(11224) INTRODUCTION TO SOFTWARE ENGINEERING

Lecture 11: Use Case Diagrams

Shereen Fouad

Use Cases

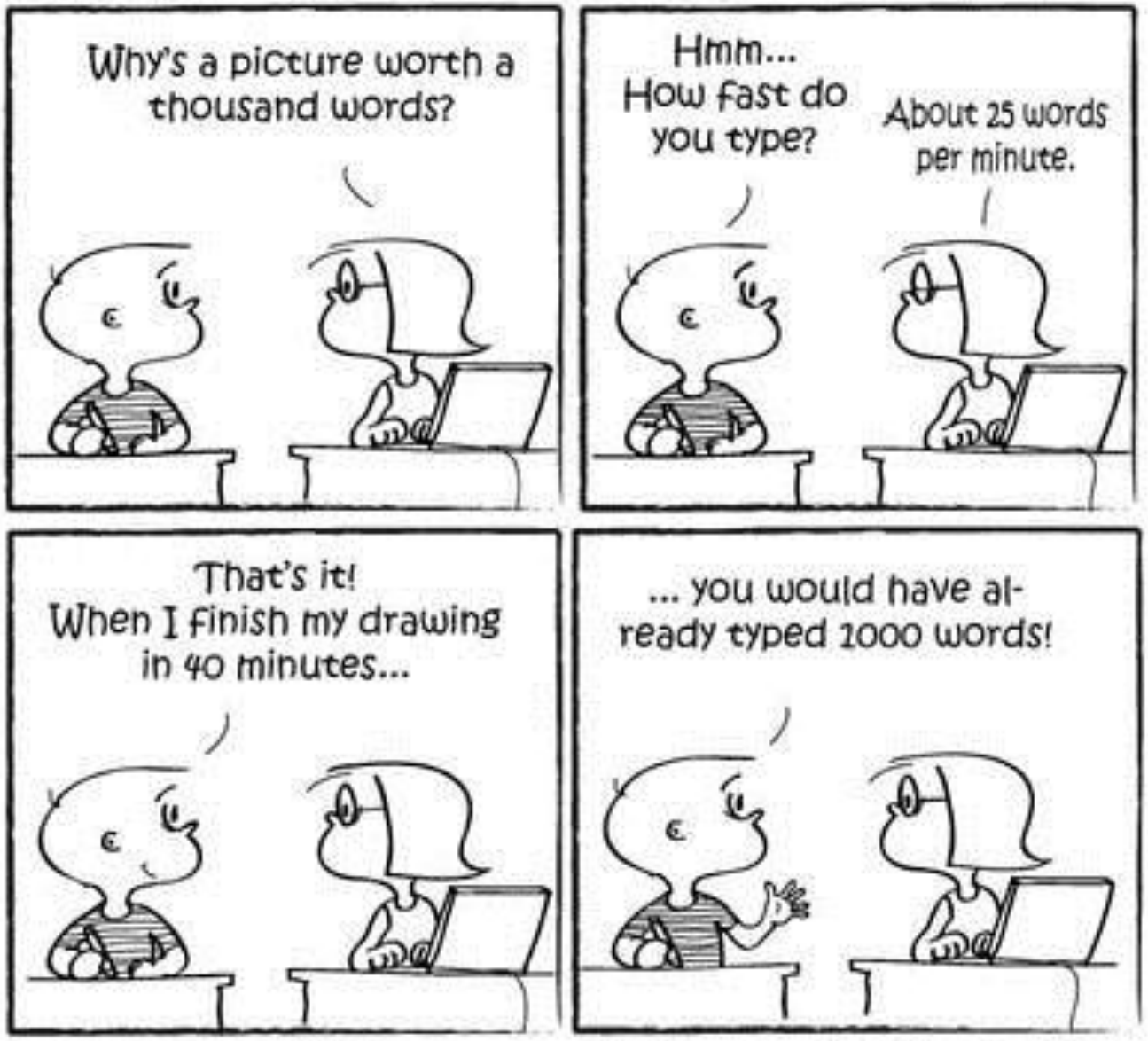
- A use case is a description of operations on a system.
- Lists of steps a system needs to follow in order o reach a goal.
- It describes the requirements of a system in terms of the interactions that should occur between the system and various people and other systems that can interact with it.
- It is basically used in the analysis phase which is a later stage of the requirement phase.
- Use cases can be represented as textual documents or in a diagram
- Today we will cover the USE CAESE diagram

```
Name         Checkout
Initiator    Customer
Goal         Collect all items currently in customer's basket, take
             payment and schedule order fulfilment.

Main Success Scenario
1.   Customer requests to checkout
2.   System obtains current basket for customer
3.   System creates new order for customer
4.   System adds order details from basket to order
5.   Include Use Case "Finalise Order and Collect Payment"
6.   System dispatches order to OrderFulfilment

Extension
3.   No basket for customer found
     1. Fail
3.   Basket for customer is empty
     1. Fail
6.   Payment collection failed
     1. Fail
```

Use Case Diagram

- UML provides a use case diagram
- A *use case diagram* is a visual representation of the relationships between actors and use cases together that documents the system’s intended behavior
- Graphical diagram does not provide any specifications/details about processes.

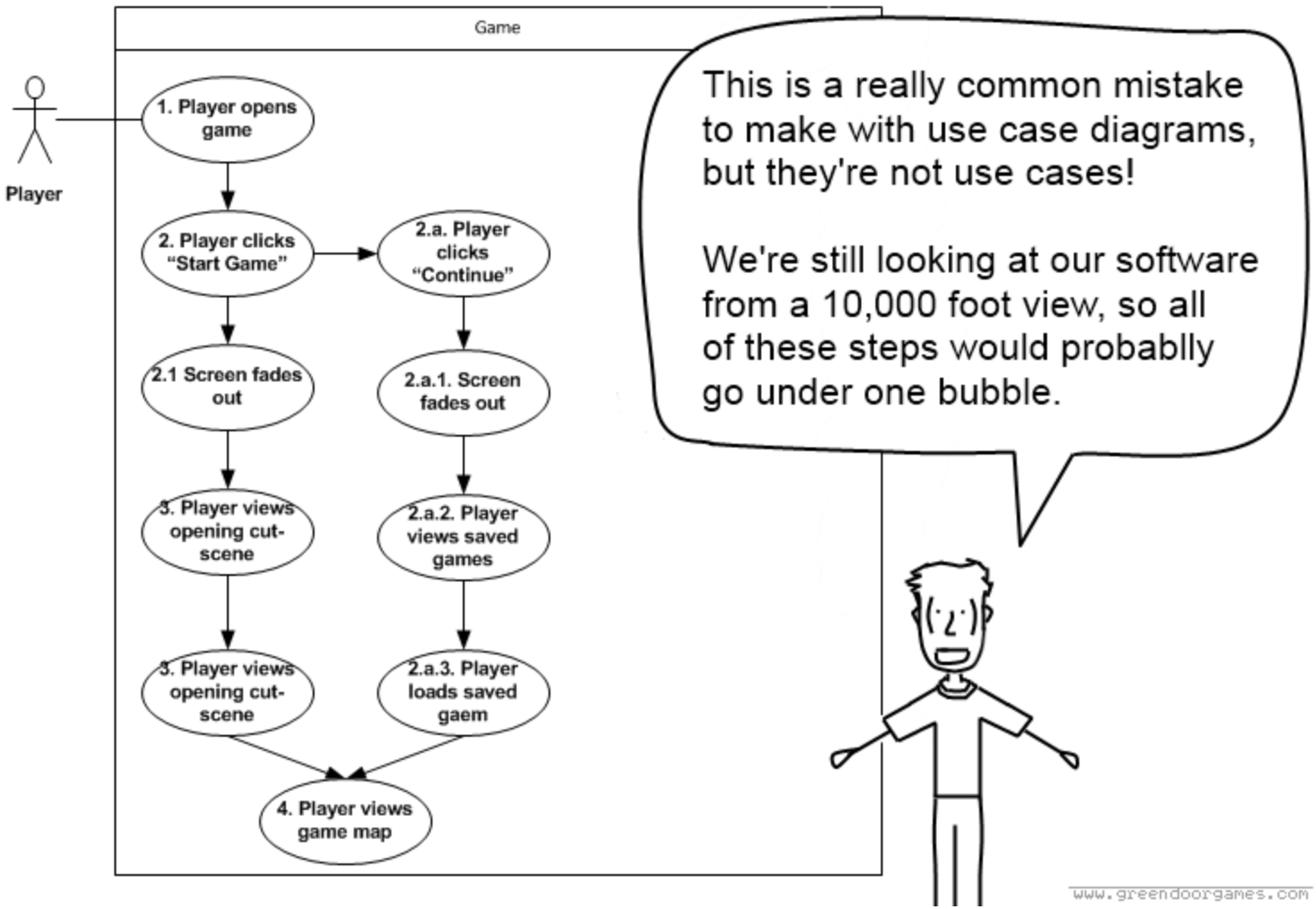
giggleBites

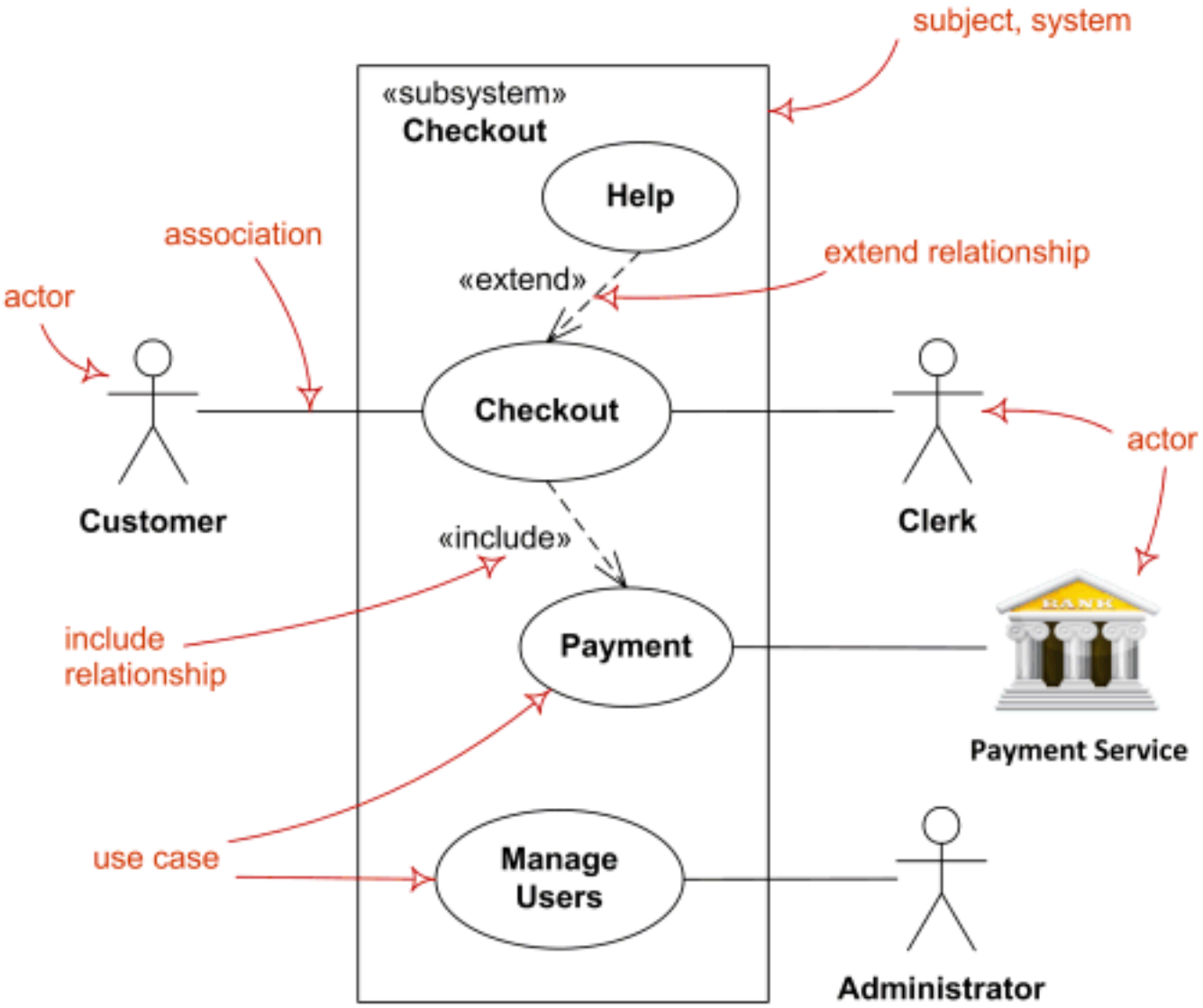


Use Case Diagram vs. Textual Use Case

- Textual Use cases, also named blue prints, are very specific and include details about processes
- They include all kinds of design decisions that programmer needs to translate to code.

- Use case diagram do not provide much information as given in the textual use cases
- Look at the system from a global perspective
- Useful only as an index to use cases in the system


Game
Player
1. Player opens game
2. Player clicks "Start Game"
2.a. Player clicks "Continue"
2.1 Screen fades out
2.a.1. Screen fades out
3. Player views opening cut-scene
2.a.2. Player views saved games
3. Player views opening cut-scene
2.a.3. Player loads saved gaem
4. Player views game map
This is a really common mistake to make with use case diagrams, but they're not use cases!
We're still looking at our software from a 10,000 foot view, so all of these steps would probablly go under one bubble.
www.greendoorgames.com


subject, system
«subsystem»
Checkout
Help
association
extend relationship
actor
«extend»
Checkout
actor
Customer
Clerk
«include»
include
relationship
Payment
Payment Service
use case
Manage
Users
Administrator

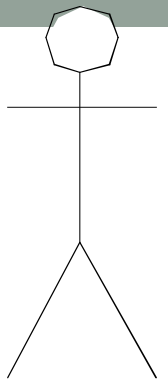
Use Case Componentss

- 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 communicate with the use case.

Use Case

- Accomplish a particular task
- 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
- Labelled using a descriptive phrase
- Represented by an oval

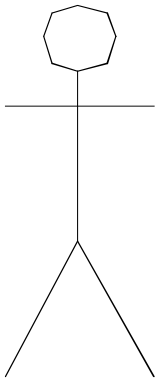
Actor



- An entity that interacts with the system (person, machine, or other)
- It is **not** part of the system itself
- Anyone or anything that must interact with the system to:
  - Input information to the system;
  - Receive information from the system; or
  - Both input information to and receive information from the system
- The name should identify the **role** or set of roles the actor plays relative to one or more use cases

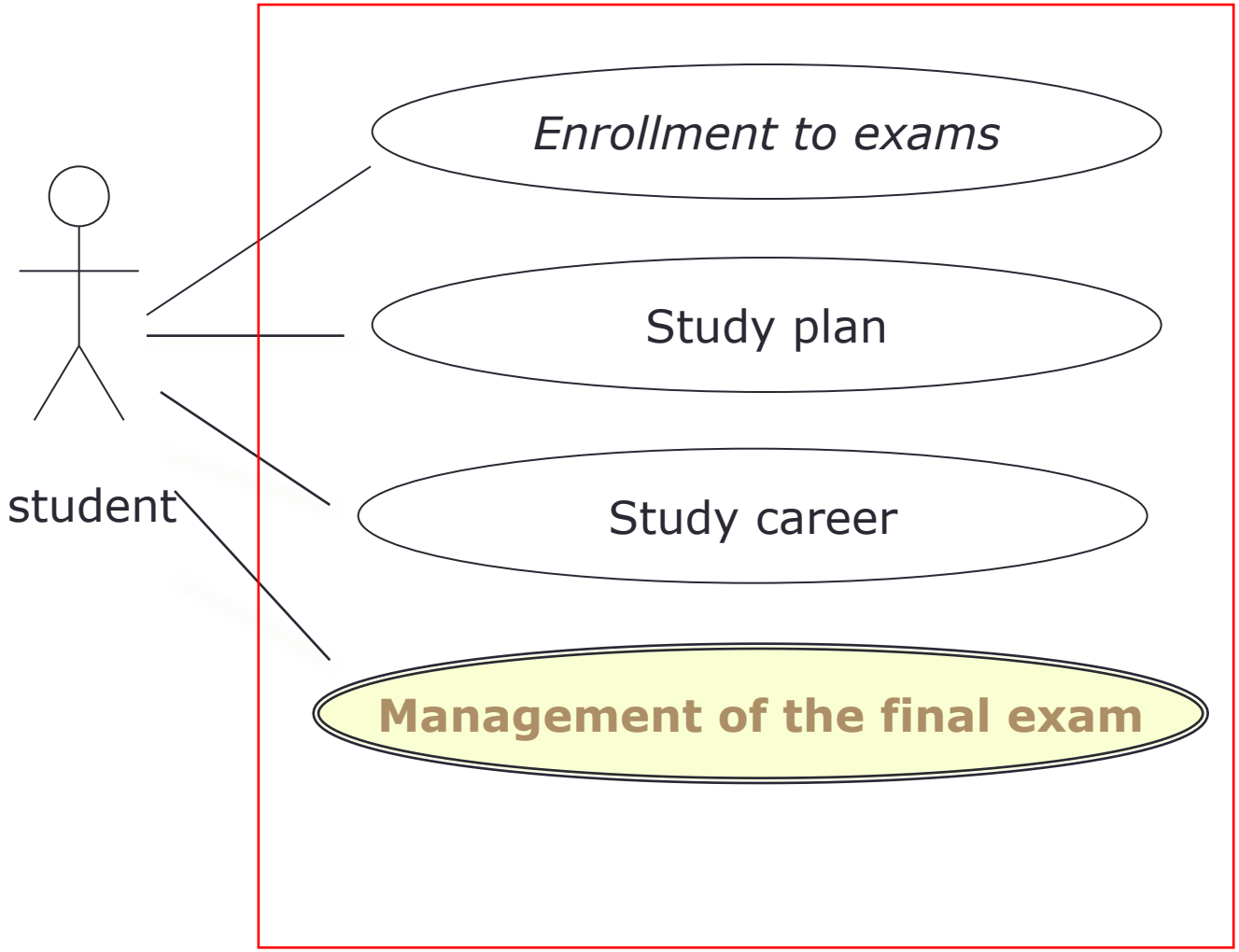
Use Case - Boundary

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



Actor

Use Case Diagram Example



Types of relationship

Between actors and use cases:

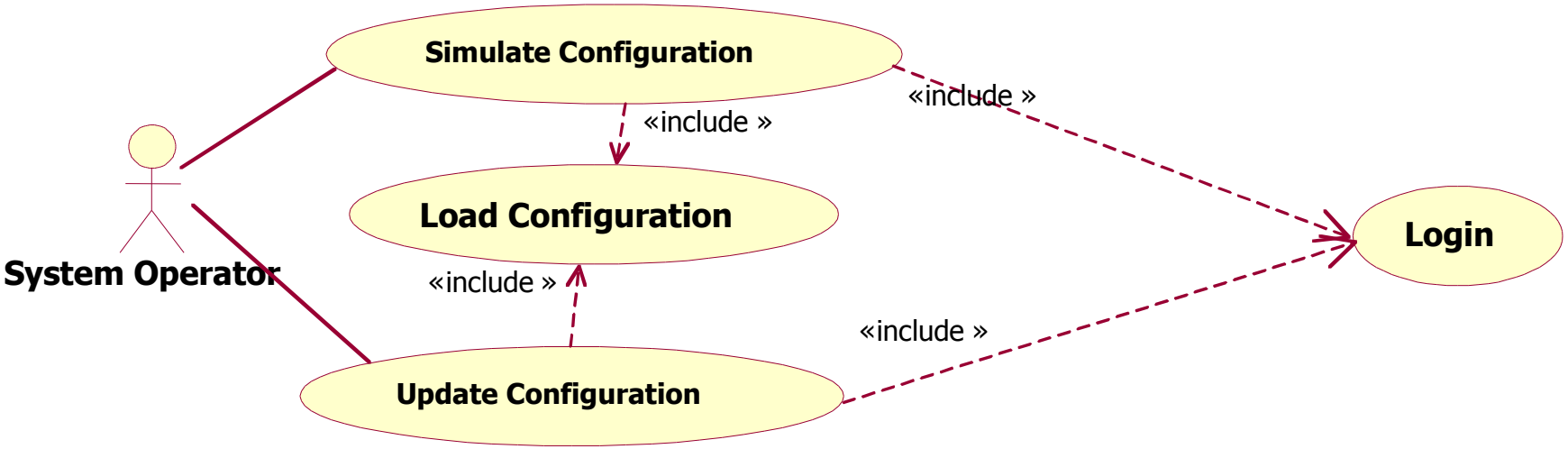
- An **actor** communicates with **use cases** because actors want measurable results.

Between use cases (not between actors and use cases) that you might find useful.

- «**include**» and
- «**extend**»

<<include>>

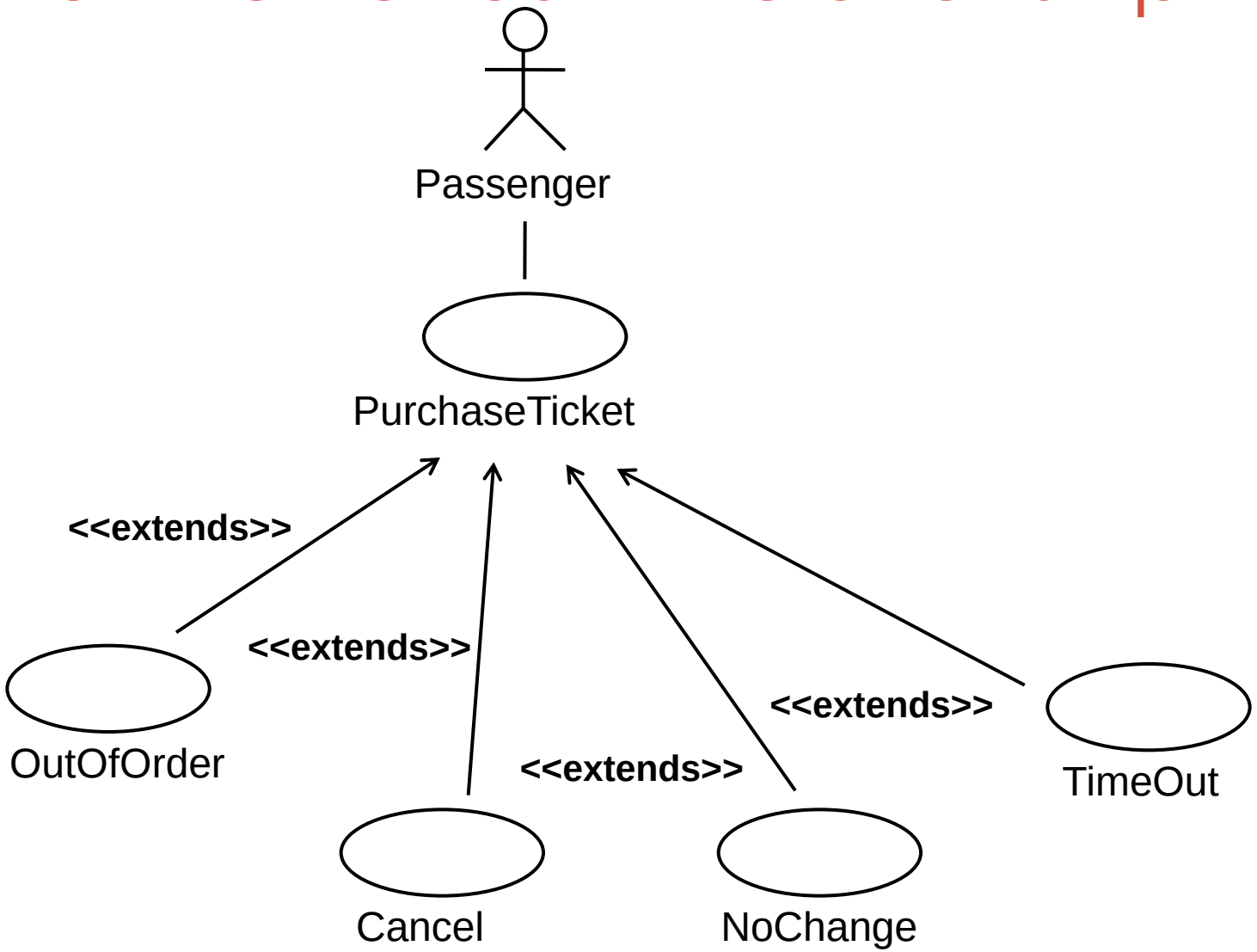
- <<includes>> relationship represents common functionality needed in more than one use case
- <<includes>> behavior is factored out for reuse, not because it is an exception
- The direction of a <<includes>> relationship is to the using use case (unlike the direction of the <<extends>> relationship).



«extend»

- <<extends>> relationships model exceptional or seldom invoked cases
- The exceptional event flows are factored out of the main event flow for clarity
- The direction of an <<extends>> relationship is to the extended use case
- Use cases representing exceptional flows can extend more than one use case.

The <<*extends*>> Relationship



Quiz!!

Which among the following can be heuristic for Use case diagram ?

a) Product can be made actor

b) Never name actors with noun phrases

c) Name Use cases with verb phases

What is true in context to extensions ?

a) Once the basic flow is defined, the extensions can be specified

b) The alternatives are called extension as they extend the activity flow in a different direction from branch point

c) All of the mentioned

Goal of software _____ is to derive an architectural rendering of a system.

a) analysis.
b) design.
c) coding.

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?

Try your hand at creating a version of a very simple use case diagram

Use Cases

- Here is a scenario for a medical clinic.

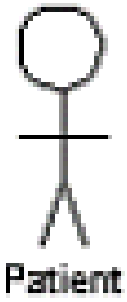
- A *patient calls the clinic to make an appointment for a yearly checkup. The receptionist finds the nearest empty time slot in the appointment book and schedules the appointment for that time slot.*

- We want to draw a use case diagram for this scenario.

Use Cases

- 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. "*

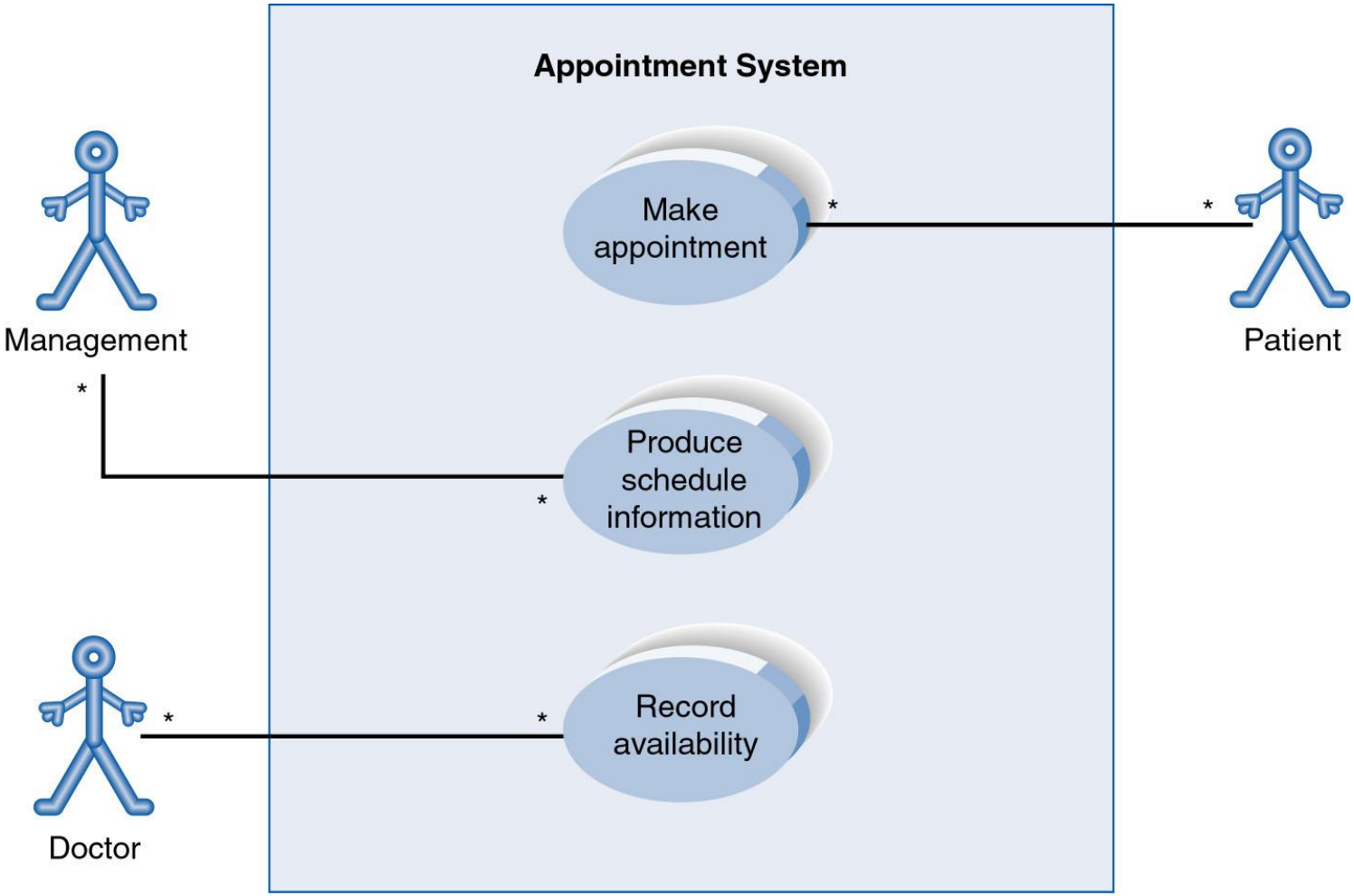


Use Cases

- 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 Diagram

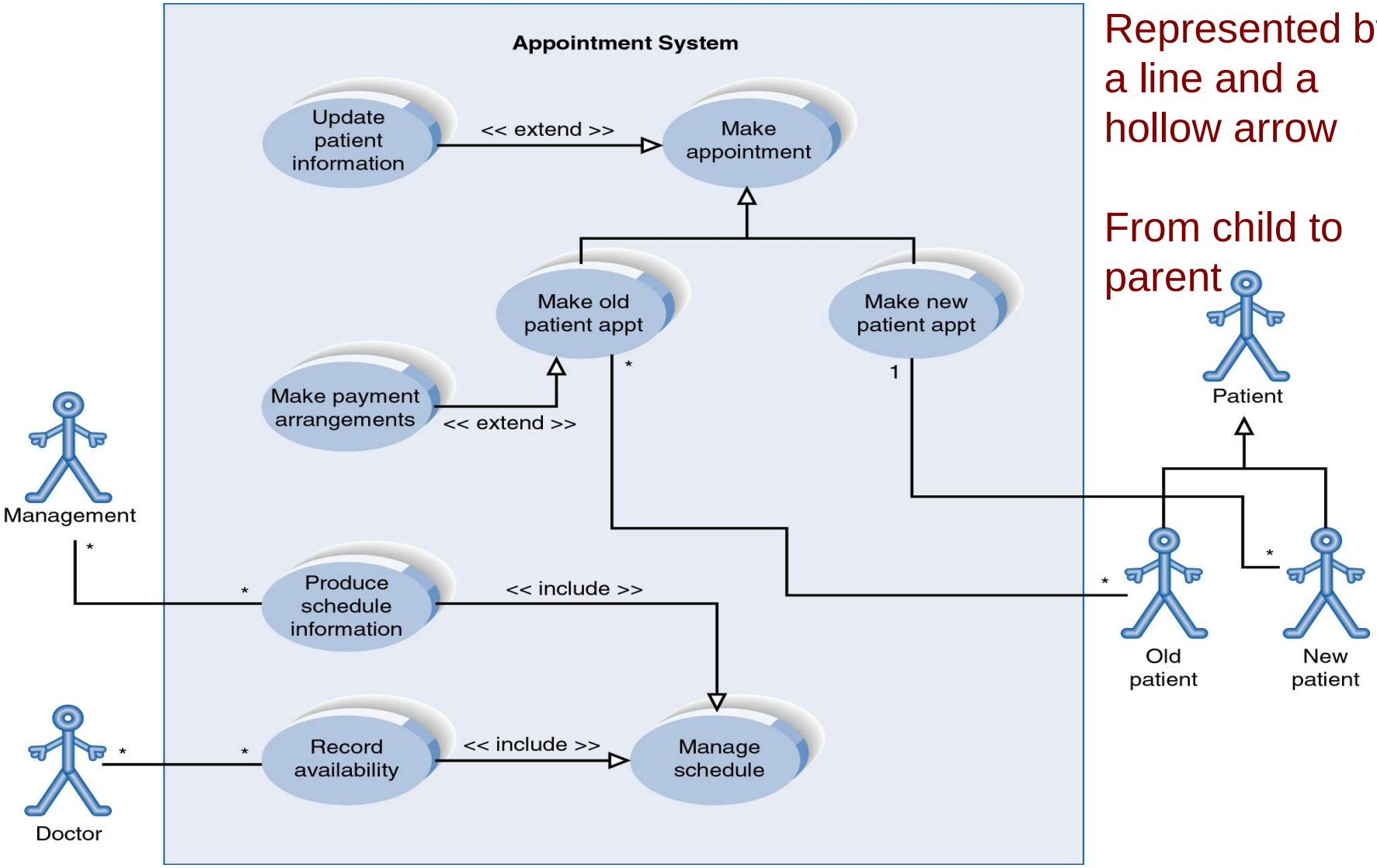


Example of Relationships

Generalization Relationship

Represented by a line and a hollow arrow

From child to parent



Exam Paper Summer 2014

The library lends books, DVDs and CDs. The system should keep a catalog of all the titles in the library. There may be multiple copies of each title in the library. A copy could be on the shelves, on loan to a user, held for a user who has reserved it, held for repair, or it may be lost or discarded. Books can be borrowed for 4 weeks, CDs for 2 weeks and DVDs for 1 week. No user is allowed to have more than 6 copies on loan at any one time. A user can reserve up to 6 titles. If any copy of the title is available (i.e. on the shelves) or becomes available by being returned by another user, then that copy is assigned to the first reservation in the reservation queue, removed from the shelves and held by the librarian for the user. If the user has not collected the copy within a week, then the reservation is cancelled and the copy becomes available again (possibly to be immediately held for another reservation). When a damaged copy is returned it is sent for repair and discarded if it is too damaged. A user can renew their loan of a copy as many times as they like, unless someone else reserves it, in which case they have to return it by the end of their current loan period. A copy that is 10 weeks overdue is assumed to be lost. At any time a copy that is on the shelf may be chosen by the librarian to be discarded and removed from the library.

Exam Paper Summer 2014

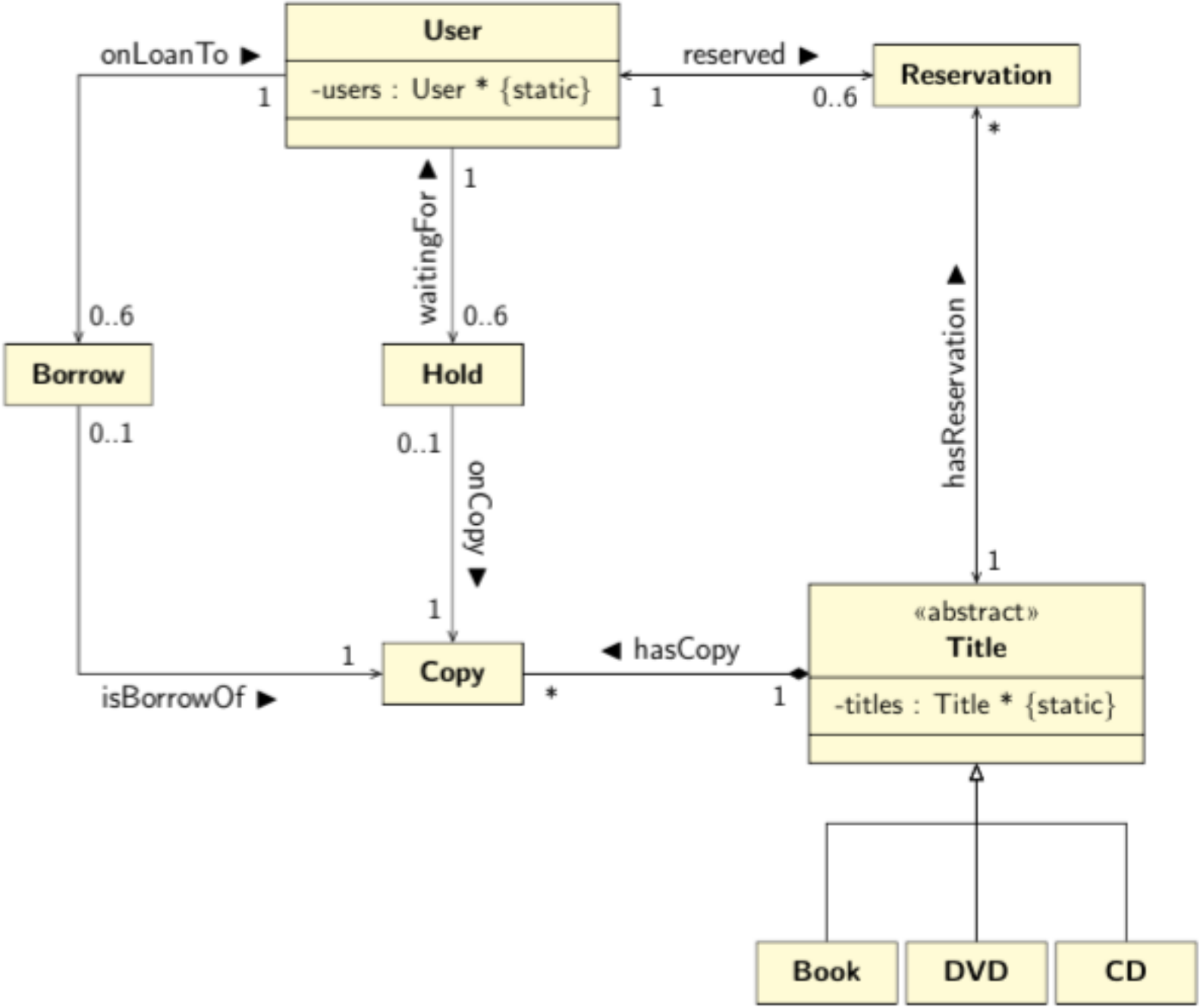
(a) Conduct a class design for the above application description and draw a UML Class Diagram to describe your final design. You may omit attributes and operations from your class diagram.

(Marks will be divided equally between your choice of classes, the relationships between the classes and the cardinalities on those relationships.)

[30%]

(b) Write a use case to reserve a title for a user.

[10%]


User
-users : User * {static}
onLoanTo
reserved
1
1
0..6
Reservation
*
1
waitingFor
1
0..6
0..6
Borrow
Hold
hasReservation
0..1
0..1
onCopy
1
1
«abstract»
Title
1
Copy
hasCopy
isBorrowOf
*
1
-titles : Title * {static}
Book
DVD
CD

(b) An acceptable use case would be as follows, although a large amount of variation is possible.

```
Name          Reserve title for user
Initiator     User
Goal          Allows a user to search for an item in the library
              catalog and reserve it
Main Success Scenario
1.    User submits search criteria to the system
2.    System shows search results to the user
3.    User selects title from search results to reserve
4.    System reserves title

Extension
3.    Search returns empty results
      1. System reports fault to User
      2. Resume 1
3.    User chooses not to reserve any title
      1. Stop
4.    User already has 6 titles reserved
      1. System reports fault to User
      2. Fail
```