Find Some Actors in This ?

Home surveillance functions:

- Connect to one or more video cameras placed inside/outside house.
- Control pan/zoom for cameras.
- **Define camera monitoring zones.**
- **Display camera views** on PC.
- **Access camera views** via the Internet.
- Selectively record camera output digitally.
- **Replay camera output.**

26

In-class “Use Cases” Diagrams



27

For Next Class

- Continue to Study Chapter 7 Sections 7.6-7.9 in Pressman (Requirements Engineering)
- Text-based Use Cases

28