Chapter 6: Functional Modelling

ITU 07302

SYSTEM ANALYSIS AND DESIGN

WITH OBJECT ORIENTED PROGRAMMING

Based on Course Textbook:

Systems Analysis and Design With UML 2.0 An Object-Oriented Approach, Second Edition

Alan Dennis, Barbara Wixom, and David Tegarden

Objectives

- 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.

USE-CASE DESCRIPTIONS

Key Ideas

- 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?

- Describe basic functions of the system
  - What the **user** can do
  - How the system responds
- Use cases are building blocks for continued design activities.

How Are Use-Cases Created?

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

Types of Use-Cases

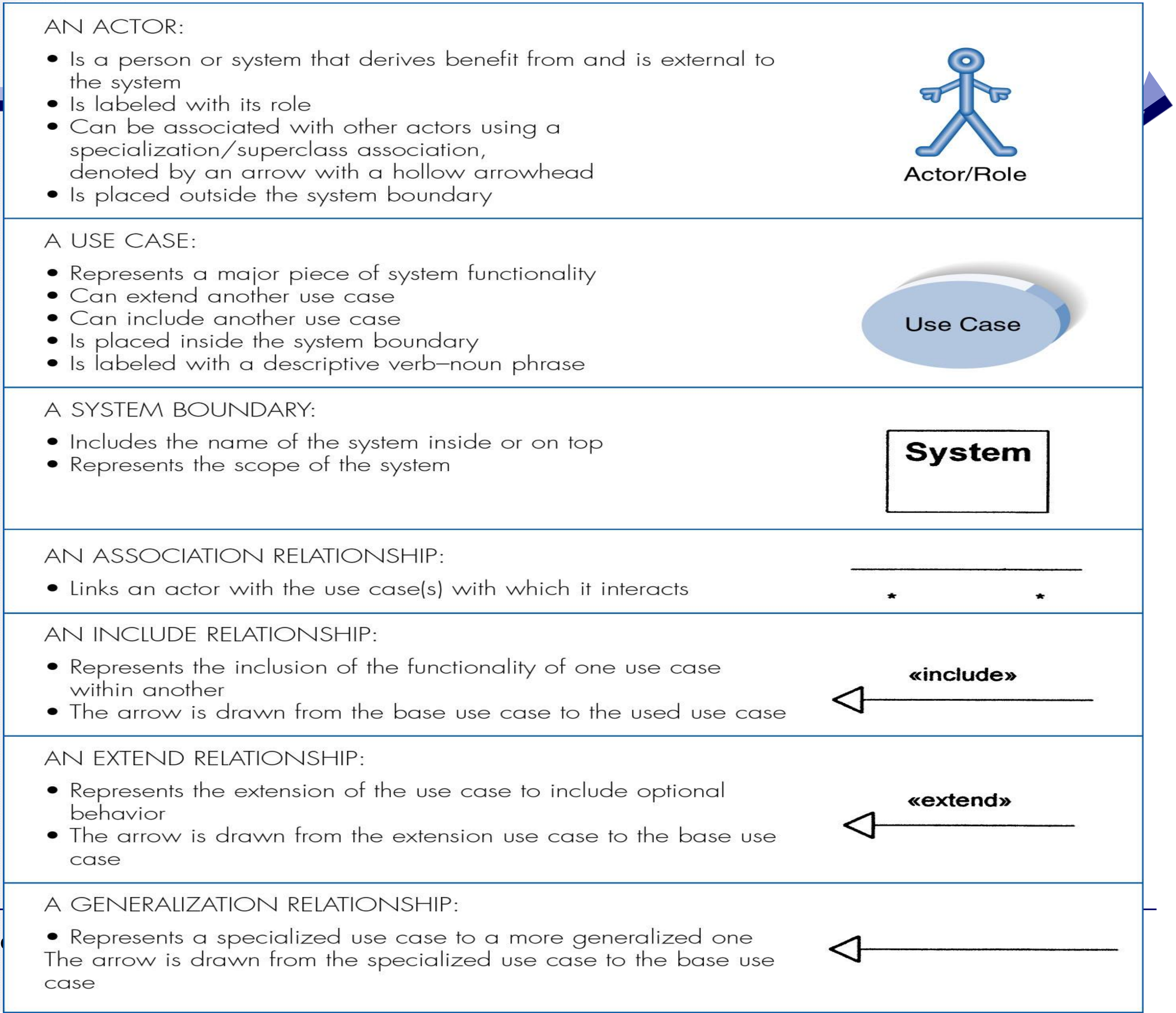
- Overview versus detail
  - The use case represents an important business process.
  - The use case supports revenue generation or cost reduction.
  - Technology needed to support the use case is new or risky and therefore will require considerable research.
- Essential versus real

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:		

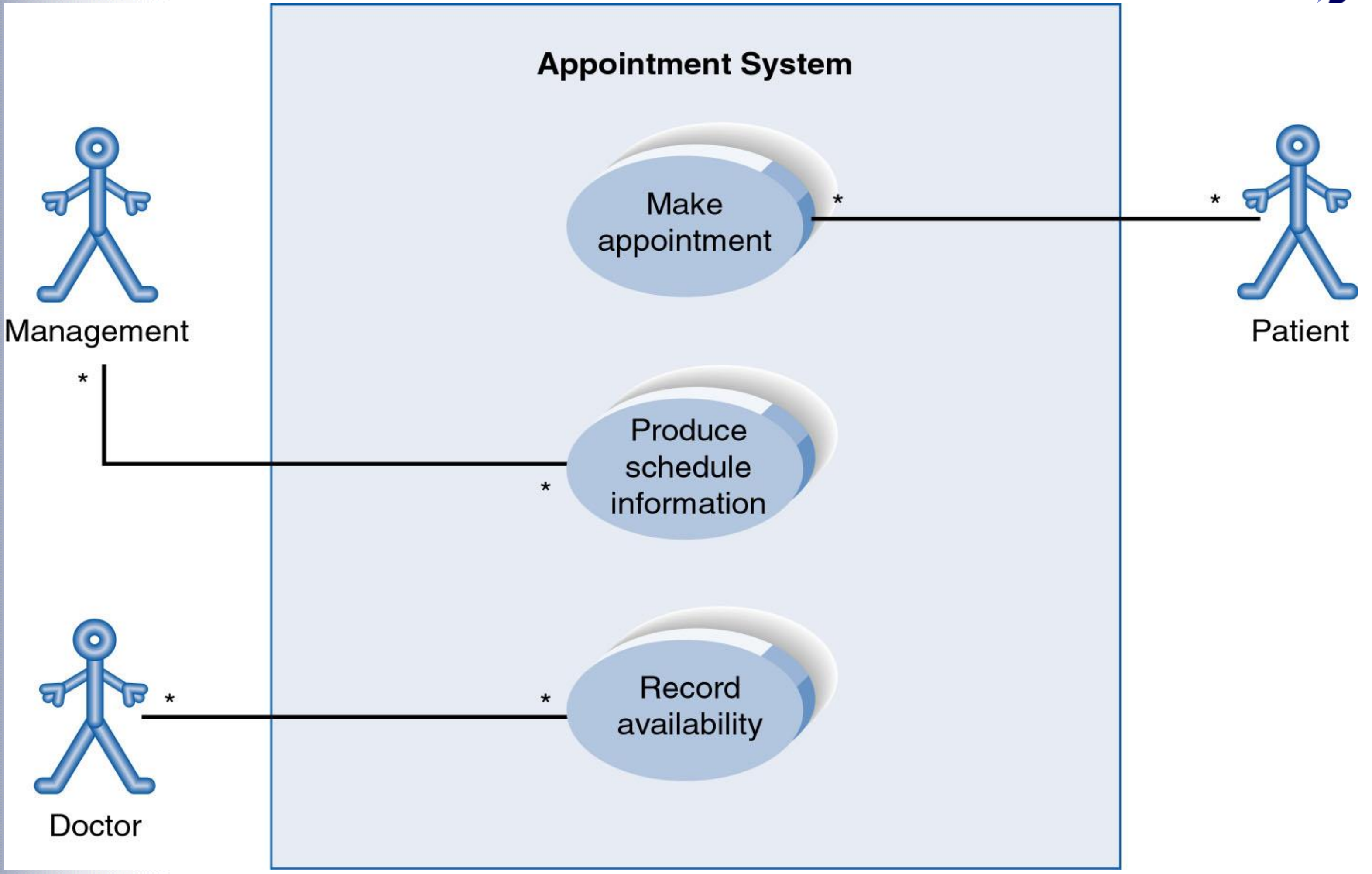
USE-CASE DIAGRAMS

Use-Case Diagram Syntax

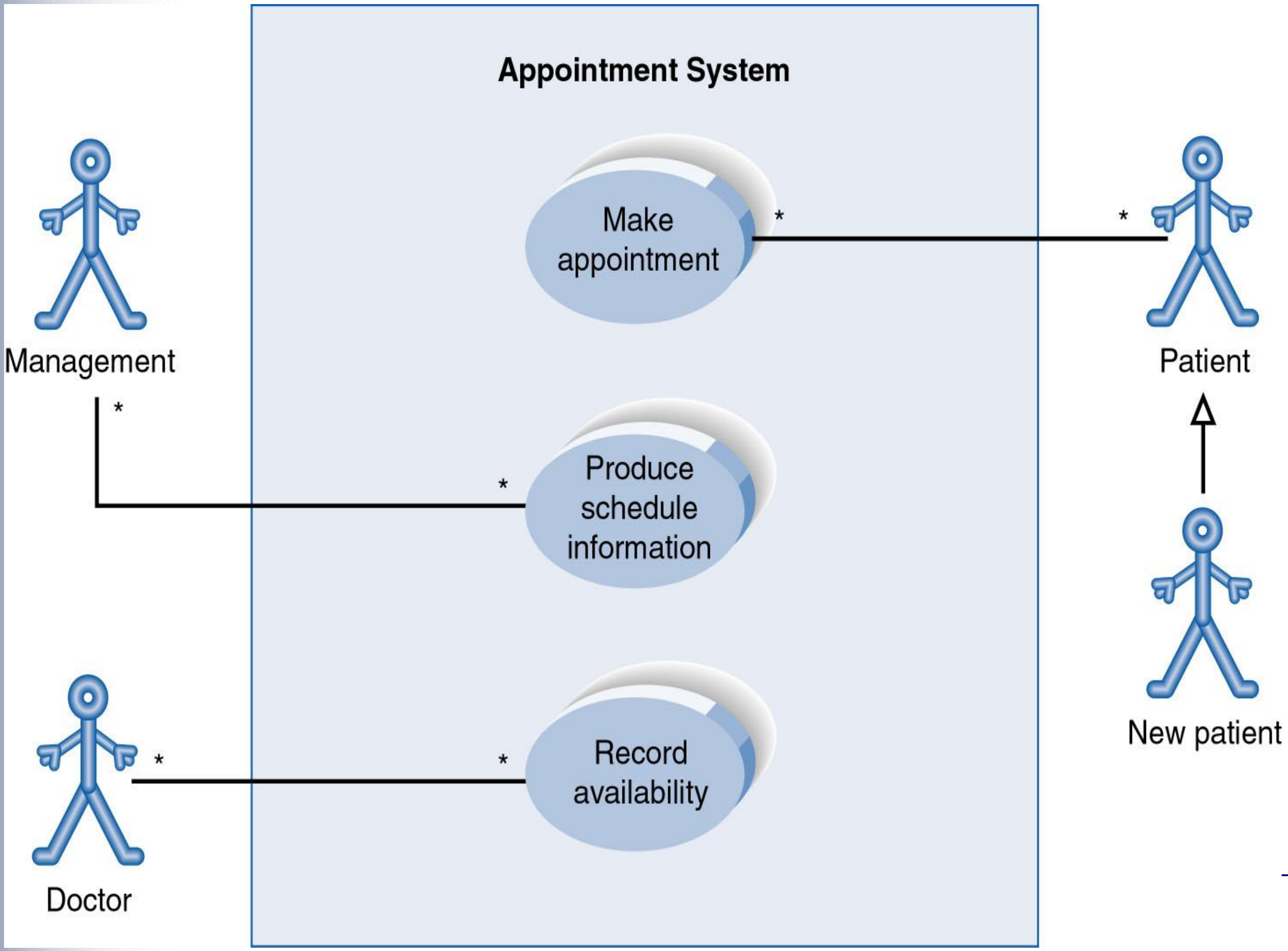


AN ACTOR: • Is a person or system that derives benefit from and is external to the system • Is labeled with its role • Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead • Is placed outside the system boundary	Actor/Role
A USE CASE: • Represents a major piece of system functionality • Can extend another use case • Can include another use case • Is placed inside the system boundary • Is labeled with a descriptive verb–noun phrase	Use Case
A SYSTEM BOUNDARY: • Includes the name of the system inside or on top • Represents the scope of the system	System
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 used use case	«include»
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	«extend»
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	

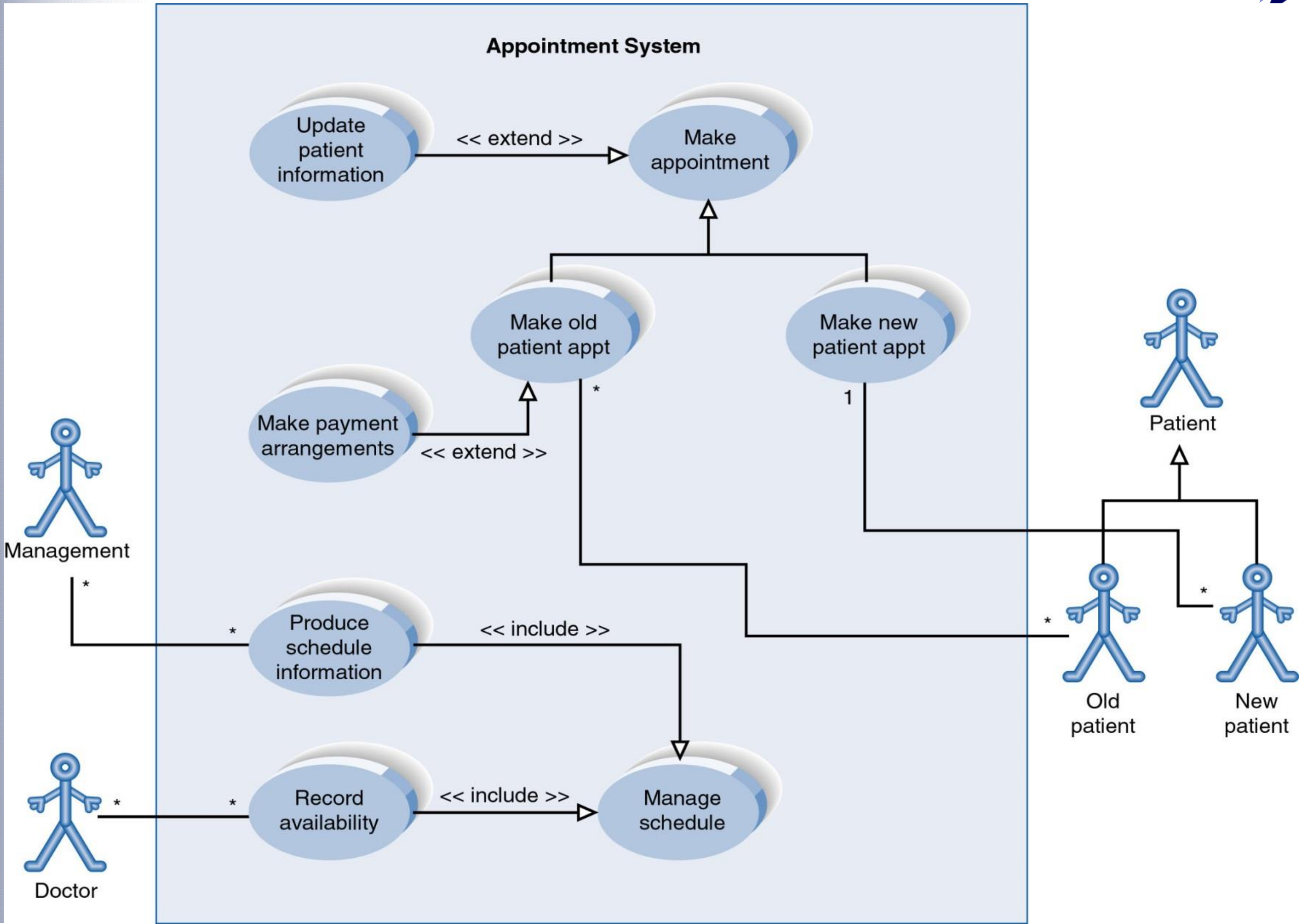
The Use-Case Diagram for Appointment System



Use-Case Diagram with Specialised Actor



Extend and Include Relationships



CREATING USE-CASE DESCRIPTIONS AND USE-CASE DIAGRAMS

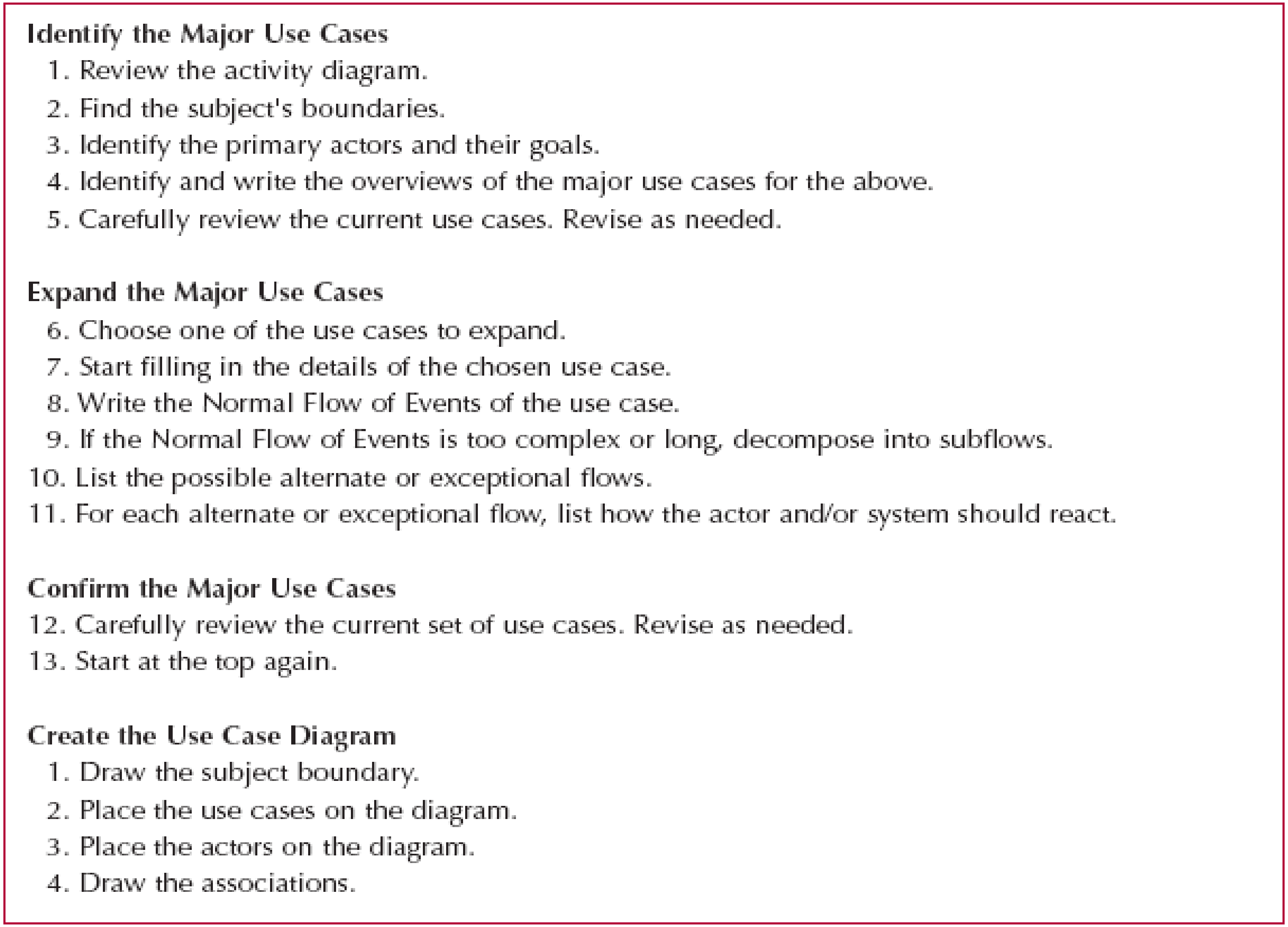
4 Major Steps in Writing Use-Cases…

- 1. Identify the major use-cases
- 2. Expand the major use-case
- 3. Confirm the major use-cases
- 4. Create the use-case diagram

1. Identifying the Major Use-Cases

- Identify the system's boundaries
- List the primary actors
- List the goals of each primary actor
- Identify and write the major use-cases
- Carefully review use-cases

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.

2. Expand the Major Use-Cases

- Choose one major use-case to expand
- Fill in details on the use-case template
- Fill in the steps of the normal flow of events
- Normalize the size of each step
- Describe alternate or exceptional flows
- Simplify and organize as necessary

3. 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

4. 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

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.