Use Cases

What is a Use Case?

- A formal way of representing how a business <u>system interacts</u> with its environment
- Illustrates the <u>activities</u> that are performed by the users of the system
- A scenario-based technique in the UML
- A <u>sequence of actions</u> a system performs that yields a valuable result for a particular actor.

Use Case Analysis

- What is an Actor?
- A <u>user</u> or <u>outside system</u> that interacts with the system being designed in order to obtain some value from that interaction
- Use Cases describe <u>scenarios</u> that define the interaction 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 <u>external observer</u>. The emphasis is on *what* a system does rather than *how*.

- Use case diagrams are closely connected to scenarios. A **<u>scenario</u>** 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 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

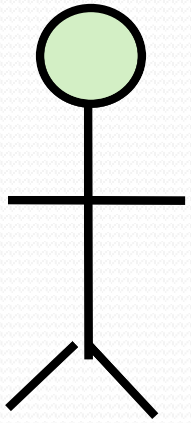
- Who is interested in the scenario/system?
- Where in the organization is the scenario/system be used?
- Who will benefit from the use of the scenario/system?
- Who will supply information to this scenario/system, use information, and remove information?
- Does one person play several different roles?
  - May have an actor for each role
- Do several people play the same role?
  - Only use one actor per role (no matter how many people play that role)

Questions for Identifying Other Actors

- What other entity is interested in the scenario/system?
- What other entity will supply info the scenario/system, 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 to 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

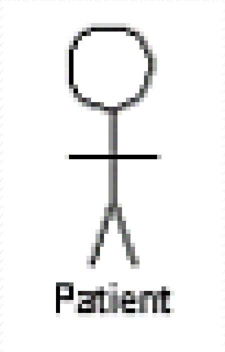
- A **use case** is a summary of scenarios for a single task or goal.
- An **actor** is who or what initiates the events involved in the task of the use case. Actors are simply roles that people or objects play.

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**.

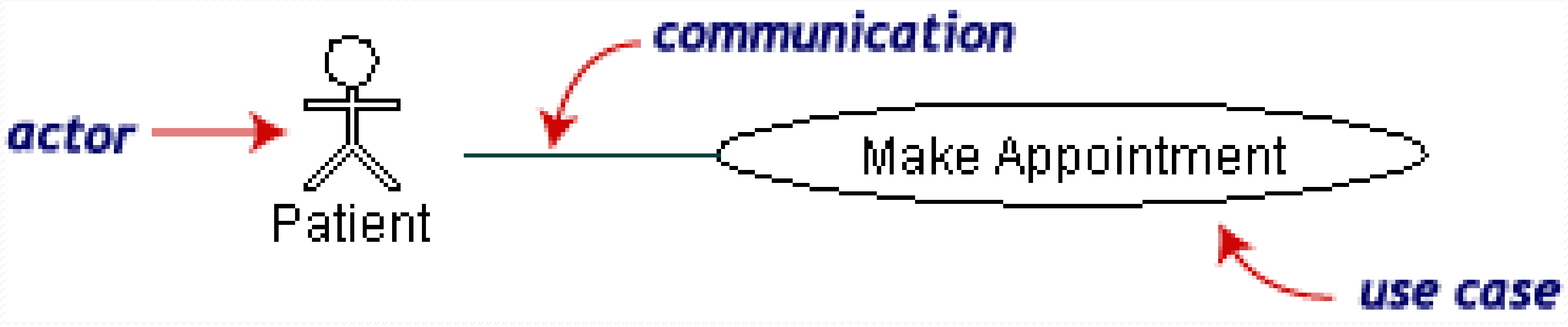


Use Cases

- 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 <u>feature</u> needed in a software system.
- The **actor(s)** who trigger the use case to activate.
- The **communication** line to show how the actors communicate with the use case.

Use Case Diagram - Use Case

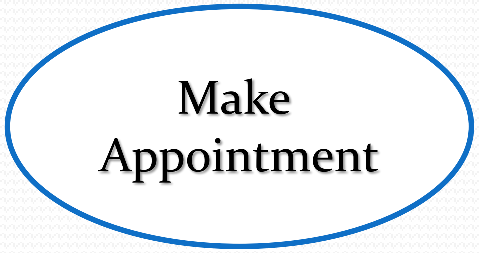
- A major process performed by the system that benefits an actor(s) in some way
- <u>Models a dialogue between an actor and the system</u>
- Represents the functionality provided by the system

Use Case

- Each use case in a use case diagram <u>describes one and only one *function* </u>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

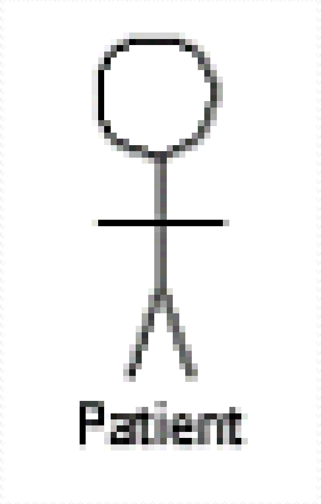
Use Case

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



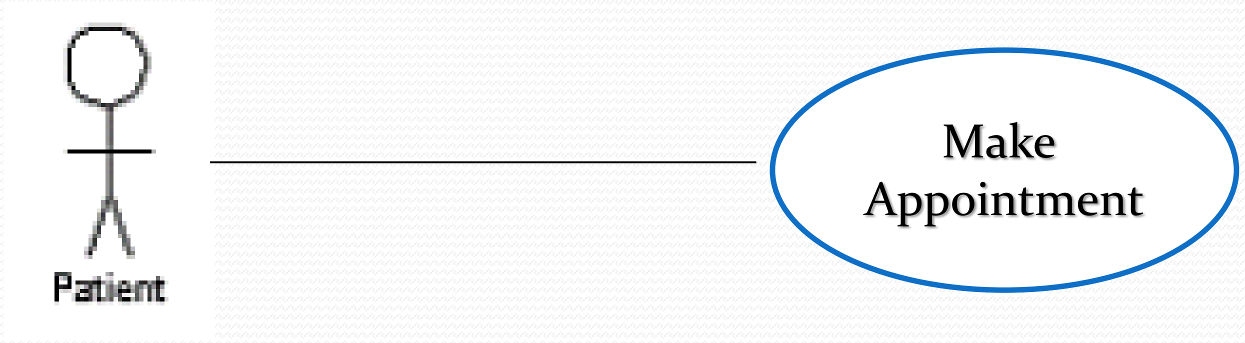
Use Case - Actor

- Labelled using a descriptive noun or phrase
- Represented by a stick character



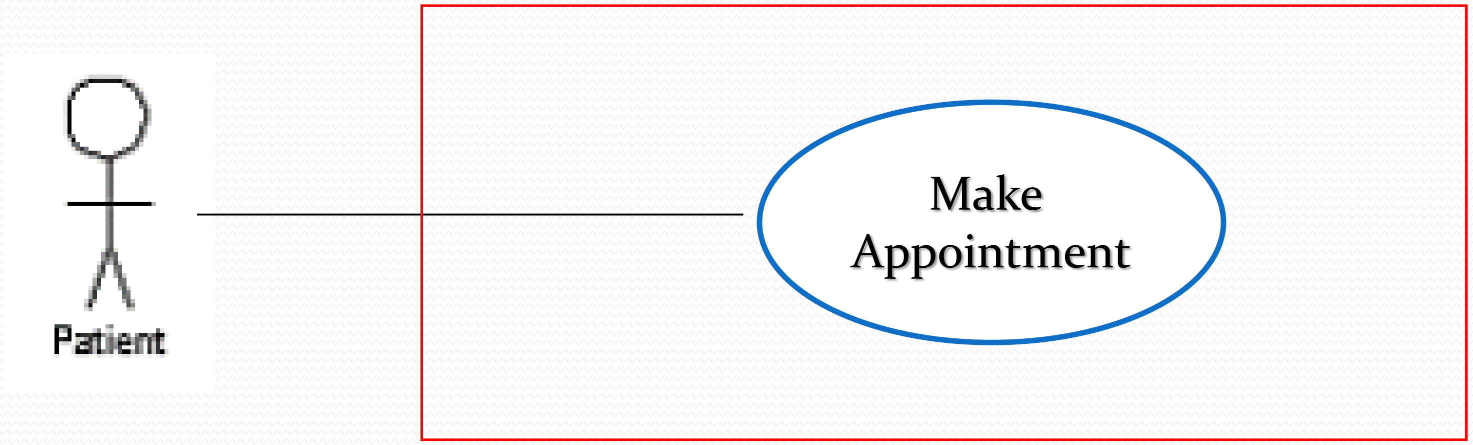
Use Case - Relationships

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

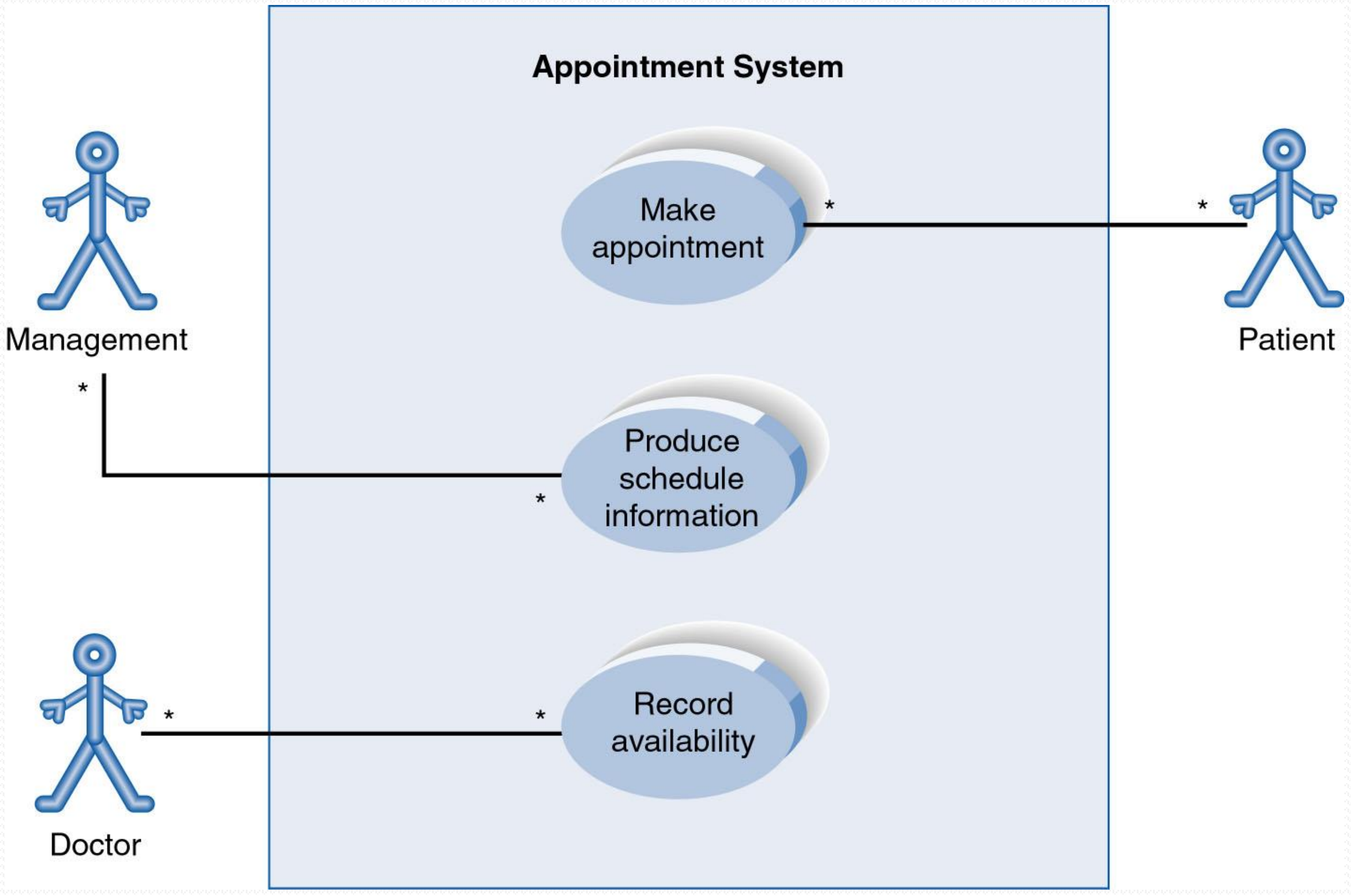


Use Case - Relationships

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



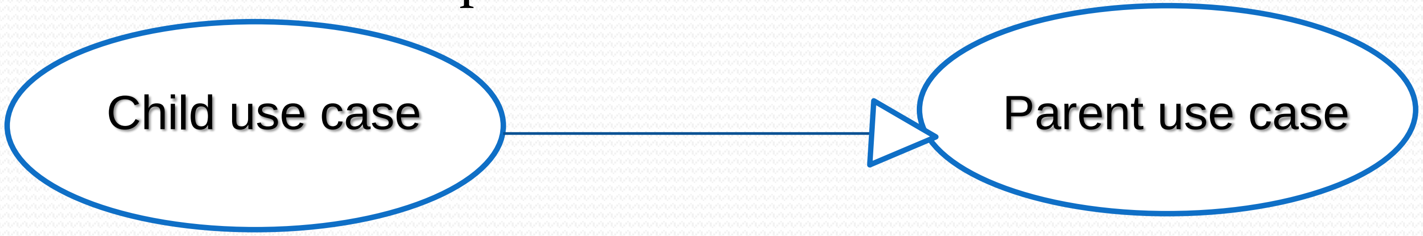
Use-Case Diagram



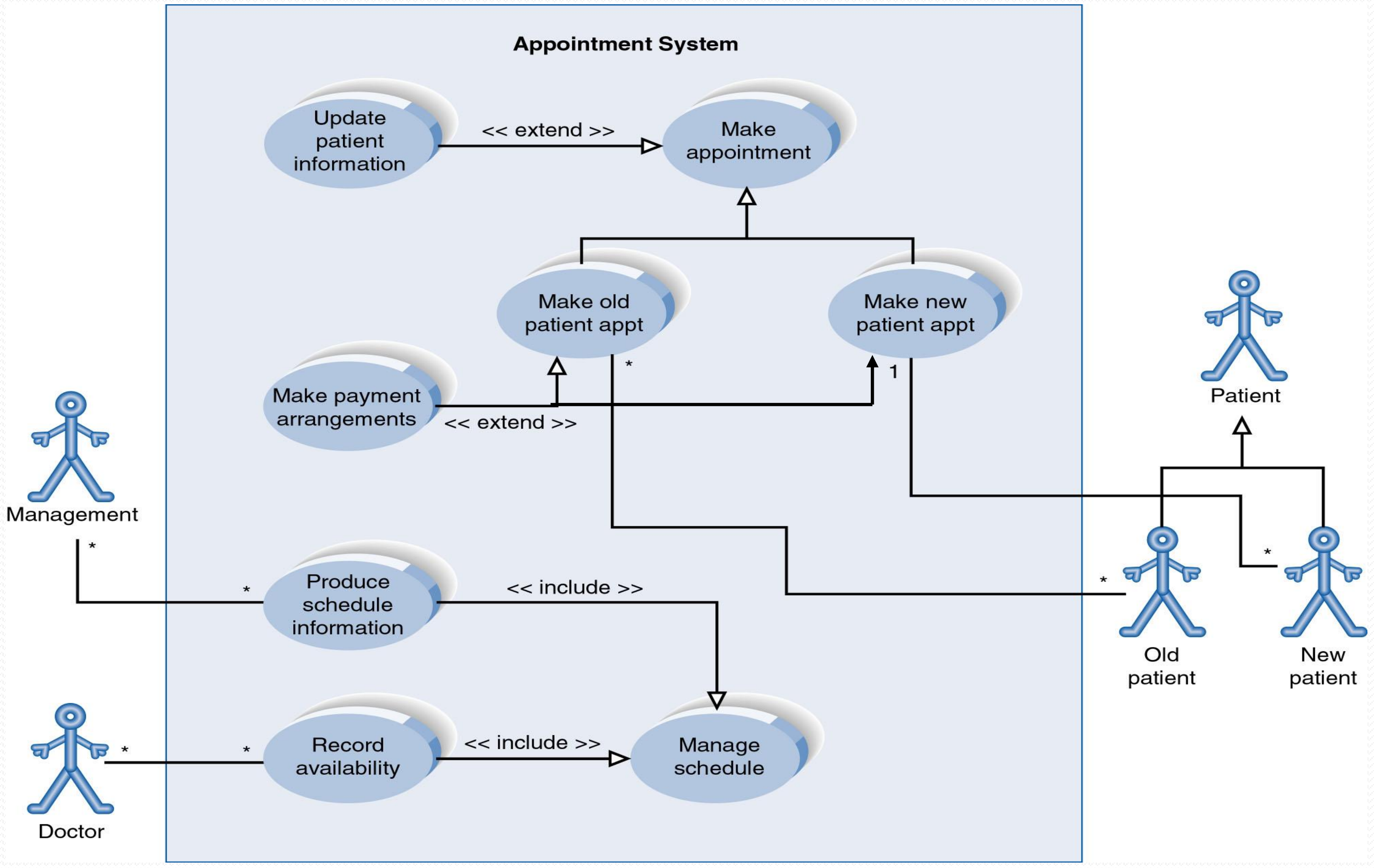
A use case diagram is a collection of actors, use cases, and their communications.

Use Case Diagram - stereotypes

- UML defines three stereotypes of association between Use Cases, «include», «extend» and generalization.
- 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

Benefits of Use Cases

- RUP's primary element in requirements capture
- Described using language of customer (domain language)
- RUP is Use Case Driven
- Easily-understood communication mechanism
- Make traceability of requirements easy.
- Provide summary of what the system should do at an abstract level.
- Easy to describe functional requirements.

Difficulties with Use Cases

- Transition from functional description to object description / class design.
- Makes <u>reuse of class </u>difficult.
  - ‘Since UCs do not talk about classes, developers often wind up doing things their own way, making reuse difficult.
- Stating <u>non-functional</u> requirements are difficult (where do you say that X must execute at Y/sec?)