

INF 117

Project in Software Engineering

Lecture Notes ~Spring Quarter, 2008

Michele Rousseau
Set 3 - Requirements

What's Next

APRIL 2008

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
6 Week 2	7 Team Website	8	9	10 Req Iter: #1 Project Plan #1	11	12
13 Week 3	14	15 Taxes Due	16 Stud. Pres-Req Order 1,2,3 Team App: #1 Peer Eval: #1	17 Req Iter: #2 Test Plan It #1	18 Team Log #1	19
20 Week 4	21	22 Earth Day	23	24 Req. Iter. #3 Des. Iter: #1 Project Plan #2	25 Cust. Milestone Req. Approved	26
27 Week 5	28	29	30 Stud. Pres-Des Order 2,3,1 Team App: #2 Peer Eval #2	1 Des. Iter: #2 Test-Plan It #2 (Incl Des)	2 Team Log #2 Course Log #1	

Announcements

Let me know this week about Showcase event

- June 12th, 3:30p-6:30p
- Note: Time has been extended

DUE this week:

- Today: Website
- Thursday: Requirements Iteration #1
- Project Plan #1

Set 3 3

Today's Class

Requirements → UML

- Scenarios
- Use-Case Diagrams (extended)
- Class Diagrams
- ... any other diagrams to communicate

Requirements

- How do you know when you are done
- Mis-use Cases
- Don't forget things like non-functional requirements
- Glossary

Project Plans

Set 3 4

What is UML?

Unified Modeling Language (UML)

Let's break it down:

Unified

In 1994,

- Two important methodologists Rumbaugh and Booch decided to unify their approaches in 1994

In 1995, another methodologist, Jacobson, joined the team

- His work focused on use cases

In 1997,

- the Object Management Group (OMG) started to standardize UML

Set 3 5

Models

Models are *abstract representations*

Contain essential characteristics and omit non-essential details

Models can be representations of the world

- Domain models
- Requirements

Models can be representations of software

- Specifications
- Design
- Systems

Set 3 6

Why make models?

- ⌘ Systems are complex and hard to understand
- ⌘ Clarify certain aspects (increase visibility)
- ⌘ What can you do with models?
 - Express your ideas and *communicate* with others
 - *Reason* about the system
 - ▣ detect errors
 - ▣ predict qualities
 - *Generate parts of the real system*
 - ▣ Code
 - ▣ Schemas

Can *reverse engineer* a system to make a model

Architecture not usually a good example...

...BUT

Imagine the building the Biltmore

- ⌘ 175,000 Sq. Ft. (that's 4 acres)
- ⌘ 250 rooms
- ⌘ 35 bd/43 ba.
- ⌘ 65 fireplaces
- ⌘ 125,000 acre lot (that's 195 sq mi)

Where would you begin?



- ⌘ A model helps reduce complexity
 - Eliminates details that are not necessary at the time
 - Allows you to divide and conquer large tasks

What constitutes a good model?

A model should...

- ⌘ *Abstract* away unnecessary details
- ⌘ Provide a means to *reason* about the system
- ⌘ Use a standard notation
- ⌘ Be *understandable* by clients and users
- ⌘ Lead software engineers to have insights about the system
- ⌘ Be a tool for *communication*

UML is a ...

- ⌘ Semi-formal graphical (visual) modeling language
- ⌘ Object Modeling Language (OML)
- ⌘ A way to *communicate* details...
- ⌘ UML is *descriptive* → tries not to be prescriptive

... essentially
it is a set of diagrams used to model the system

Types of UML Diagrams

<u>Structure</u>	<u>Behavior</u>
(6 types)	(4 types)
⌘ Class diagrams	⌘ Activity diagram
⌘ Object diagram	⌘ Use Case diagram
⌘ Package diagram	⌘ State machine diagram
⌘ Composite structure diagram	⌘ Interaction diagrams <ul style="list-style-type: none"> ● Sequence diagram ● Communication diagram ● Interaction overview diagram ● Timing diagram
⌘ Component diagram	
⌘ Deployment Diagram	

If the appropriate diagram is not part of UML
use it anyways

UML & Requirements

- K Use Cases**
 - Describe how people interact with the system
- K Class Diagram**
 - Drawn from a conceptual perspective
 - Can build up a rigorous vocab of the domain
- K Activity Diagram**
 - Shows the workflow of the org.
 - Shows how s/w and human activities interact
 - Context for Use Cases
 - Details of complex Use Cases

Set 3 13

UML & Requirements

- K State Diagram**
 - Shows states and events that change the state
 - Can be useful with interesting life cycles

Communication is key

Customers may not be familiar with S/W techniques
Break the rules if it enhances Communication

Set 3 14

Scenarios

- K A Scenario** is an example of what happens when someone interacts with the system
- K Describes the system from an external viewpoint**
- K EXAMPLE Scenario** – Medical Clinic:
 - "A patient calls the clinic to make an appointment for a yearly checkup. The receptionist finds the nearest empty time slot in the appointment book and schedules the appointment for that time slot."

Set 3 15

What is a Use Case?

- K A use case** is a reason to use the system
- K Again** – describes the system from an *external* viewpoint
"provides an outsider's view"
- K A way of formalizing scenarios**
- K A summary of scenarios for a single task or goal**
- K Treat system as a black box**
 - Don't incorporate design decisions
 - Applies unnecessary constraints at design

Use Case Diagrams describe the dynamic behavior of the system

Set 3 16

Use Case Basics

- K Actors**
 - who or what initiates the events involved in that task
 - roles that people/objects/systems (anything *external* to the system) play
 - Represented as stick figures
- K Use Case** – some system function (a summary of related scenarios)
 - Represented as an oval
- K Communication** (or Communication Association)
 - A Connection between the actor and the use case
 - Represented as a line

```

graph LR
    Actor[Patient] -- communication --> UseCase((Make Appointment))
  
```

Set 3 17

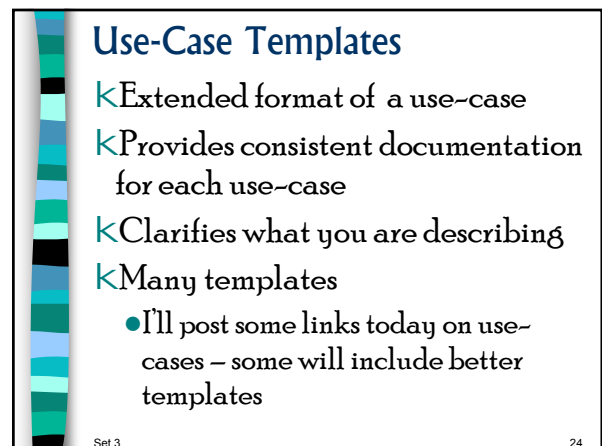
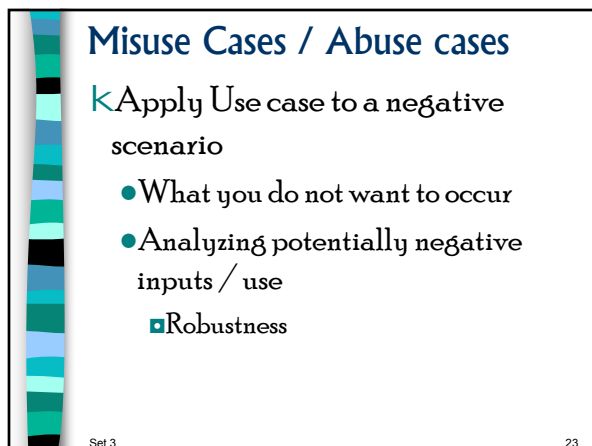
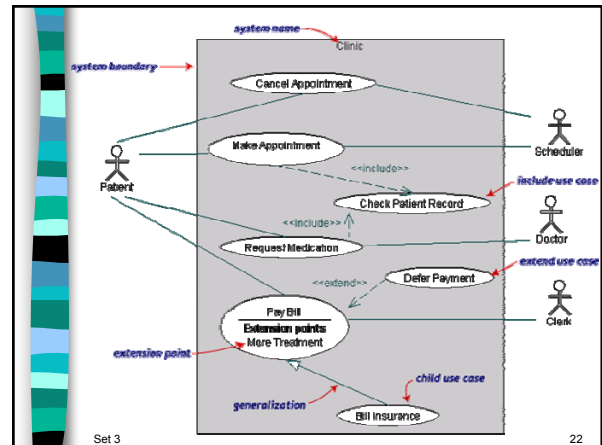
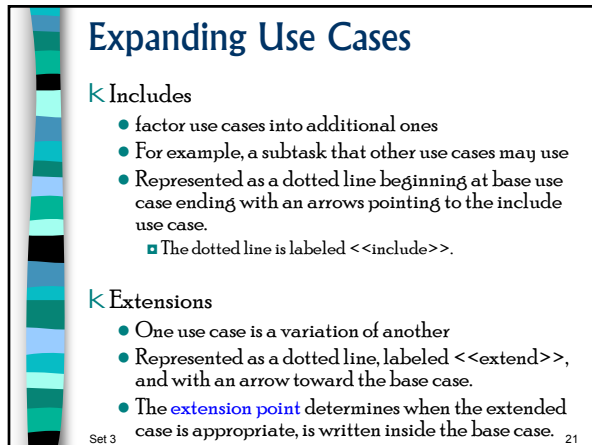
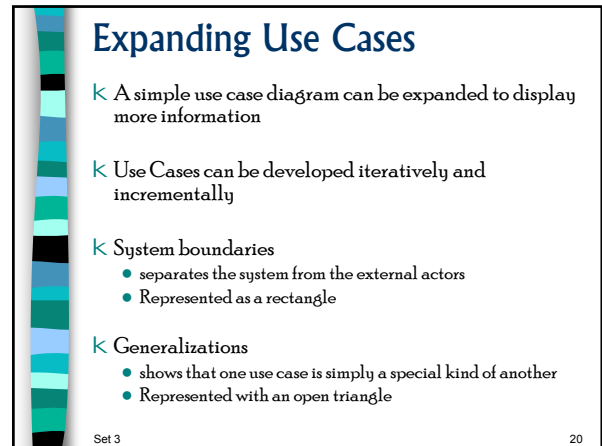
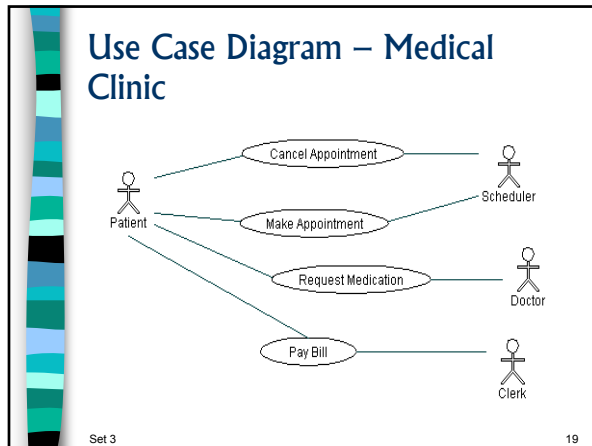
Use Case Diagrams

- K A collection of actors, use cases, and their associations**

Use case diagrams are helpful in three areas

- K Determining features (requirements)**
 - New use cases often generate new requirements.
 - Can happen during design and system analysis
- K Communicating with clients**
 - Simple notation makes them easy to understand
- K Generating test cases**
 - The collection of scenarios for a use case may suggest a suite of test cases for those scenarios

Set 3 18





Project Plans

- ↳ Basically, a timeline for your project
- ↳ Include all the minor details not included in the class schedule
- ↳ How will you get there?
- ↳ Include things you need to learn
 - With deadlines
- ↳ Many templates online

Set 3

25



Templates

- ↳ Look online
- ↳ Check out ready-set
- ↳ Use previous texts
 - Or previous course-work
- ↳ Decide which is most appropriate for your application

Set 3

26