Prof. Dr. Nizamettin AYDIN

naydin@yildiz.edu.tr
http://www.yildiz.edu.tr/~naydin

1

Functional Modeling

Objectives:

- Understand the rules and style guidelines for activity diagrams.
- Understand the rules and style guidelines for use cases and use case diagrams.
- Understand the process used to create use cases and use case diagrams.
- Be able to create functional models using activity diagrams, use cases, and use case diagrams.
- Become familiar with the use of use case points.

- Requirements-gathering techniques
  - interviewing, JAD, and observation.
- Using these techniques, the analyst determined the requirements and created a requirements definition.
  - requirements definition defines what the system is to do.
- UML has been accepted as the standard notation by the Object Management Group (OMG)
- almost all OO development projects today utilize activity diagrams and use cases
  - to document and organize the requirements that are obtained during the analysis phase.

3

- An activity diagram can be used for any type of process modeling activity.
- Process models depict how a business system operates.
  - They illustrate the processes or activities that are performed and how objects (data) move among them.
- A process model can be used to document
  - the current system (i.e., as-is system)
  - the new system being developed (i.e., to-be system),

whether computerized or not.

4

- A use case is a formal way of representing how a business system interacts with its environment.
  - illustrates the activities that are performed by the users of the system.
    - As such, use case modeling is often thought of as an external or functional view of a business process in that it shows how the users view the process, rather than the internal mechanisms by which the process and supporting systems operate.
- Use cases can be used to document
  - the current system (i.e., as-is system)
  - the new system being developed (i.e., to-be system).

5

- Activity diagrams and use cases are logical models
  - describing the business domain's activities without suggesting how they are conducted.
  - sometimes referred to as problem domain models.
- When reading an activity diagram or use case, in principle, you should not be able to tell if an activity is computerized or manual
  - if a piece of information is collected by paper form or via the Web;
  - if information is placed in a filing cabinet or a large database.

6

- These physical details are defined during the design phase,
  - when the logical models are refined into physical models.
- These models provide information that is needed to ultimately build the system.
- By focusing on logical activities first,
  - analysts can focus on how the business should run without being distracted with implementation details.

7

Business Process Modeling with Activity Diagrams

- Elements of an Activity Diagram
- Guidelines for Creating Activity Diagrams

- Business process models describe the different activities that when combined together support a business process.
- Business processes typically cut across functional departments
- From an OO perspective, they cut across multiple objects.
  - As such, many of the earlier OO systems development approaches tended to ignore business process modeling.
  - Instead, they focused only on modeling processes via use cases and behavioral models.
- The use of UML 2.0's activity diagrams as a means to build business process models will be introduced.

9

Activity diagrams

- used to model the behavior in a business process independent of objects.
- can be viewed as sophisticated data flow diagrams that are used in conjunction with structured analysis.
- include notation that addresses the modeling of parallel, concurrent activities and complex decision processes.
  - they can be used to model everything from a high-level business workflow that involve many different use cases, to the details of an individual use case, all the way down to the specific details of an individual method.
- activity diagrams can be used to model any type of process.

10

Elements of an Activity Diagram

- Activity diagrams
  - portray the primary activities and the relationships among the activities in a process.
- Next couple of slides will present elements of the activity diagrams

11

Actions and activities

- **An Action:**
  - Is a simple, non-decomposable piece of behavior
  - Is labeled by its name

- **An Activity:**
  - Is used to represent a set of actions
  - Is labeled by its name

12

Actions and activities

- performed for some specific business reason.
- can represent manual or computerized behavior.
- should have a name that begins with a verb and ends with a noun
  - e.g., “Make Appointment” or “Make Payment Arrangements”.
- Names should be short, yet contain enough information
  - the reader should understand exactly what they do.
- Difference between an action and an activity:
  - an activity can be decomposed further into a set of activities and/or actions
  - an action represents a simple nondecomposable piece of the overall behavior being modeled.

13

Object Nodes

- **An Object Node:**
  - Is used to represent an object that is connected to a set of Object Flows
  - Is labeled by its class name

- Activities and actions typically modify or transform objects.
- Object nodes model these objects in an activity diagram.
- Object nodes are portrayed in an activity diagram as a rectangle.
- The name of the class of the object is written inside of the rectangle.
  - Essentially, object nodes represent the flow of information from one activity to another activity.

14

Control Flows and Object Flows

- **A Control Flow:**
  - Shows the sequence of execution
- **An Object Flow:**
  - Shows the flow of an object from one activity/action to another activity/action
- Control flows
  - model the paths of execution through a business process.
  - solid line with an arrowhead showing the direction of flow.
  - can only be attached to actions or activities.
- Object flows
  - model the flow of objects through a business process.
  - dashed line with an arrowhead showing the direction of flow.
  - An individual object flow must be attached to an action or activity on one end and an object node on the other end.

15

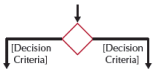
Control Nodes

- **An Initial Node:**
  - Portrays the beginning of a set of actions or activities
- **A Final-Activity Node:**
  - Is used to stop all control flows and object flows in an activity (or action)
- **A Final-Flow Node:**
  - Is used to stop a specific control flow or object flow

16

Control Nodes

- **A Decision Node:**
  - used to represent a test condition to ensure that the control flow or object flow only goes down one path
  - labeled with the decision criteria to continue down the specific path



- **A Merge Node:**
  - used to bring back together different decision paths that were created using a decision-node

17

Control Nodes

- **A Fork Node:**
  - Is used to split behavior into a set of parallel or concurrent flows of activities/actions
- **A Join Node:**
  - Is used to bring back together a set of parallel or concurrent flows of activities/actions

18

Control Nodes

- An initial node
- portrays the beginning of a set of actions or activities.
- shown as a small filled in circle.
- A final-activity node
- used to "stop the process" being modeled.
  - Any time a final-activity node is reached, all actions and activities are ended immediately, regardless of whether they are completed.
- represented as a circle surrounding a small filled-in circle
- A final-flow node
- stops a specific path of execution through the business process,
  - but allows the other concurrent or parallel paths to continue.
- shown as a small circle with an X in it.

19

Control Nodes

- The decision node
- used to represent the actual test condition that is used to determine which of the paths exiting the decision node is to be traversed.
  - In this case, each of the exiting paths must be labeled with a guard condition.
    - A guard condition represents the value of the test for that particular path to be executed.
- The merge node
- used to bring back together multiple mutually exclusive paths that have been split based on an earlier decision

20

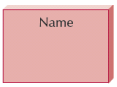
Control Nodes

- The fork and join nodes allow parallel and concurrent processes to be modeled
- The fork node
- used to split the behavior of the business process into multiple parallel or concurrent flows.
- Unlike the decision node, the paths are not mutually exclusive
  - i.e., both paths are executed concurrently.
- The join node
- brings back together the separate parallel or concurrent flows in the business process into a single flow.

21

Swimlanes

- **A Swimlane:**
  - used to break up an activity diagram into rows and columns to assign the individual activities/actions to the individuals or objects that are responsible for executing the activity/action
  - labeled with the name of the individual or object responsible



22

Swimlanes

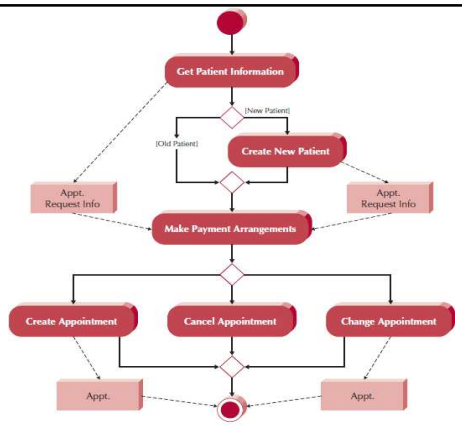
- As already described, activity diagrams can be useful in modeling a business process independent of any object implementation.
- However, there are times that it is useful to break up an activity diagram in a manner that is useful in assigning responsibility to objects or individuals that would actually perform the activity.
  - This is especially useful when modeling a business workflow.
- This is accomplished through the use of swimlanes.

23

Activity Diagram Example

- Following figure presents a simple activity diagram that represents part of an appointment system for a doctor's office.
- Activity Diagram for Appointment System:

24



25

Creating Activity Diagrams

1. Since an activity diagram can be used to model any kind of process, you should set the context or scope of the activity being modeled.
   - Once you have determined the scope, you should give the diagram an appropriate title.
2. You must identify the activities, control flows, and object flows that occur between the activities.
3. You should identify any decisions that are part of the process being modeled.
4. You should attempt to identify any prospects for parallelism in the process.
5. You should draw the activity diagram.

26

Creating Activity Diagrams

- When drawing an activity diagram, you should limit the diagram to a single initial node that starts the process being modeled.
  - This node should be placed at the top or top-left of the diagram depending on the complexity of the diagram.
- For most business processes, there should only be a single final-activity node.
  - This node should be placed at the bottom or bottom-right of the diagram.
- Since most high-level business processes are sequential, not parallel, the use of a final-flow node should be limited.

27

Creating Activity Diagrams

- When modeling high-level business processes or workflows, you should only include the more important decisions in the activity diagrams.
  - In those cases, be sure that the guard conditions associated with the outflows of the decision nodes are mutually exclusive.
- The outflows and guard conditions should form a complete set
  - i.e., all potential values of the decision are associated with one of the flows
- As in decision modeling, you should only include forks and joins to represent the more important parallel activities in the process.

28

Creating Activity Diagrams

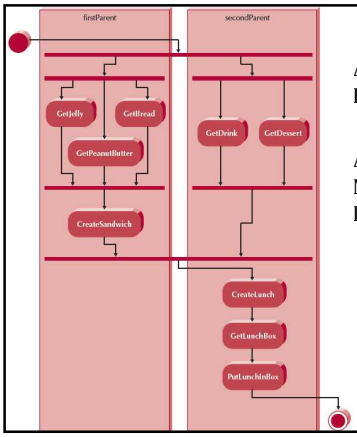
- When laying out the activity diagram, you should attempt to minimize line crossings to enhance the readability of the diagram.
- You also should lay out the activities on the diagram in a left to right and/or top to bottom order based on the order that the activities are executed.
- Swimlanes should only be used to simplify the understanding of an activity diagram.
  - the swimlanes should enhance the readability of the diagram.
- Also, when there are object flows among the activities associated with the different individuals (swimlanes) executing the activities of the process, it is useful to show the actual object flowing from one individual to another individual by including an object node "between" the two individuals (i.e., between the two swimlanes).

29

Creating Activity Diagrams

- Finally, you should challenge any activity that does not have any outflows or any inflows.
- Activities with no outflows are referred to as black-hole activities.
- If the activity is truly an end point in the diagram, the activity should have a control flow from it to a final activity or final-flow node.
- An activity that does not have any inflow is known as a miracle activity.
  - In this case, the activity is either missing an inflow from the initial node of the diagram or from another activity.

30



Activity Diagram Example:

Activity Diagram for Making a School Box Lunch

31

USE-CASE DESCRIPTIONS

A use case illustrates the activities that are performed by users of a system.

Use cases are logical models

they describe the activities of a system without specifying how the activities are implemented.

What are Use-Case Descriptions?

- Use cases are simple descriptions of a system's functions from the bird's-eye view of the users.
- Use case diagrams are functional diagrams in that they portray the basic functions of the system
  - what the users can do and how the system should respond to the user's actions.
- Creating use case diagrams is a two-step process
  - the users work with the project team to write text-based use case descriptions,
  - the project team translates the use case descriptions into formal use case diagrams.
- Both are based on the identified requirements and the activity diagram description of the business process.

33

What are Use-Case Descriptions?

- Use case descriptions contain all the information needed to produce use case diagrams.
- Although it is possible to skip the use case description step and move directly to creating use case diagrams and the other diagrams that follow, users often have difficulty describing their business processes using only use case diagrams.
- Through the creation of use case descriptions, users can describe the required details of each individual use case.
- As to which should come first—use case descriptions or use case diagram—technically speaking, it really does not matter.
  - Both should be done to fully describe the requirements that the IS must meet.

34

What are Use-Case Descriptions?

- Use cases are the primary drivers for all of the UML diagramming techniques.
- The use case communicates at a high level what the system needs to do, and all of the UML diagramming techniques build on this by presenting the use case functionality in a different way for a different purpose.
- Use cases are the building blocks by which the system is designed and built.

35

What are Use-Case Descriptions?

- Use cases capture the typical interaction of the system with the system's users (end users and other systems).
- These interactions represent the external, or functional, view of the system from the perspective of the user.
- Each use case describes one and only one function in which users interact with the system,
  - although a use case may contain several "paths" that a user can take while interacting with the system.
- Each path through the use case is referred to as a scenario.

36

What are Use-Case Descriptions?

- When creating use cases, the project team must work closely with the users to gather the requirements needed for the use cases.
  - This is often done through interviews, JAD sessions, and observation.
- A good place to look for potential use cases is the activity diagram representation of the business process.
  - In many cases, the activities identified in the activity diagram will become the use cases in the business process being modeled.
- The key thing to remember is that each use case is associated with one and only one role that users have in the system.
- It is possible that multiple users will play the same role.
  - As such, use cases should be associated with the roles "played" by the users and not with the users themselves.

37

How Are Use-Cases Created?

- Two steps:
  - Write text-based case descriptions
  - Translate descriptions into diagrams
- Describes one and only one function, but may have multiple paths.
- Developed working with users for content.

38

Types of Use Case

- There are many different types of use cases.
- But classification is done based on the purpose of the use case and the amount of information that the use case contains:
  - overview versus detail
  - essential versus real

39

overview versus detail

- An overview use case
  - used to enable the analyst and user to agree on a high-level overview of the requirements.
    - Typically, they are created very early in the process of understanding the system requirements, and they only document basic information about the use case such as its name, ID number, primary actor, type, and a brief description.
  - Once the user and the analyst agree upon a high-level overview of the requirements, the overview use cases can be converted to detail use cases.
- A detail use case
  - documents, as far as possible, all of the information needed for the use case.

40

essential versus real

- An essential use case
  - describes the minimum essential issues necessary to understand the required functionality.
- A real use case
  - describe a specific set of steps.
- For example,
    - an essential use case in a dentist office might say that the receptionist should attempt to "Match the Patient's desired appointment times with the available times," while a real use case might say that the receptionist should "Look up the available dates on the calendar using MS Exchange to determine if the requested appointment times were available."
- The primary difference:
  - essential use cases are implementation independent,
  - real use cases are detailed descriptions of how to use the system once it is implemented.
    - tend to be used only in detailed design, implementation, and testing.

41

Elements of a Use-Case Description

- A use case description contains all the information needed to build the diagrams that follow,
- but it expresses it in a less formal way that is usually simpler for users to understand.
- There are three basic parts to a use case description:
  - overview information,
  - relationships,
  - the flow of events.
- Following figure shows a sample use case description.

42

Elements of a Use-Case Description

Use Case Name:	ID:	Importance Level:
Primary Actor:		Use Case Type:
Stakeholders and Interests:		
Brief Description:		
Trigger:		
Relationships: (Association, Include, Extend, Generalization)		
Normal Flow of Events:		
Subflows:		
Alternate/Exceptional Flows:		

43

Use Case Description Example…

Use Case Name: Make appointment	ID: 2	Importance Level: High
Primary Actor: Patient	Use Case Type: Detail, essential	

Stakeholders and Interests:
Patient - wants to make, change, or cancel an appointment
Doctor - wants to ensure patient's needs are met in a timely manner

Brief Description: This use case describes how we make an appointment as well as changing or canceling an appointment.

Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment.

Type: External

Relationships:
Association: Patient
Include: Make Payment Arrangements
Extend: Create New Patient
Generalization:

Normal Flow of Events:
1. The Patient contacts the office regarding an appointment.
2. The Patient provides the Receptionist with their name and address.
3. The Receptionist validates that the Patient exists in the Patient database.
4. The Receptionist executes the Make Payment Arrangements use case.
5. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment, or change an existing appointment.

44

…Use Case Description Example

If the patient wants to make a new appointment,
the S-1: new appointment subflow is performed.
If the patient wants to cancel an existing appointment,
the S-2: cancel appointment subflow is performed.
If the patient wants to change an existing appointment,
the S-3: change appointment subflow is performed.
6. The Receptionist provides the results of the transaction to the Patient.

Subflows:
S-1: New Appointment
1. The Receptionist asks the Patient for possible appointment times.
2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment.
S-2: Cancel Appointment
1. The Receptionist asks the Patient for the old appointment time.
2. The Receptionist finds the current appointment in the appointment file and cancels it.
S-3: Change Appointment
1. The Receptionist performs the S-2: cancel appointment subflow.
2. The Receptionist performs the S-1: new appointment subflow.

Alternate/Exceptional Flows:
3a: The Receptionist executes the Create New Patient use case.
S-1, 2a1: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule.
S-1, 2a2: The Patient chooses one of the proposed times or decides not to make an appointment.

45

overview information

- identifies the use case and provides basic background information about the use case.
- The use case name of the use case should be a verb-noun phrase (e.g., Make Appointment).
- The use case ID number provides a unique way to find every use case and also enables the team to trace design decisions back to a specific requirement.
- The use case type is either overview or detail and essential or real.
- The primary actor is usually the trigger of the use case
  - the person or thing that starts the execution of the use case.
    - The primary purpose of the use case is to meet the goal of the primary actor.
- The brief description is typically a single sentence that describes the essence of the use case.

46

overview information

- The importance level can be used to prioritize the use cases.
  - OO development tends to follow a RAD-phased development approach, in which some parts of the system are developed first and other parts are only developed in later versions.
- The importance level enables the users to explicitly prioritize
  - which business functions are most important and need to be part of the first version of the system,
  - which are less important and can wait until later versions if necessary.
- The importance level can use a fuzzy scale, such as high, medium, and low

47

overview information

- It can also be done more formally using a weighted average of a set of criteria.
- For example, Larman suggests rating each use case over the following criteria using a scale from zero to five:
  - The use case represents an important business process.
  - The use case supports revenue generation or cost reduction.
  - Technology needed to support the use case is new or risky and therefore will require considerable research.
  - Functionality described in the use case is complex, risky, and/or time critical. Depending on a use case's complexity, it may be useful to consider splitting its implementation over several different versions.
  - The use case could increase understanding of the evolving design relative to the effort expended.

Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*

48

overview information

- A use case may have multiple stakeholders that have an interest in the use case.
  - As such, each use case lists each of the stakeholders with their interest in the use case (e.g., Patient and Doctor).
  - The stakeholders' list always includes the primary actor (e.g., Patient).
- Each use case typically has a trigger
  - the event that causes the use case to begin.
    - For example,"Patient calls and asks for a new appointment or asks to cancel or change an existing appointment."
- A trigger can be
  - an external trigger,
    - such as a customer placing an order or the fire alarm ringing,
  - a temporal trigger,
    - such as a book being overdue at the library, or time to pay the rent.

49

Relationships

- explain how the use case is related to other use cases and users.
- There are four basic types of relationships:
  - association,
  - extend,
  - include,
  - generalization.
- An association relationship
- documents the communication that takes place between the use case and the actors that use the use case.
  - An actor is the UML representation for the role that a user plays in the use case.
    - For example, in previous figure, the Make Appointment use case is associated with the actor, Patient. In this case, a patient makes an appointment.
- All actors involved in the use case are documented with the association relationship

50

Relationships

- An extend relationship
  - represents the extension of the functionality of the use case to incorporate optional behavior.
- An include relationship
  - represents the mandatory inclusion of another use case.
  - enables functional decomposition
    - the breaking up of a complex use case into several simpler ones.
- The generalization relationship
  - allows use cases to support inheritance.

51

Flow of Events

- Finally, the individual steps within the business process are described.
- There are three different categories of steps or flows that can be documented:
  - normal flow of events,
  - subflows,
  - alternate or exceptional flows:
- The normal flow of events
  - includes only those steps that normally are executed in a use case.
  - The steps are listed in the order in which they are performed.

52

Flow of Events

- In some cases, it is recommended to decompose the normal flow of events into a set of subflows
  - to keep the normal flow of events as simple as possible.
- Each of the steps of the subflows is listed.
- These subflows are based on the control flow logic in the activity diagram representation of the business process.
- Alternate or exceptional flows are ones that do happen but are not considered to be the norm. These must be documented.
- The primary purpose of separating out alternate or exceptional flows is to keep the Normal Flow of Events as simple as possible.

53

Optional Characteristics

- There are other characteristics of use cases that could be documented. These include :
  - the level of complexity of the use case,
  - the estimated amount of time it takes to execute the use case,
  - the system with which the use case is associated,
  - specific data flows between the primary actor and the use case,
  - any specific attribute, constraint, or operation associated with the use case,
  - any preconditions that must be satisfied for the use case to execute,
  - any guarantees that can be made based on the execution of the use case.
- There is no standard set of characteristics of a use case that must be captured.

54

Guidelines for Creating Use-Case Descriptions

- Write each set in the form of Subject-Verb-Direct Object (and sometimes Preposition-Indirect Object).
- Make sure it is clear who the initiator of the step is.
- Write the steps from the perspective of the independent observer.
- Write each step at about the same level of abstraction.
- Ensure the use case has a sensible set of steps.
- Apply the KISS principle liberally.
- Write repeating instructions after the set of steps to be repeated.

55

Your Turn

- How would you make requirements gathering (interviews, questionnaires, observation, and document analysis) more effective by knowing that eventually you will be creating use-case descriptions and diagrams?

56

USE-CASE DIAGRAMS

summarizes all of the use cases for the part of the system being modeled together in one picture.

illustrates in a very simple way the main functions of the system and the different kinds of users that will interact with it.

An analyst can use the use case diagram to better understand the functionality of the system at a very high level.

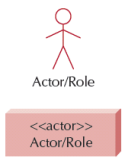
Elements of a use case diagram

- The use case diagram is drawn early on in the SDLC when gathering and defining requirements for the system
  - because it provides a simple, straightforward way of communicating to the users exactly what the system will do.
- In this manner, the use case diagram can encourage the users to provide additional requirements that the written use case may not uncover.
- Next the elements of a use case diagram will be described

58

Actor

- **An Actor:**
- a person or system that derives benefit from and is external to the subject
- depicted as either a stick figure (default) or if a non-human actor is involved, as a rectangle with <<actor>> in it
- labeled with its role
- can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead
- Are placed outside the subject boundary



59

Association

- Use cases are connected to actors through association relationships,
  - show with which use cases the actors interact
- A line drawn from an actor to a use case depicts an association.
- The association typically represents two-way communication between the use case and the actor.
- If the communication is only one way, then a solid arrowhead can be used to designate the direction of the flow of information.
- It is possible to represent the multiplicity of the association
  - "*"

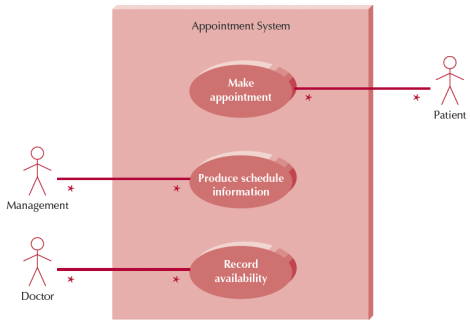
60

Use case

- **A Use Case:**
- represents a major piece of system functionality
- can extend another use case
- can include another use case
- placed inside the system boundary
- labeled with a descriptive verb-noun phrase

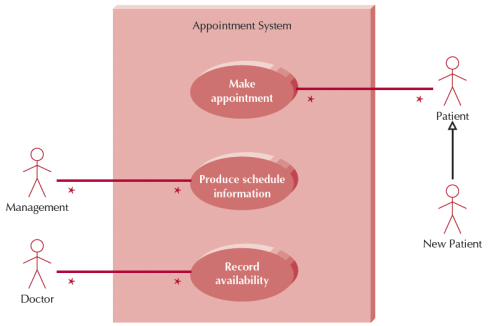
61

Use Case Diagram for Appointment System



62

Use Case Diagram with Specialized Actor



63

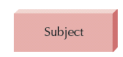
Use case

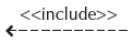
- There are times when a use case
  - includes,
  - extends,
  - generalizes

  the functionality of another use case on the diagram.
- These are shown using
  - include relationships,
  - extend relationships,
  - generalization relationships.
- To increase the ease of understanding a use case diagram, "higher-level" use cases normally are drawn above the "lower-level" ones.

64

- **A Subject Boundary:**
  - Includes the name of the subject inside or on top
  - Represents the scope of the subject,
    - e.g., a system or an individual business process
- **An Association Relationship:**
  - Links an actor with the use case(s) with which it interacts
- **An Include Relationship:**
  - Represents the inclusion of the functionality of one use case within another
  - The arrow is drawn from the base use case to the included use case



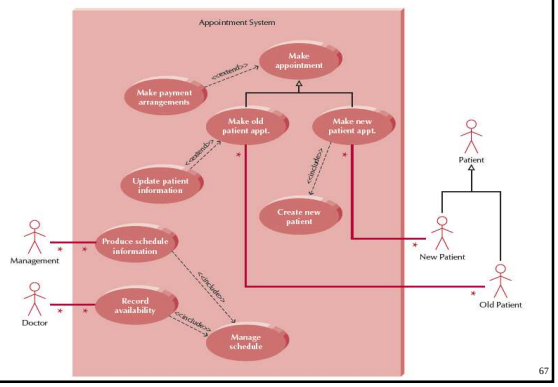


65

- **An Extend Relationship:**
  - Represents the extension of the use case to include optional behavior
  - The arrow is drawn from the extension use case to the base use case
- **A Generalization Relationship:**
  - Represents a specialized use case to a more generalized one
  - The arrow is drawn from the specialized use case to the base use case

66

Extend and Include Relationships



67

Subject Boundary

- The use cases are enclosed within a subject boundary,
  - a box that defines the scope of the system and clearly delineates what parts of the diagram are external or internal to it
- One of the more difficult decisions to make is where to draw the subject boundary.
- A subject boundary can be used to separate
  - a software system from its environment,
  - a subsystem from other subsystems within the software system,
  - an individual process in a software system.
- They also can be used to separate an information system, including both software and internal actors, from its environment.
  - As such, care should be taken to carefully decide on what the scope of the information system is to include.

68

CREATING USE-CASE DESCRIPTIONS AND USE-CASE DIAGRAMS

Use cases are used to describe the functionality of the system and as a model of the dialog between the actors and the system.

As such, they tend to be used to model both the contexts of the system and the detailed requirements for the system.

Even though the primary purpose of use cases is to document the functional requirements of the system, they also are used as a basis for testing the evolving system.

Major Steps in Writing Use-Cases

- The most effective process has thirteen steps to create use case descriptions and four additional steps to create use case diagrams .

- Identify the major use-cases
- Expand the major use-case
- Confirm the major use-cases
- Create the use-case diagram

70

Identifying the Major Use-Cases

- Review the activity diagram.
- Find the subject's boundaries.
- Identify the primary actors and their goals.
- Identify and write the overviews of the major use cases for the above.
- Carefully review the current use cases. Revise as needed.

71

Expand the Major Use-Cases

- Choose one of the use cases to expand.
- Start filling in the details of the chosen use case.
- Write the Normal Flow of Events of the use case.
- If the Normal Flow of Events is too complex or long, decompose into subflows.
- List the possible alternate or exceptional flows.
- For each alternate or exceptional flow, list how the actor and/or system should react.

72

Confirm the Major Use Cases

- Review the current set
  - Consider semantics and syntax
  - Helpful to involve the users
- Iterate the entire set of steps until all use cases are defined

73

Create the Use-Case Diagram

- Start with system boundary
- Place elements in order to be easy to read
- Place actors on the diagram
- Conclude by connecting actors to use cases by lines

74

Writing Effective Use-Case Descriptions

Identify the Major Use Cases

1. Review the activity diagram.
2. Find the subject's boundaries.
3. Identify the primary actors and their goals.
4. Identify and write the overviews of the major use cases for the above.
5. Carefully review the current use cases. Revise as needed.

Expand the Major Use Cases

6. Choose one of the use cases to expand.
7. Start filling in the details of the chosen use case.
8. Write the Normal Flow of Events of the use case.
9. If the Normal Flow of Events is too complex or long, decompose into subflows.
10. List the possible alternate or exceptional flows.
11. For each alternate or exceptional flow, list how the actor and/or system should react.

Confirm the Major Use Cases

12. Carefully review the current set of use cases. Revise as needed.
13. Start at the top again.

Create the Use Case Diagram

1. Draw the subject boundary.
2. Place the use cases on the diagram.
3. Place the actors on the diagram.
4. Draw the associations.

75

Selecting the Appropriate Techniques

	Interviews	JAD	Questionnaires	Document Analysis	Observation
Type of Information	As-Is Improve. To-Be	As-Is Improve. To-Be	As-Is Improve.	As-Is	As-Is
Depth of Information	High	High	Medium	Low	Low
Breadth of Information	Low	Medium	High	High	Low
Integration of Info.	Low	High	Low	Low	Low
User Involvement	Medium	High	Low	Low	Low
Cost	Medium	Low-Medium	Low	Low	Low-Medium

76

Refining Project Size with Case Points

- Create essential use cases and use case diagram
- Determine Unadjusted Actor Weighting Table
- Obtain Unadjusted Use Case Weight Total
- Compute value of Unadjusted Use Case Points

77

Your Turn

- Create a set of use cases for campus housing. Consider the steps in registering for campus housing, in being assigned to a particular unit, to being matched with roommates, to moving in.

78

Expanding the Domain

- Additional resources regarding use-cases and many other object-oriented development topics can be found at:
- http://www.omg.org

79

Summary

- Use-case descriptions are the basis for further analysis and design.
- They are created based on 7 guidelines and 13 steps.
- Use-case diagrams present a graphical overview of the main functionality of a system.

80