

Are you ready?

☐ A Yes

☐ B No



提交

Software Engineering

Part 2 Modeling

Chapter 9 Requirements Modeling Scenario- Based Methods

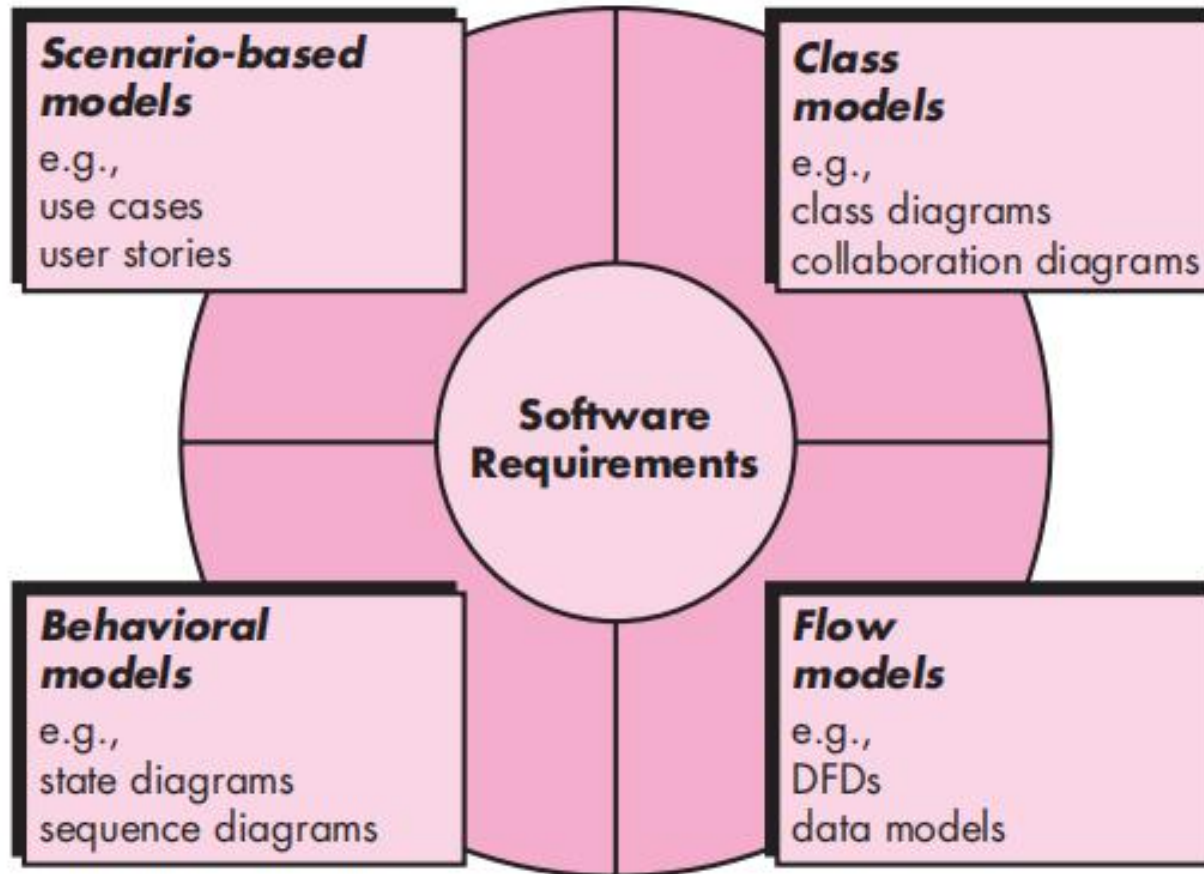
Contents

- 9.1 Requirements Analysis
 - 9.1.1 Overall Objectives and Philosophy
 - 9.1.2 Analysis Rules of Thumb
 - 9.1.3 Domain Analysis
 - 9.1.4 Requirements Modeling Approaches
- 9.2 Scenario-based Modeling
 - 9.2.1 Creating a Preliminary Use Case
 - 9.2.2 Refining a Preliminary Use Case
 - 9.2.3 Writing a Formal Use Case
- 9.3 UML Models That Supplement the Use Case
 - 9.3.1 Developing an Activity Diagram
 - 9.3.2 Swimlane Diagrams

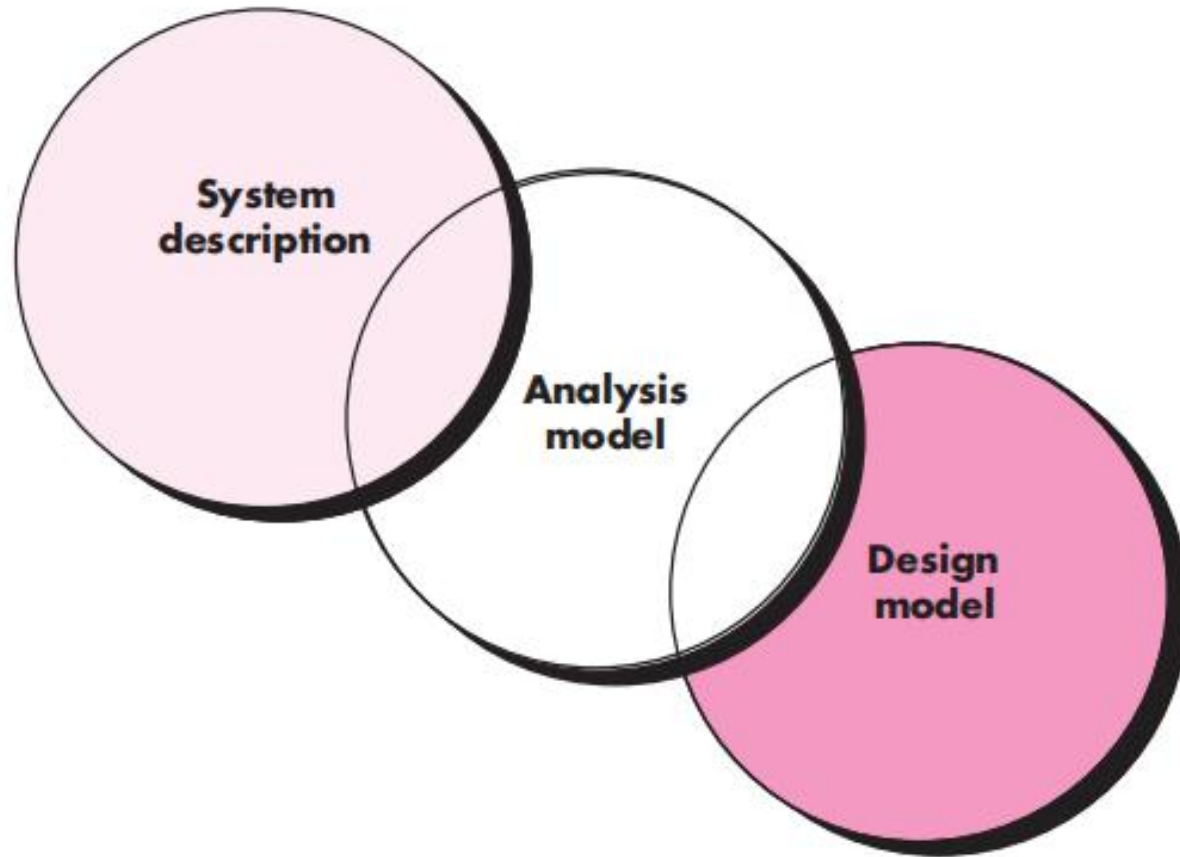
9.1 Requirements Analysis

- Requirements analysis
 - specifies software's **operational characteristics**
 - indicates **interface** with other system elements
 - establishes **constraints** that software must meet
- Requirements analysis allows the software engineer to:
 - **elaborate** on basic requirements established during earlier requirement engineering tasks
 - **build models** that depict user scenarios, functional activities, problem classes and their relationships, system and class behavior, and the flow of data as it is transformed.

9.1.1 Elements of Requirements Analysis



9.1.1 A Bridge between the system description and the design model

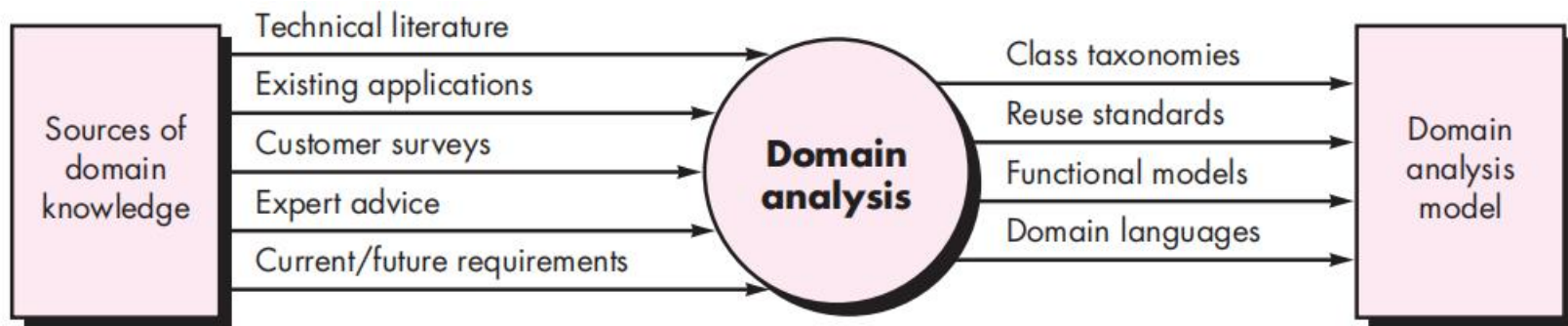


9.1.2 Rules for creating analysis model

- The model should focus on **requirements** that are visible **within the problem or business domain**. The level of abstraction should be relatively high.
- Each element of the analysis model 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 non-functional models until design.
- Be certain that the **analysis model provides value to all stakeholders**.
- Keep the model **as simple as** it can be.

9.1.3 Domain Analysis

- Define the domain to be investigated.
- Collect a representative sample of applications in the domain.
- Analyze each application in the sample.
- Develop an analysis model for the objects.



Input and output for domain analysis

9.1.3 Domain Analysis

Let's try a role play for domain analysis!



9.1.3 Domain Analysis

SAFEHOME



Domain Analysis

The scene: Doug Miller's office, after a meeting with marketing.

The players: Doug Miller, software engineering manager, and Vinod Raman, a member of the software engineering team.

The conversation:

Doug: I need you for a special project, Vinod. I'm going to pull you out of the requirements-gathering meetings.

Vinod (frowning): Too bad. That format actually works . . . I was getting something out of it. What's up?

Doug: Jamie and Ed will cover for you. Anyway, marketing insists that we deliver the Internet capability along with the home security function in the first release of *SafeHome*. We're under the gun on this . . . not enough time or people, so we've got to solve both problems—the PC interface and the Web interface—at once.

Vinod (looking confused): I didn't know the plan was set . . . we're not even finished with requirements gathering.

Doug (a wan smile): I know, but the time lines are so short that I decided to begin strategizing with marketing right now . . . anyhow, we'll revisit any tentative plan once we have the info from all of the requirements-gathering meetings.

Vinod: Okay, what's up? What do you want me to do?

Doug: Do you know what "domain analysis" is?

Vinod: Sort of. You look for similar patterns in Apps that do the same kinds of things as the App you're building. If possible, you then steal the patterns and reuse them in your work.

Doug: Not sure I like the word *steal*, but basically you have it right. What I'd like you to do is to begin researching existing user interfaces for systems that control something like *SafeHome*. I want you to propose a set of patterns and analysis classes that can be common to both the PC-based interface that'll sit in the house and the browser-based interface that is accessible via the Internet.

Vinod: We can save time by making them the same . . . why don't we just do that?

Doug: Ah . . . it's nice to have people who think like you do. That's the whole point—we can save time and effort if both interfaces are nearly identical, implemented with the same code, blah, blah, that marketing insists on.

Vinod: So you want, what—classes, analysis patterns, design patterns?

Doug: All of 'em. Nothing formal at this point. I just want to get a head start on our internal analysis and design work.

Vinod: I'll go to our class library and see what we've got. I'll also use a patterns template I saw in a book I was reading a few months back.

Doug: Good. Go to work.

9.2.1 Scenario-Based Modeling

“[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

- (1) What should we write about?**
- (2) How much should we write about it?**
- (3) How detailed should we make our description?**
- (4) How should we organize the description?**

9.2.1 What to Write About?

- **Inception and elicitation** — provide you with the information you'll need to begin writing use cases.
- **Requirements gathering meetings, and other requirements engineering mechanisms** are used to
 - identify stakeholders
 - define the scope of the problem
 - specify overall operational goals
 - establish priorities
 - outline all known functional requirements, and
 - describe the things (objects) that will be manipulated by the system.
- To begin developing a set of use cases, **list the functions or activities performed by a specific actor.**

9.2.1 How Much to Write About?

- As further conversations with the stakeholders progress, the requirements gathering team develops use cases for each of the functions.
- In general, use cases are written first in an informal narrative fashion.
- If more formality is required, the same use case is rewritten using a structured format similar to the one proposed.

9.2.1 Scenario-Based Modeling

Let's try a role play for scenario!



9.2.1 Scenario-Based Modeling

SafeHome



Developing Another Preliminary User Scenario

The scene: A meeting room, during the second requirements-gathering meeting.

The players: Jamie Lazar, software team member; Ed Robbins, software team member; Doug Miller, software engineering manager; three members of marketing; a product engineering representative; and a facilitator.

The conversation:

Facilitator: It's time that we begin talking about the *SafeHome* surveillance function. Let's develop a user scenario for access to the surveillance function.

Jamie: Who plays the role of the actor on this?

Facilitator: I think Meredith (a marketing person) has been working on that functionality. Why don't you play the role?

Meredith: You want to do it the same way we did it last time, right?

Facilitator: Right . . . same way.

Meredith: Well, obviously the reason for surveillance is to allow the homeowner to check out the house while he or she is away, to record and play back video that is captured . . . that sort of thing.

Ed: Will we use compression to store the video?

Facilitator: Good question, Ed, but let's postpone implementation issues for now. Meredith?

Meredith: Okay, so basically there are two parts to the surveillance function . . . the first configures the

system including laying out a floor plan—we have to have tools to help the homeowner do this—and the second part is the actual surveillance function itself. Since the layout is part of the configuration activity, I'll focus on the surveillance function.

Facilitator (smiling): Took the words right out of my mouth.

Meredith: Um . . . I want to gain access to the surveillance function either via the PC or via the Internet. My feeling is that the Internet access would be more frequently used. Anyway, I want to be able to display camera views on a PC and control pan and zoom for a specific camera. I specify the camera by selecting it from the house floor plan. I want to selectively record camera output and replay camera output. I also want to be able to block access to one or more cameras with a specific password. I also want the option of seeing small windows that show views from all cameras and then be able to pick the one I want enlarged.

Jamie: Those are called thumbnail views.

Meredith: Okay, then I want thumbnail views of all the cameras. I also want the interface for the surveillance function to have the same look and feel as all other *SafeHome* interfaces. I want it to be intuitive, meaning I don't want to have to read a manual to use it.

Facilitator: Good job. Now, let's go into this function in a bit more detail . . .

9.2.1 Scenario-Based Modeling



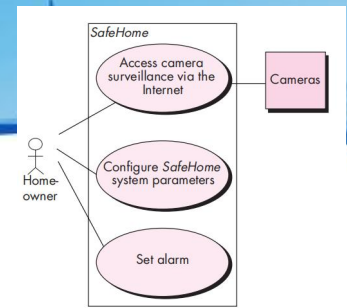
Homeowner actor:

- Select camera to view.
- Request thumbnails from all cameras.
- Display camera views in a PC window.
- Control pan and zoom for a specific camera.
- Selectively record camera output.
- Replay camera output.
- Access camera surveillance via the Internet.

9.2.2 Use-Cases

- a scenario that describes a “thread of usage” for a system
- *actors* represent roles people or devices play as the system functions
- *users* can play a number of different roles for a given scenario

9.2.3 Use-Case Diagram

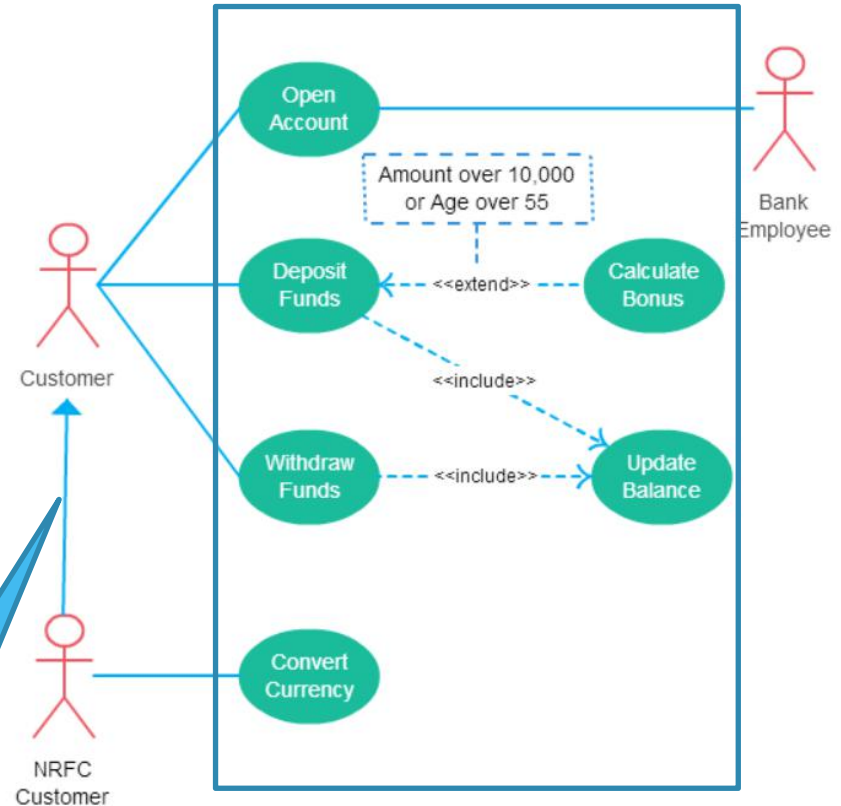


- **Use cases:** Horizontally shaped **ovals** that represent the different uses that a user might have.
- **Actors:** **Stick figures** that represent the people actually employing the use cases.
- **Associations:** **A line between actors and use cases.** In complex diagrams, it is important to know which actors are associated with which use cases.
- **System boundary boxes:** **A box** that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system.

9.2.3 Use-Case Diagram

Relationship in use-case diagram

Type	Description
Association	between actor and use case
Generalization (Inheritance)	between actor or between use case
Include (Every time)	between use case
Extend (Some case)	between use case



Inheritance

tools:

1. Umlet
2. <https://online.visual-paradigm.com/diagrams.jsp#diagramlist:proj=0&new=UseCaseDiagram>

9.2.3 Use-Case Diagram

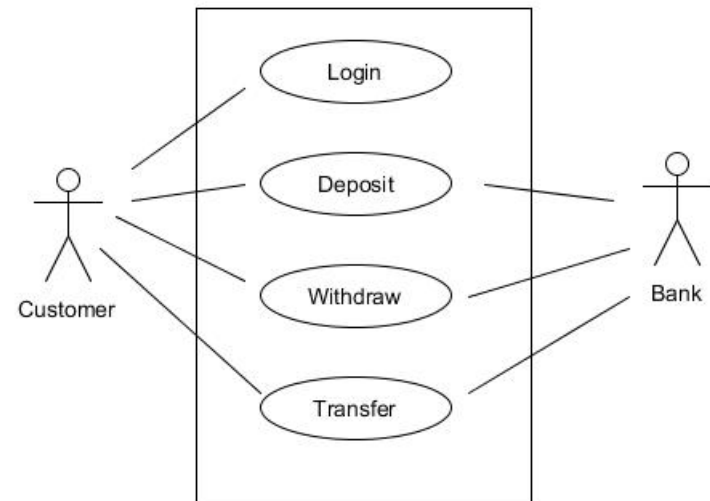
- Use case \neq Function Decomposition

Example:

Function Decomposition

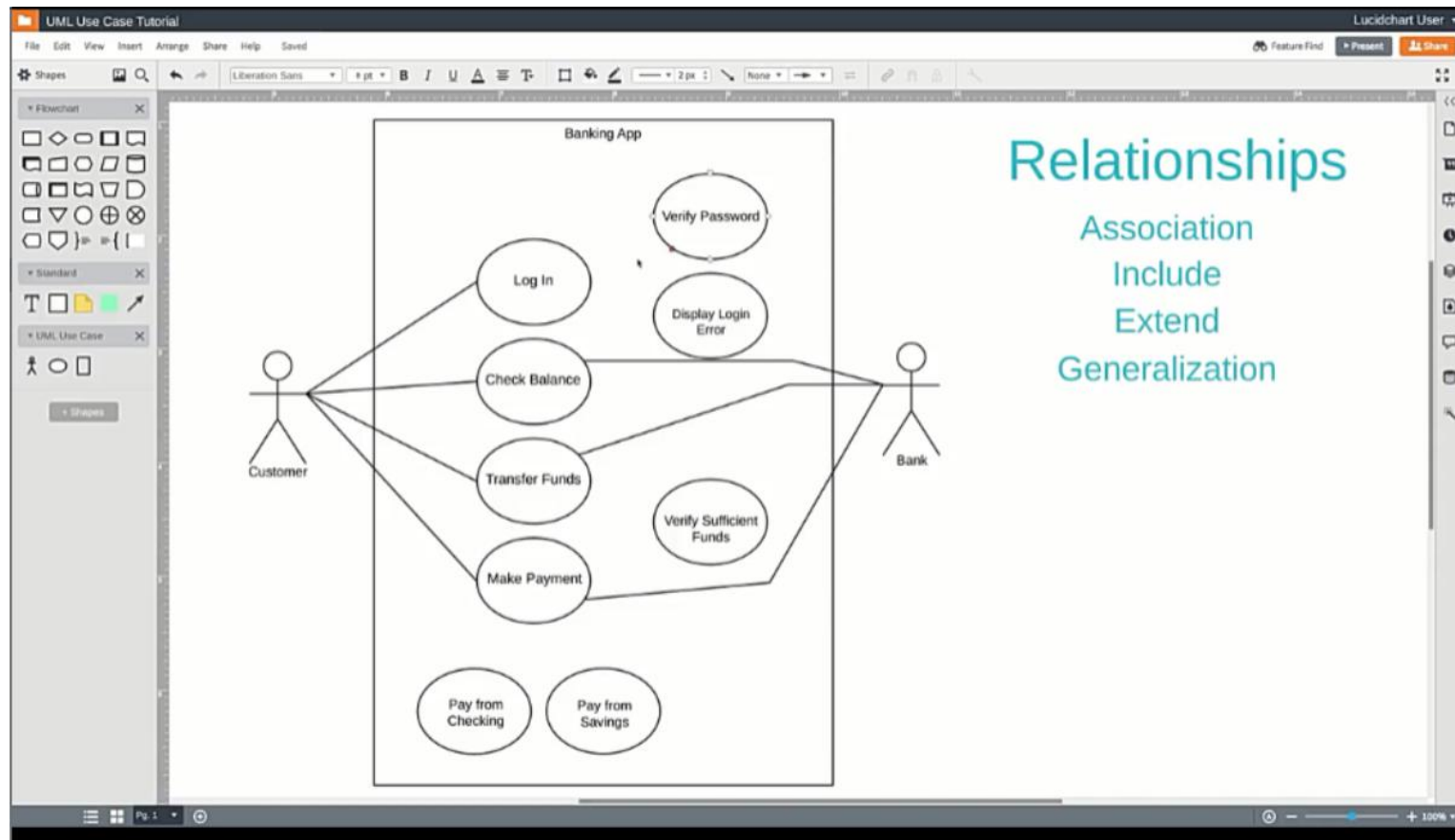
- Insert card
- Input password
- Select withdraw
- Select deposit
- Confirm balance
- Process transaction (bank)
- Input transfer account
- Input transfer money

Use case



9.2.3 Use-Case Diagram

Let's [watch a video!](#)



9.2.3 Developing a Use-Case

- What are the **main tasks or functions** that are performed by the actor?
- What **system information** will the 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?

9.2.3 Reviewing a Use-Case

- Use-cases are written first in narrative form and mapped to a template if formality is needed.
- Each primary scenario should be reviewed and refined to see if alternative interactions are possible
 - Can the actor take some other action at this point?
 - Is it possible that the actor will encounter an error condition at some point? If so, what?
 - Is it possible that the actor will encounter some other behavior at some point? If so, what?

9.2.3 Exceptions

- Describe situations (failures or user choices) that cause the system to exhibit **unusual behavior**
- **Brainstorming** should be used to derive a reasonably complete set of exceptions for each use case
- Are there cases where **a validation function** occurs for the use case?
 - Are there cases where a supporting function (actor) fails to respond appropriately?
 - Can poor system performance result in unexpected or improper use actions?
- Handling **exceptions** may require the creation of additional use cases

9.2.3 Reviewing a Use-Case

SAFEHOME



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: Moderate frequency.

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

Secondary actors: System administrator, cameras.

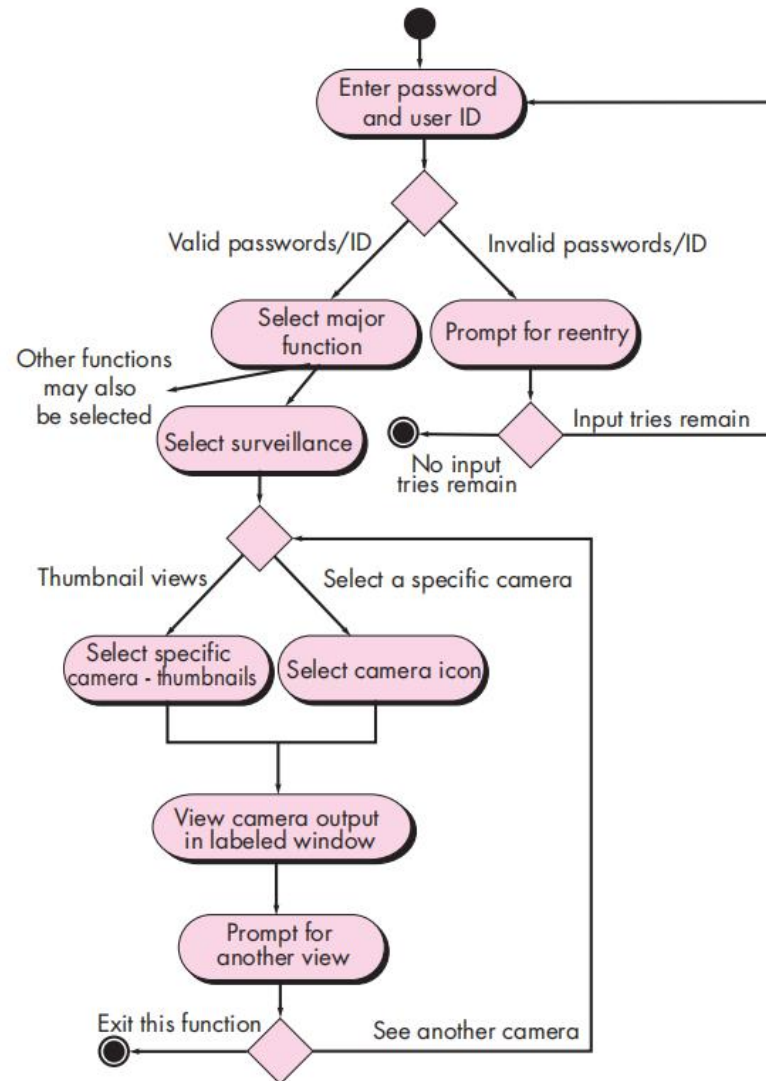
Channels to secondary actors:

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

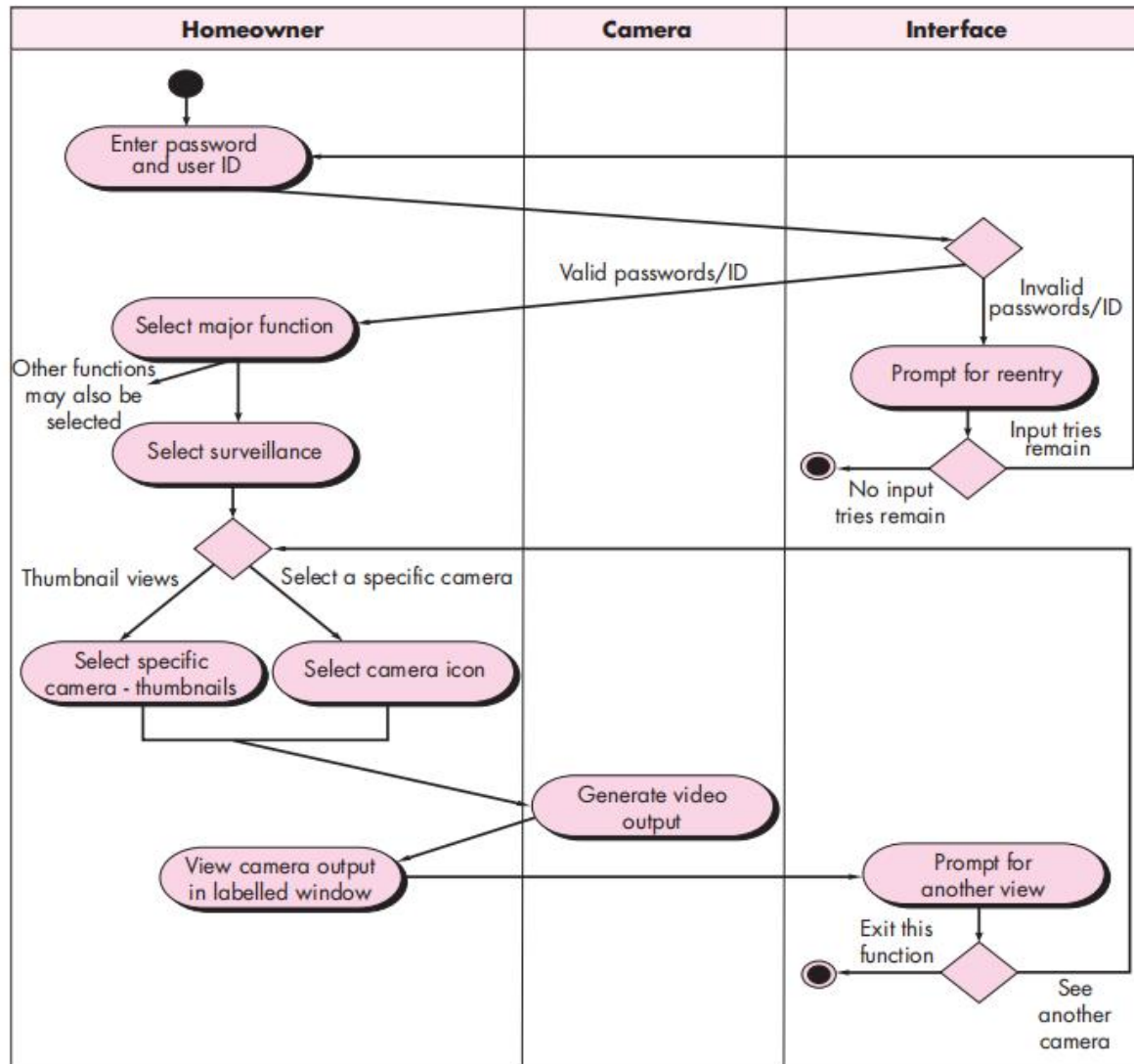
Open issues:

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?

9.3 Activity Diagram



9.3 Swimlane Diagrams



Review - Scenario-Based Methods

- Use case Diagram
- Activity Diagram
- Swimlane Diagrams



THE END