USE CASE DIAGRAMS

Use Cases

- What is a Use Case
  - 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
  - A scenario-based technique in the UML
  - A sequence of actions a system performs that yields a valuable result for a particular actor.

Use Case Analysis

- What is an Actor?
  - A user that interacts with the system being designed in order to obtain some value from that interaction
- Use Cases describe scenarios between users of the system (the actor) and the system itself.

Use Cases

- **Use case diagrams** describe what a system does from the standpoint of an external observer. The emphasis is on *what* a system does rather than *how.*

- Use case diagrams are closely connected to scenarios. A **scenario** is an example of what happens when someone interacts with the system.

Use Cases

- Here is a scenario for a 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. "*

- We want to write a use case for this scenario.
- Remember: A **use case** is a summary of scenarios for a single task or goal.

Use Cases

- Step 1 Identify the actors
- As we read the scenario, define those people or systems that are going to interact with the scenario.

- 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. "*

Questions for Identifying People Actors

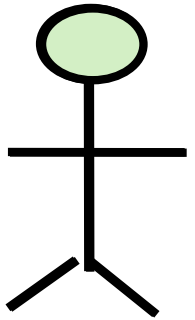
- Who is interested in the scenario/system?
- Where in the organization is the scenario/system be used?
- Who will benefit from the use of the scenario/system?
- Who will supply the scenario/system with this information, use this information, and remove this information?
- Does one person play several different roles?
- Do several people play the same role?

Questions for Identifying Other Actors

- What other entity is interested in the scenario/system?
- What other entity will supply the scenario/system with this information, use this information, and remove this information?
- Does the system use an external resource?
- Does the system interact with a legacy system?

Actors

- An Actor is outside or external the system.
- It can be a:
  - Human
  - Peripheral device (hardware)
  - External system or subsystem
  - Time or time-based event
- Represented by stick figure



Use Cases

- A **use case** is a summary of scenarios for a single task or goal.
- An **actor** is who or what initiates the events involved in the task of the use case. Actors are simply roles that people or objects play.
- So as we read our scenario, what or who is the actor????

Use Cases

- So as we read our scenario, what or who is the actor????

- *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. "*

- The actor is a **Patient**.

Use Cases

- The **use case** is a summary of scenarios for a single task or goal.

- So What is the Use Case????

- The Use Case is **Make Appointment.**
- It is a use case for the medical clinic.

Use Cases

- The picture below is a **Make Appointment** use case for the medical clinic.
- The actor is a **Patient**. The connection between actor and use case is a **communication association** (or **communication** for short).

Actors are stick figures. Use cases are ovals. Communications are lines that link actors to use cases.

Use Case Componentss

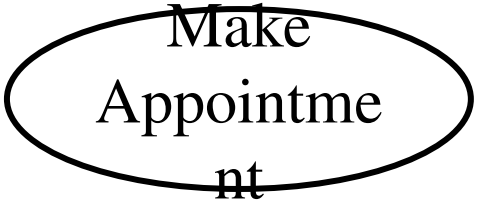
- The use case has three components.
- The **use case** task referred to as the use case that represents a feature needed in a software system.
- The **actor(s)** who trigger the use case to activate.
- The **communication** line to show how the actors

Use Case

- Each use case in a use case diagram describes one and only one *function* in which users interact with the system
  - May contain several “paths” that a user can take while interacting with the system
  - Each path is referred to as a scenario

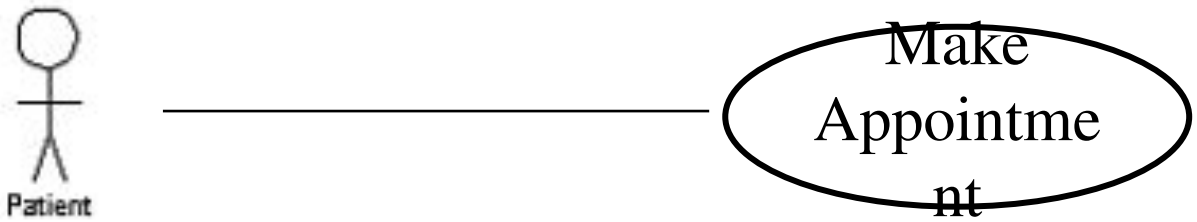
Use Case

- Labelled using a descriptive verb-noun phrase
- Represented by an oval



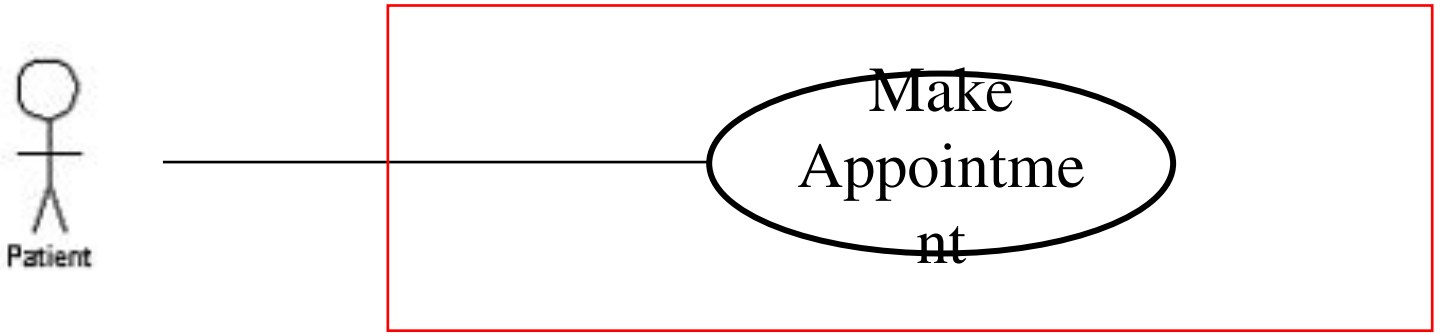
Use Case - Relationships

- Relationships
  - Represent communication between actor and use case
  - Depicted by line
  - Also called association relationship

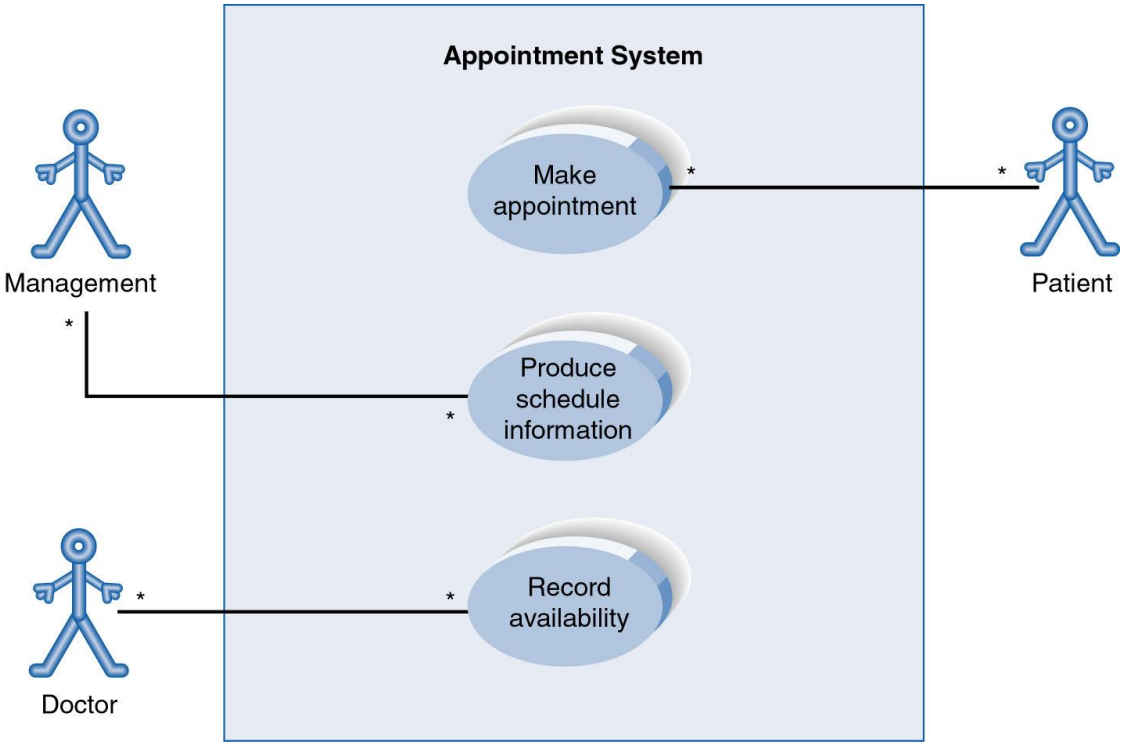


Use Case - Relationships

- Boundary
  - A boundary rectangle is placed around the perimeter of the system to show how the actors communicate with the system.



Use-Case Diagram

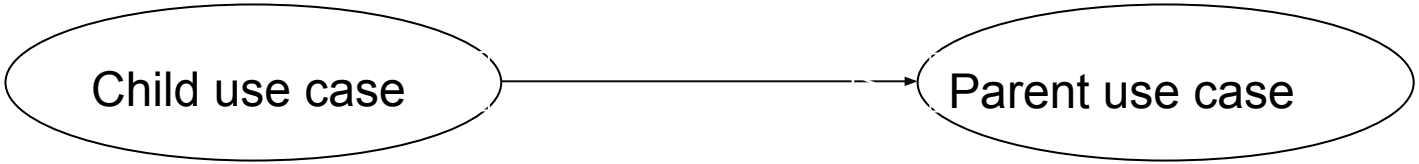


Use Case Diagram

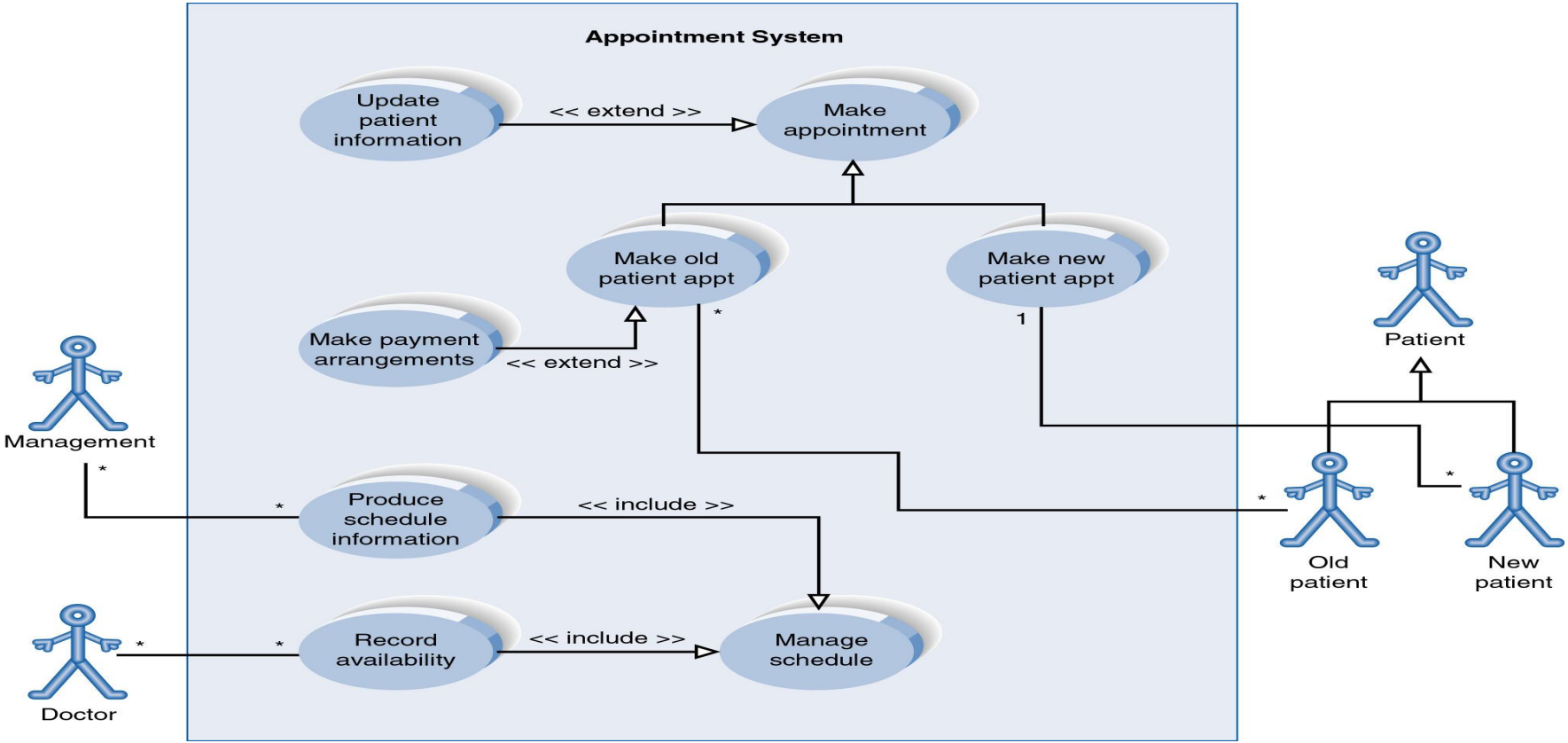
- Other Types of Relationships for Use Cases
  - Generalization
  - Include
  - Extend

Components of Use Case Diagram

- Generalization Relationship
  - Represented by a line and a hollow arrow
    - From child to parent



Example of Relationships



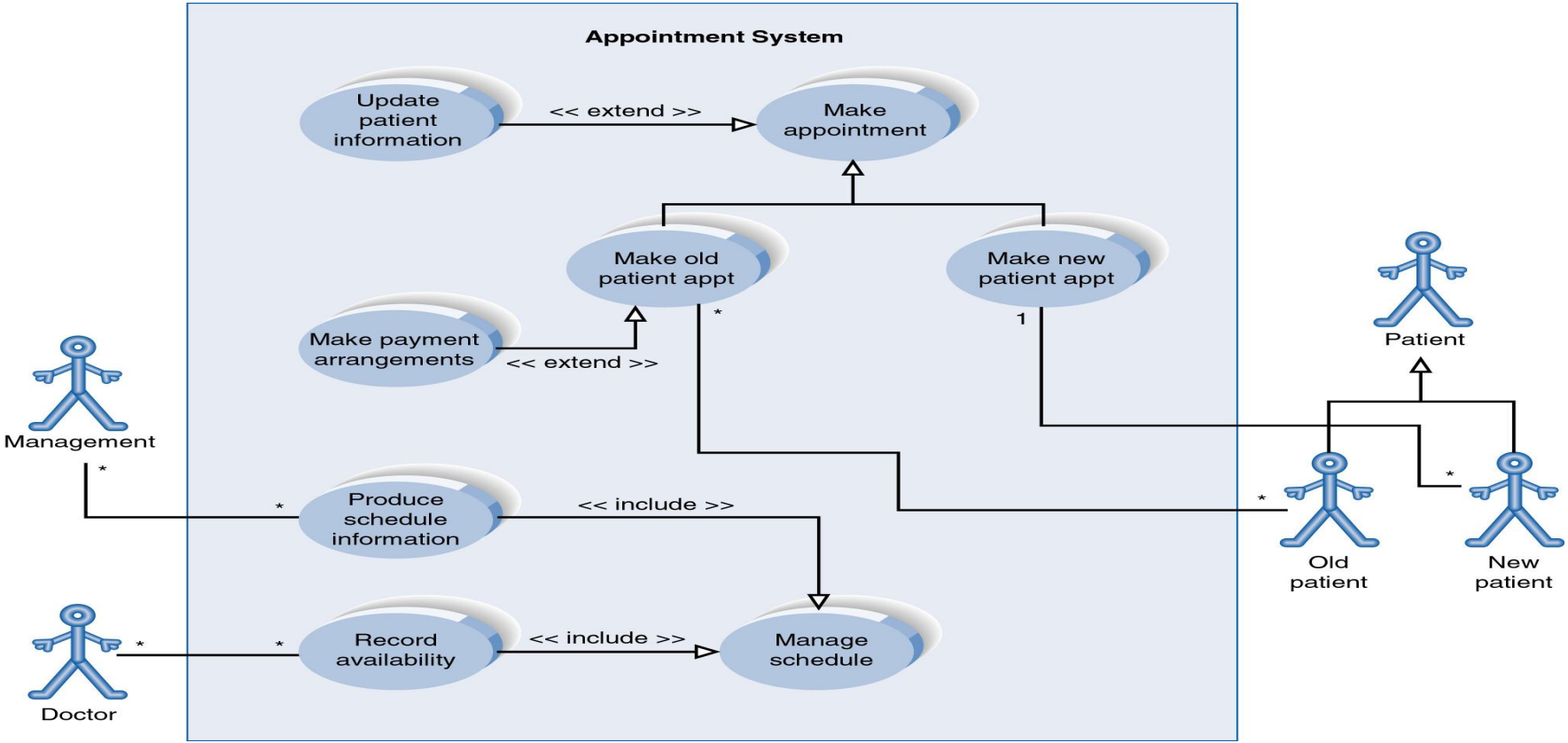
Use Case Diagram

- Include Relationship
  - Represents the inclusion of the functionality of one use case within another
  - Arrow is drawn from the base use case to the used use case
  - Write << include >> above arrowhead line

Use Case Diagram

- Extend relationship
  - Represents the extension of the use case to include optional functionality
  - Arrow is drawn from the extension use case to the base use case
  - Write << extend >> above arrowhead line

Example of Relationships



Use Case Relationships

- Pro:
  - Reduces redundancy in use cases
  - Reduces complexity within a use case
- Con
  - May introduce complexity to use case diagram

Use Case Specification/Text

Use Case ID and Name
Actor(s)
Description
Pre-condition(s)
Post-condition(s)
Control Flow
Alternative(s)/Exception(s)
Special Requirement

Example Use Case Specification

Add Comment
Brief description: The actor adds a comment
Actors: Participant
Preconditions: The actor is a registered student
Basic flow of events: 1. The actor logs into the system 2. The system authenticates the actor and starts a session 3. The actor chooses to add a comment 4. The actor is guided by the system to fill in required information to add a comment 5. The system acknowledges that comment added 6. The actor leaves the system
Extensions: 1a. The system fails to authenticate the actor. • The system informs the actor and doesn't allow the actor to proceed 3a. The system fails to add a comment • The system informs the actor
Post-conditions: The new comment is added
Special requirements: Actor is connected to Internet and uses a Java enabled Mobile phone

Benefits of Use Cases

- Use cases are the primary vehicle for requirements capture in RUP
- Use cases are described using the language of the customer (language of the domain which is defined in the glossary)
- Use cases provide a contractual delivery process (RUP is Use Case Driven)
- Use cases provide an easily-understood communication mechanism
- When requirements are traced, they make it difficult for requirements to fall through the cracks
- Use cases provide a concise summary of what the system should do at an abstract (low modification cost) level.

Difficulties with Use Cases

- As functional decompositions, it is often difficult to make the transition from functional description to object description to class design
- Reuse at the class level can be hindered by each developer “taking a Use Case and running with it”. Since UCs do not talk about classes, developers often wind up in a vacuum during object analysis, and can often wind up doing things their own way, making reuse difficult
- Use Cases make stating non-functional requirements difficult (where do you say that X must execute at Y/sec?)
- Testing functionality is straightforward, but unit testing the particular implementations and non-functional requirements is not obvious