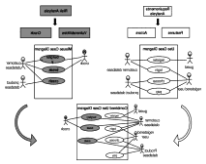
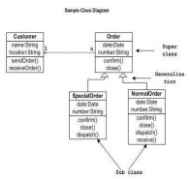
Requirements Modeling

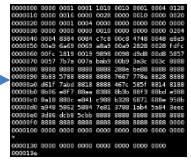
Some of the slides are designed and adapted from slides provided by Software Engineering: A Practitioner's Approach, 8e





What do we have before that?

```
if (hour < 18) {
   greeting = "Good day";
} else {
   greeting = "Good evening";
}
```



Requirements

Analysis

Design

- results in the specification of software's operational characteristics, indicates software's interface with other system elements, and establishes constraints that software must meet.
- elaborate on basic requirements established during the inception, elicitation, and negotiation tasks that are part of requirements engineering

- Software requirement Analysis
  - Problem recognition
  - Throughout analysis modeling, your primary
    - focus is on what, not how.

What user interaction occurs in a particular circumstance,

what objects does the system manipulate,

what functions must the system perform,

what behaviors does the system exhibit,

what interfaces are defined, and

what constraints apply?

- Modeling
- Specification
- Review

Overall Objectives and Philosophy

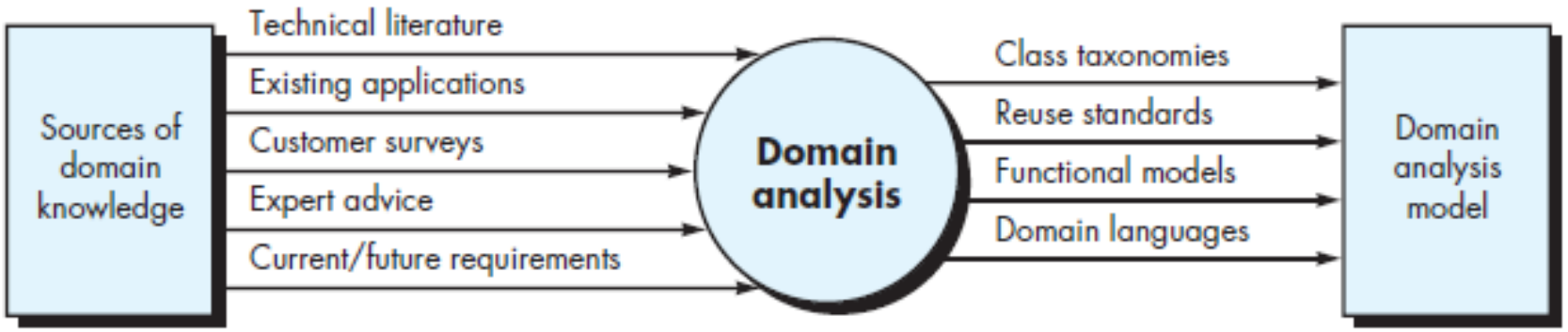
- should model what is known and use that model as the basis for design of the software increment
- The requirements model must achieve three primary objectives:
  - **to describe what** the **customer requires**,
  - **to establish a basis for the creation of a software design**, and
  - **to define a set of requirements** that can be **validated once the software is built**.
- important to note that all elements of the requirements model will be directly traceable to parts of the design model

Analysis Rules of Thumb

Arlow and Neustadt [Arl02] suggest a number of worthwhile rules of thumb that should be followed when creating the analysis modelling

- **focus on requirements** *that are visible within the problem or business domain.*
- **should add to an overall understanding of software requirements** *and provide insight into the information domain, function, and behavior of the system.*
- **Delay consideration of infrastructure and other nonfunctional models** *until design*
- **Minimize coupling** throughout the system.
- **Keep the model as simple** as it can be.

Domain Analysis



Input and output for domain analysis

Requirements Modeling Approaches

- **Structured analysis:**
  data and the processes that transform the data as separate entities

- **Object-oriented analysis**:
  focuses on the definition of classes and the manner in which they collaborate with one another to effect customer requirements.

UML and the Unified Process are predominantly object oriented.

UML

- Unified modelling language
- UML provides extension mechanisms
- Unified, specify, Visual, Language, Models, documenting
- Eight diagrams and one language

Use-Case **Class/object** **Interaction**

Activity **State**

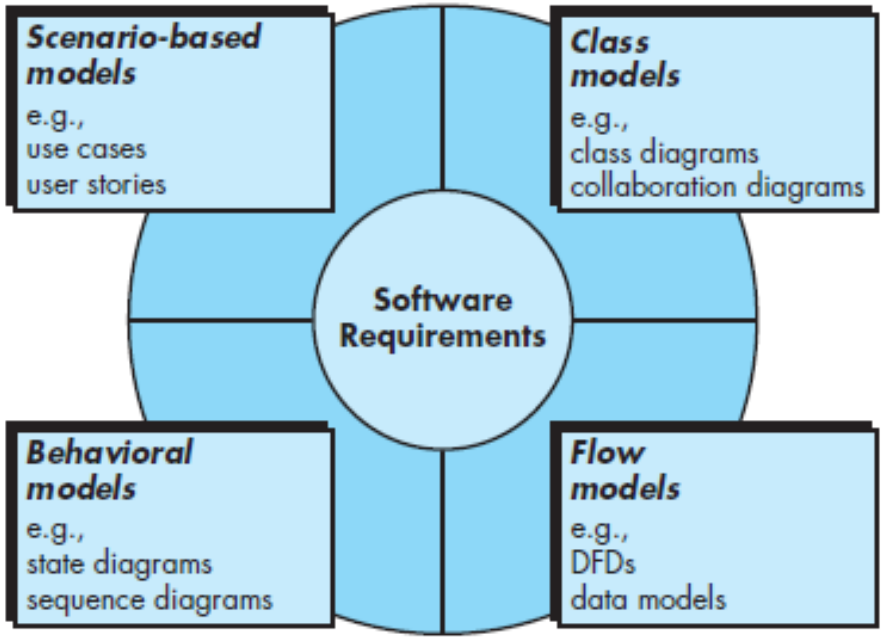
Component **Sequence** **Deployment**

Some of the slides are designed and adapted from slides provided by Software Engineering: A Practitioner's Approach, 8e

Requirements Modeling Approaches

represent object-oriented classes (attributes and operations) and the manner
in which classes collaborate to achieve system requirements.

of requirements from the point of view of various system "actors"



how the software behaves as a consequence of external "events."

of requirements from the point of view of various system "actors"

Elements of the analysis model

Scenario based Modeling

Some of the slides are designed and adapted from slides provided by Software Engineering: A Practitioner's Approach, 8e

Creating a Preliminary Use Case

Alistair Cockburn characterizes a **use case** as a **"contract for behavior"**

"contract"

actor —— uses a computer-based system ——→ to accomplish some goal

use case captures the **interactions** that occur between **producers and consumers** of information and the system itself

But how do you know

(1) what to write about,

(2) how much to write about it,

(3) how detailed to make your description, and

(4) how to organize the description?

What to Write About?

Inception and elicitation—provide you with the information you'll need to begin writing use cases.

To begin developing a set of use cases,

list the functions or activities performed by a specific actor.

by an evaluation of activity diagrams developed as part of requirements modeling which we discuss later on.

Use case: Access camera surveillance via the Internet—display camera views (ACS-DCV)

Actor: homeowner
If I'm at a remote location, I can use any PC with appropriate browser software to log on to the *SafeHome Products* website. I enter my user ID and two levels of passwords and once I'm validated, I have access to all functionality for my installed *SafeHome* system. To access a specifi c camera view, I select "surveillance" from the major function buttons displayed. I then select "pick a camera" and the fl oor plan of the house is displayed. I then select the camera that I'm interested in. Alternatively, I can look at thumbnail snapshots from all cameras simultaneously by selecting "all cameras" as my viewing choice. Once I choose a camera, I select "view" and a one-frame-per-second view appears in a viewing window that is identifi ed by the camera ID. If I want to switch cameras, I select "pick a camera" and the original viewing window disappears and the fl oor plan of the house is displayed again. I then select the camera that I'm interested in. A new viewing window appears.

Actor: homeowner
1. The homeowner logs onto the SafeHome Products website.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects the "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the fl oor plan of the house.
8. The homeowner selects a camera icon from the fl oor plan.
9. The homeowner selects the "view" button.

Refining a Preliminary Use Case

description of alternative interactions is essential for a complete understanding of the function

each step in the primary scenario is evaluated by asking the following questions

- Can the actor take some other action at this point?
- Is it possible that the actor will encounter some error condition at this point? If so, what might it be?
- Is it possible that the actor will encounter some other behavior at this point (e.g., behavior that is invoked by some event outside the actor's control)? If so, what might it be?

Answers to these questions result in the creation of a set of **secondary scenarios**

Refining a Preliminary Use Case

The homeowner selects “pick a camera.” (PS)

The system displays the floor plan of the house.(PS)

- Can the actor take some other action at this point? ▪ YES
- Is it possible that the actor will encounter some error condition at this point? If so, what might it be? ▪ YES

- Is it possible that the actor will encounter some behavior at this point?
  - YES

Answers to these questions result in the creation of a set of **secondary scenarios**

View thumbnail snapshots for all cameras.”

No floor plan configured for this house

Alarm condition encountered

Write a formal use case

identifies the overall scope of the use case.

what is known to be true before the use case is initiated.

identifies the event or condition that "gets the use case started"

lists the specific actions that are required by the actor and the appropriate system responses.

identify the situations uncovered as the preliminary use case is refined

Use Case Template for Surveillance

Use case: Access camera surveillance via the Internet—display camera views (ACS-DCV)

Iteration: 2, last modification: January 14 by V. Raman.

Primary actor: Homeowner.

Goal in context: To view output of camera placed throughout the house from any remote location via the Internet.

Preconditions: System must be fully configured; appropriate user ID and passwords must be obtained.

Trigger: The homeowner decides to take a look inside the house while away.

Scenario:

1. The homeowner logs onto the *SafeHome Products* website.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects the "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Exceptions:

1. ID or passwords are incorrect or not recognized—see use case **Validate ID and passwords.**
2. Surveillance function not configured for this system—system displays appropriate error message; see use case **Configure surveillance function.**
3. Homeowner selects "View thumbnail snapshots for all camera"—see use case **View thumbnail snapshots for all cameras.**
4. A floor plan is not available or has not been configured—display appropriate error message and see use case **Configure floor plan.**
5. An alarm condition is encountered—see use case **alarm condition encountered.**

Priority: Moderate priority, to be implemented after basic functions.

When available: Third increment.

Frequency of use: Infrequent.

Channel to actor: Via PC-based browser and Internet connection.

Secondary actors: System administrator, cameras.

Channels to secondary actors:

1. System administrator: PC-based system.
2. Cameras: wireless connectivity.

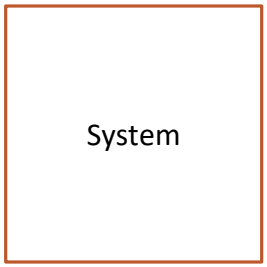
Open issues:

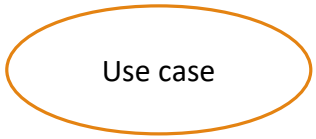
1. What mechanisms protect unauthorized use of this capability by employees of *SafeHome Products*?
2. Is security sufficient? Hacking into this feature would represent a major invasion of privacy.
3. Will system response via the Internet be acceptable given the bandwidth required for camera views?
4. Will we develop a capability to provide video at a higher frames-per-second rate when high-bandwidth connections are available?

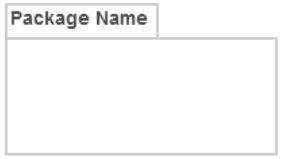
Use case Diagrams

- Consists of
  - An actor
  - Use case
  - System
  - Package

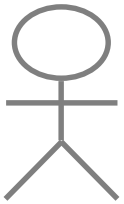
An Actor







Use case Analysis



An Actor

An actor is a model for an external entity which interacts (communicates) with the system:

- Actor have a name
- External system (Another system)
- Physical environment (e.g. Weather)

An actor has a unique name and an optional description

♦ Examples:

Name

Optional description

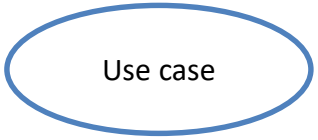
Passenger: A person in the train

Customer: A person who buys online

Use Case

- represents a function or an action within the system.
- Each use case has a descriptive name
- Describes what a system does but not how it does it.
- Use case names must be unique within a given package
- drawn as an oval and named with the function.

• Use cases can be described textually, with a focus on the event flow between actor and system

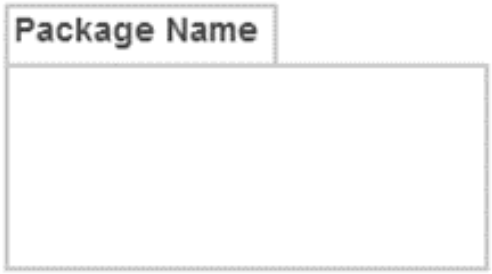


System

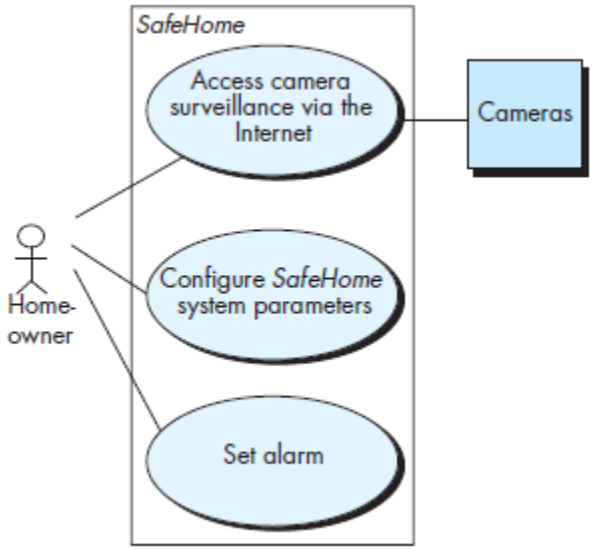
- define the scope of the use case
- drawn as a rectangle.
- useful when you're visualizing large systems

Package

- optional element that is extremely useful in complex diagrams.
- used to group together use case.



Preliminary
Use case diagram



use case diagram for the SafeHome system

Developing an Activity Diagram

Some of the slides are designed and adapted from slides provided by Software Engineering: A Practitioner's Approach, 8e

Preliminary
Use case diagram

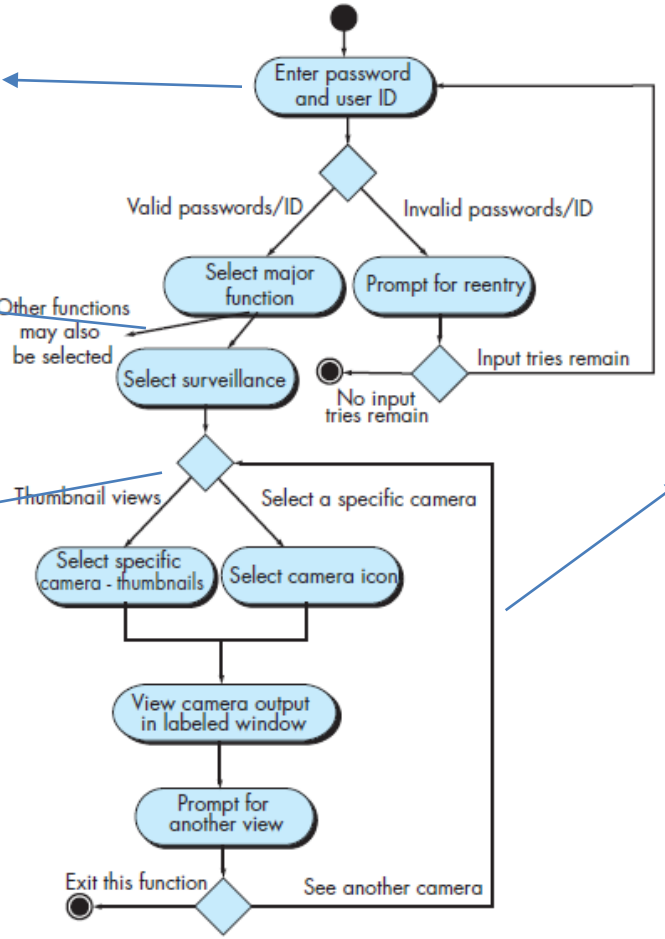
UML activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario

rounded rectangles to imply a specific system function

arrows to represent flow through the system

decision diamonds to depict a branching decision

solid horizontal lines to indicate that parallel activities are occurring



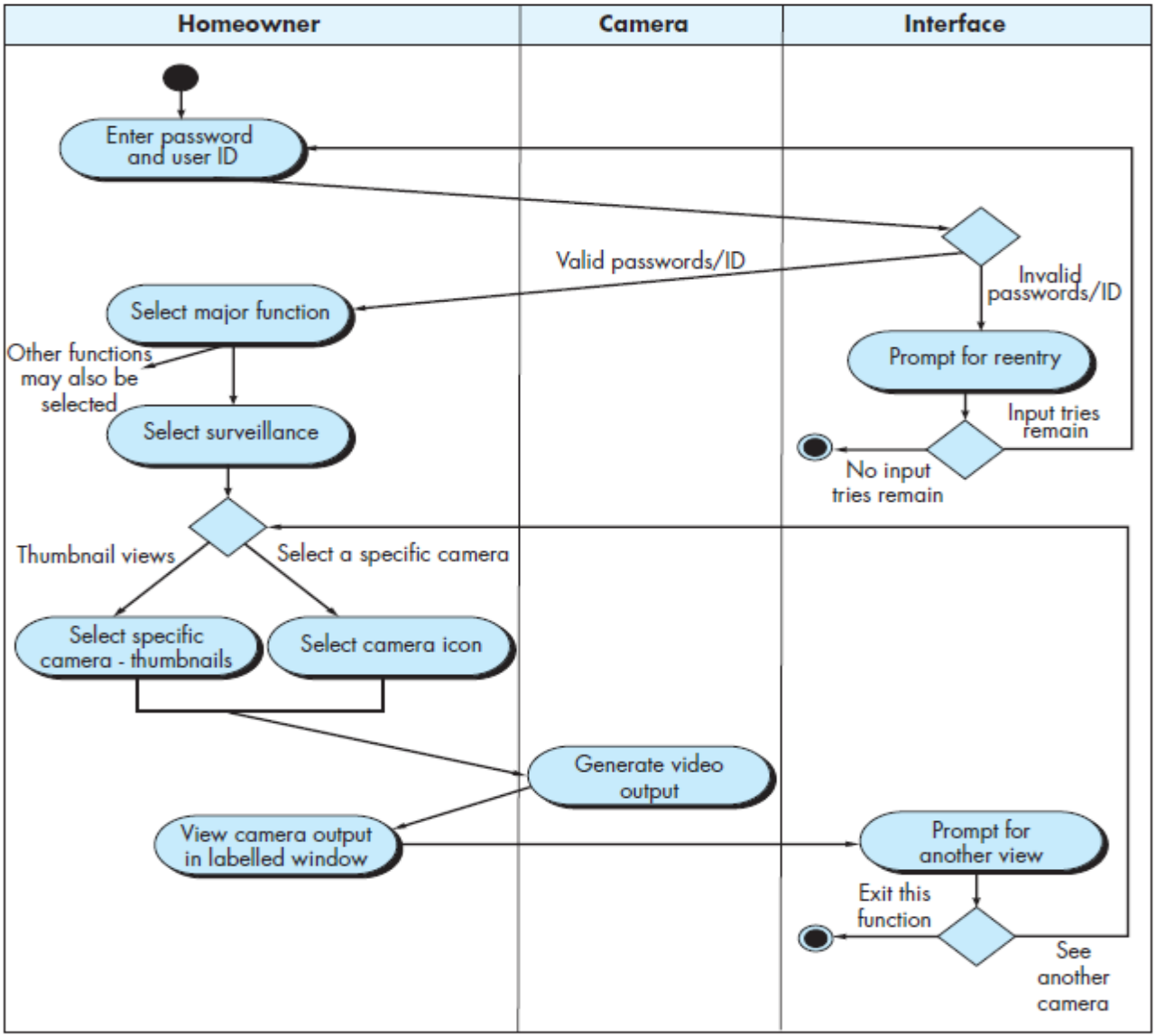
Developing an swimlane Diagram

Swimlane diagram

- UML swimlane diagram is a useful variation of the activity diagram

- Allows to represent the flow of activities described by the use case and at the same time indicate which actor (if there are multiple actors involved in a specific use case) or analysis class has responsibility for the action described by **an activity rectangle**

Swimlane diagram



Swimlane diagram for Access camera surveillance via the Internet—display camera views function.

- **Requirements Elicitation**

Deadline: 09/25/2017
Template on beach board

- **Requirements Analysis**

- Identify the actors within your system
- Write the use case of your system (Follow the template which is on slide 15).
- Draw a UML use case diagram of your system. You'll have to make a number of assumptions about the manner in which a user interacts with this system.
- Develop an activity diagram for one aspect of your system.
- Develop a swimlane diagram for one or more aspects of your system.

Deadline: 10/02/2017
Template on beach board