

# **Systems Analysis and Design with UML Version 2.0, Second Edition**

---

Alan Dennis, Barbara Wixom, and David Tegarden

## **Chapter 6: Functional Modeling**

John Wiley & Sons, Inc.  
Copyright 2005

# Copyright © 2005 John Wiley & Sons, Inc.

---

- All rights reserved. Reproduction or translation of this work beyond that permitted in Section 117 of the 1976 United States Copyright Act without the express written permission of the copyright owner is unlawful.
- Request for further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.
- The purchaser may make back-up copies for his/her own use only and not for redistribution or resale.
- The Publisher assumes no responsibility for errors, omissions, or damages, caused by the use of these programs or from the use of the information contained herein.

# 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.

# Business Process Modeling with Activity Diagrams



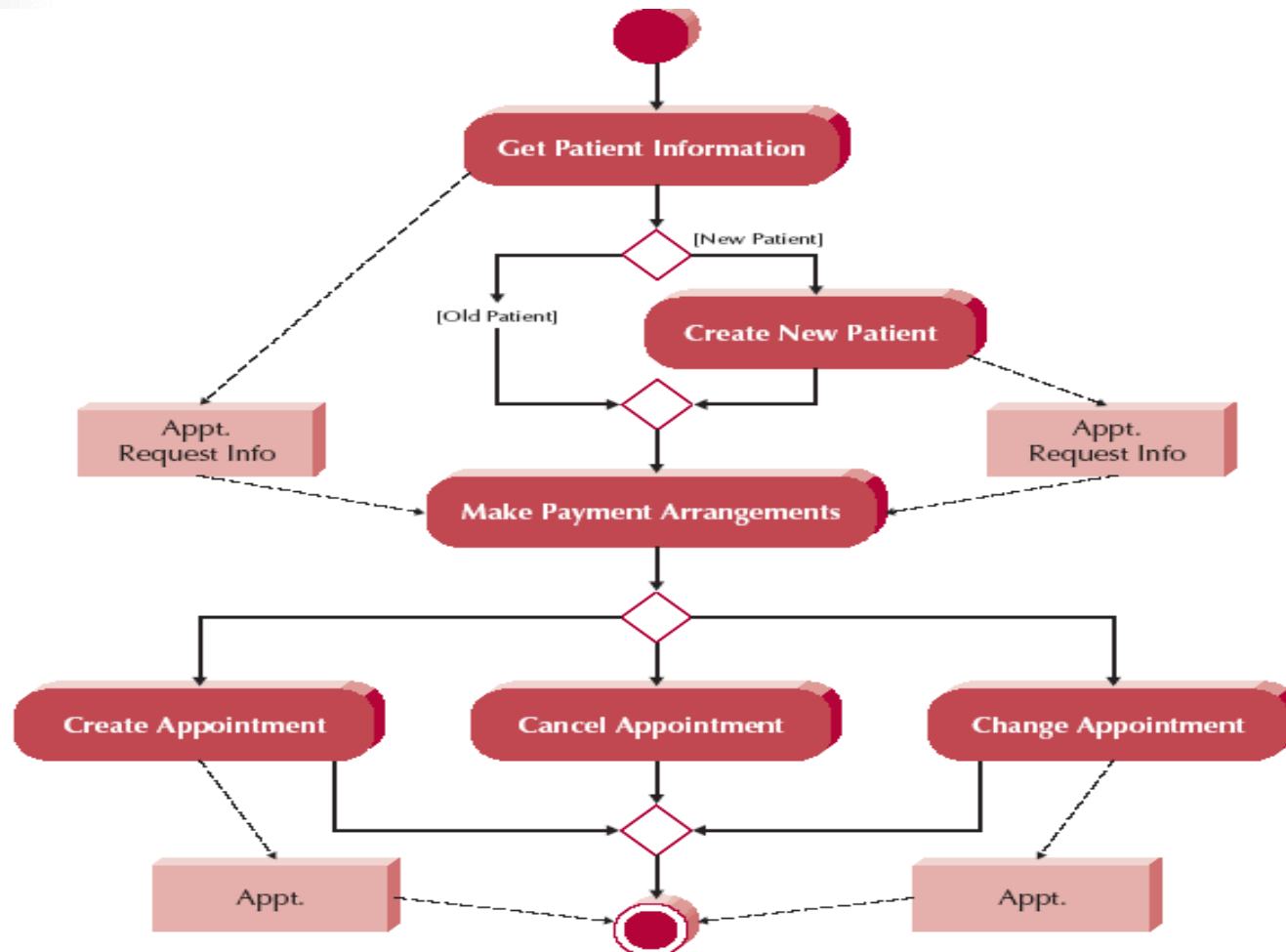
- ▣ Elements of an Activity Diagram
- ▣ Guidelines for Creating Activity Diagrams

# BPM With Activity Diagrams



- A number of activities support a business process across several departments
- Activity diagrams model the behavior in a business process
  - Sophisticated data flow diagrams
  - Addresses Parallel concurrent activities and complex processes

# Activity Diagram Example



# 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.

# 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:
  - 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.

# 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:		

# Guidelines for Creating Use-Case Descriptions



- Write each step in “SVDPI” form
- Clarify initiator and receivers of action
- Write from independent observer perspective
- Write at same level of abstraction
- Ensure a sensible set of steps
- Apply KISS principle liberally
- Write repeating instructions after the set of steps to be repeated.

# Your Turn

---



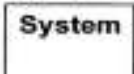




- How would you make requirements gathering (interviews, questionnaires, observation, and document analysis) more effective by knowing that eventually you will be creating use-case descriptions and diagrams?

# USE-CASE DIAGRAMS

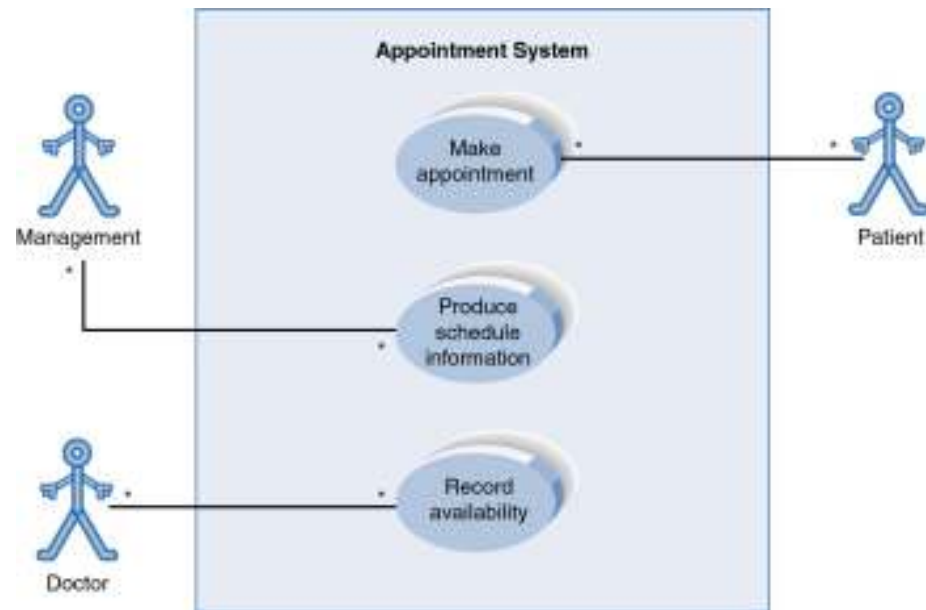




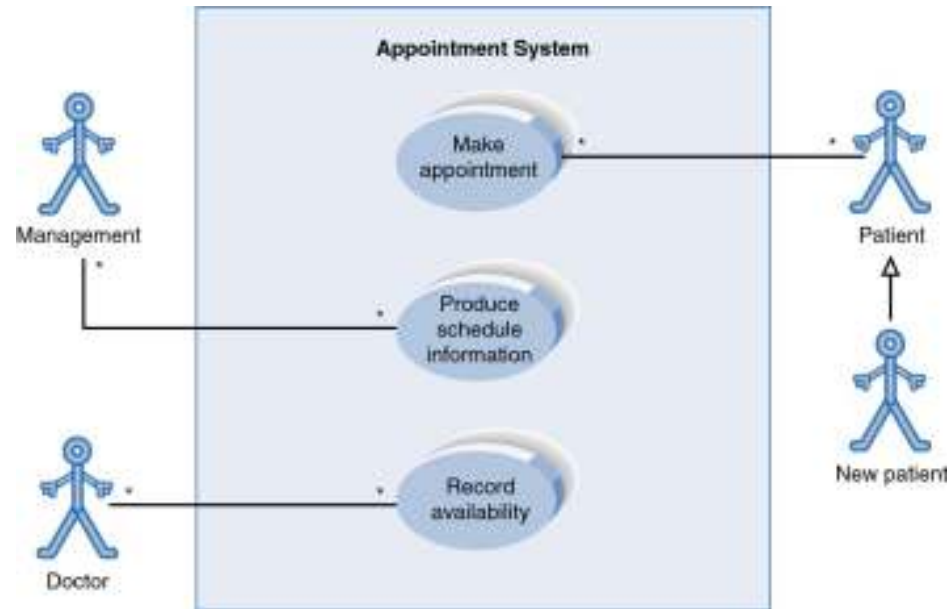
# Syntax for Use-Case Diagram

<p><b>AN ACTOR:</b></p> <ul style="list-style-type: none"> <li>• Is a person or system that derives benefit from and is external to the system</li> <li>• Is labeled with its role</li> <li>• Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead</li> <li>• Is placed outside the system boundary.</li> </ul>	 <p>Actor/Role</p>
<p><b>A USE CASE:</b></p> <ul style="list-style-type: none"> <li>• Represents a major piece of system functionality</li> <li>• Can extend another use case</li> <li>• Can include another use case</li> <li>• Is placed inside the system boundary</li> <li>• Is labeled with a descriptive verb-noun phrase</li> </ul>	 <p>Use Case</p>
<p><b>A SYSTEM BOUNDARY:</b></p> <ul style="list-style-type: none"> <li>• Includes the name of the system inside or on top</li> <li>• Represents the scope of the system</li> </ul>	 <p>System</p>
<p><b>AN ASSOCIATION RELATIONSHIP:</b></p> <ul style="list-style-type: none"> <li>• Links an actor with the use case(s) with which it interacts</li> </ul>	
<p><b>AN INCLUDE RELATIONSHIP:</b></p> <ul style="list-style-type: none"> <li>• Represents the inclusion of the functionality of one use case within another</li> <li>• The arrow is drawn from the base use case to the used use case</li> </ul>	 <p>includes</p>
<p><b>AN EXTEND RELATIONSHIP:</b></p> <ul style="list-style-type: none"> <li>• Represents the extension of the use case to include optional behavior</li> <li>• The arrow is drawn from the extension use case to the base use case</li> </ul>	 <p>extends</p>
<p><b>A GENERALIZATION RELATIONSHIP:</b></p> <ul style="list-style-type: none"> <li>• Represents a specialized use case to a more generalized one. The arrow is drawn from the specialized use case to the base use case.</li> </ul>	

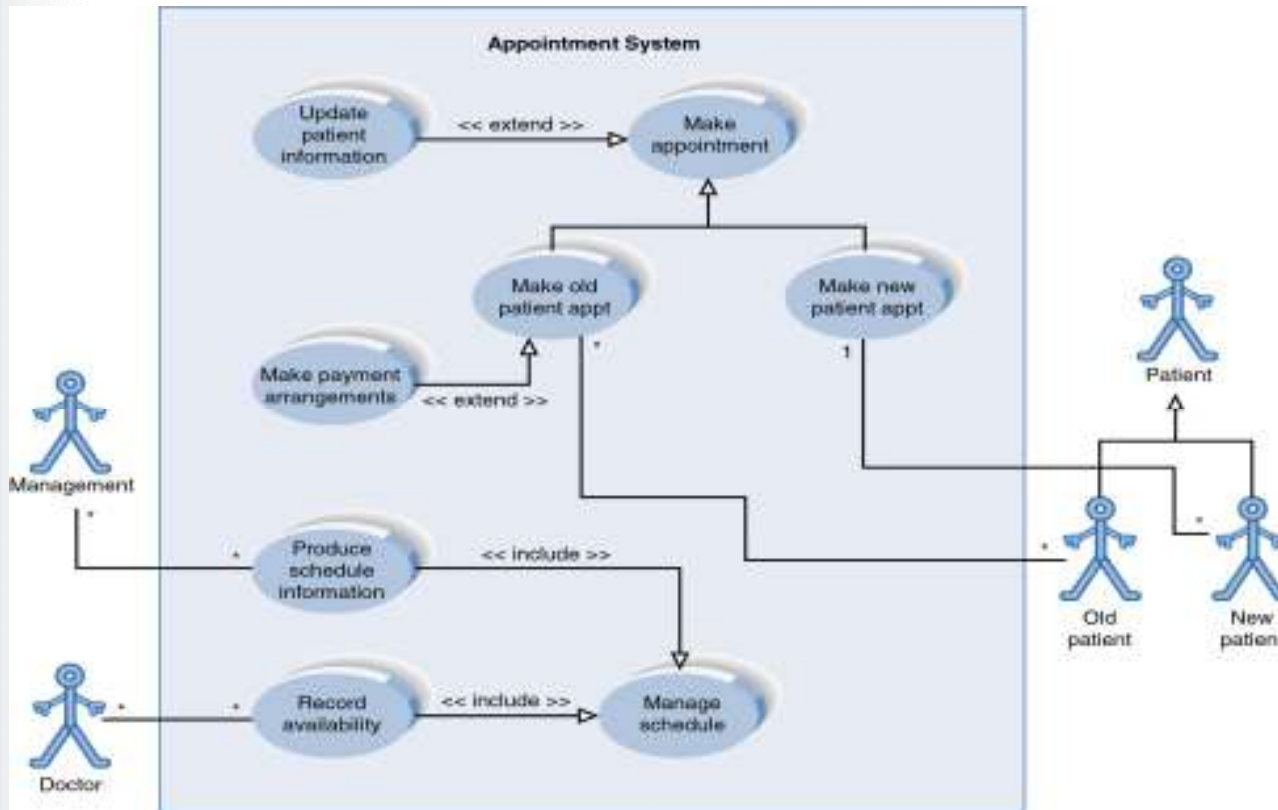
# The Use-Case Diagram for Appointment System



# Use-Case Diagram with Specialized Actor



# Extend and Include Relationships



# CREATING USE-CASE DESCRIPTIONS AND USE- CASE DIAGRAMS



# Major Steps in Writing Use-Cases

---

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

# 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.



# 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

# 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

# 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

# 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

# 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

# CD Selections

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

# 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.

# Expanding the Domain



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



# 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.