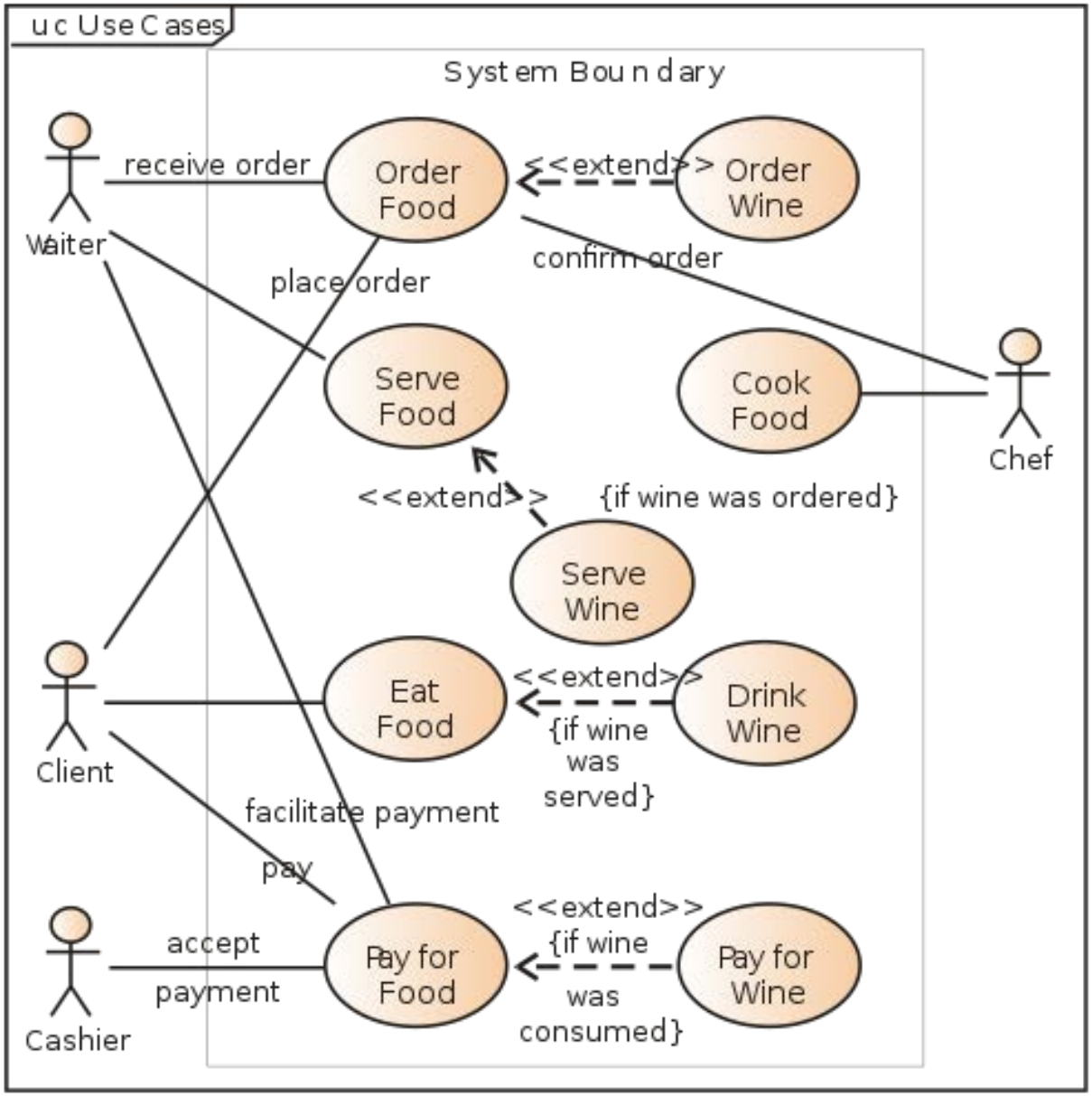
Use Case Diagram

These slides are from Paul Ammann & Jeff Offutt
www.introsoftwaretesting.com

UML Use Case Diagram

- A use case diagram in UML is a type of behavior diagram defined by and created from a Use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
- Roles of the actors in the system can be depicted.

Restaurant Model



UML Use Cases

- Use case: A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as an ellipse.
- Actor: An actor is a person, organization, or external system that plays a role in one or more interactions with the system.
- System boundary box (optional): A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

Syntax for Use Case Diagram

An Actor:

- Is a person or system that derives benefit from and is external to the subject
- Is depicted as either a stick figure (default) or if a non-human actor is involved, as a rectangle with <<actor>> in it (alternative)
- Is labeled with its role
- Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead
- Are placed outside the subject boundary

Actor/Role

<<actor>>
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

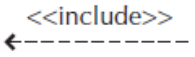
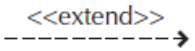
Use Case

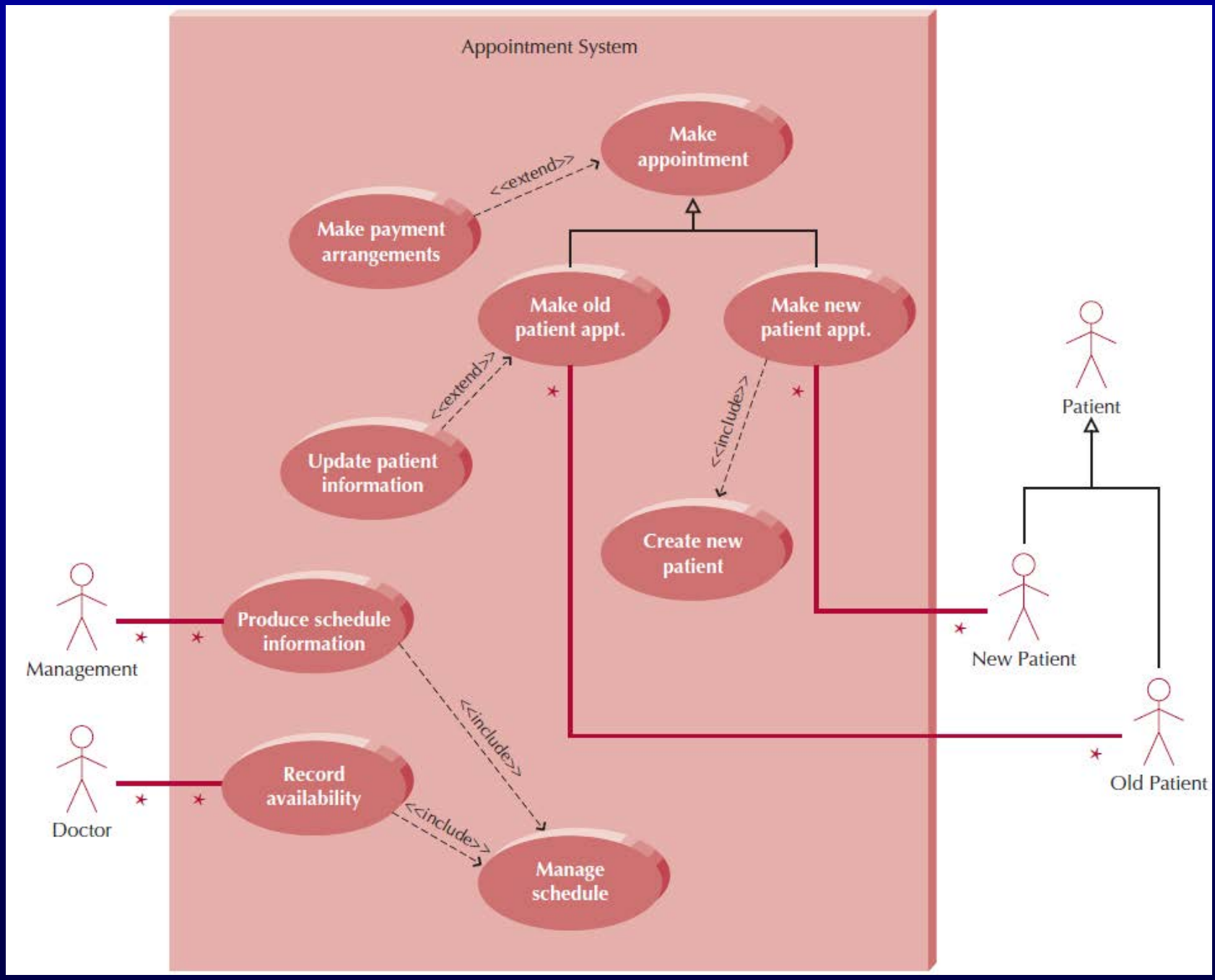
A Subject Boundary:

- Includes the name of the subject inside or on top
- Represents the scope of the subject, e.g., a system or an individual business process

Subject

Syntax for Use Case Diagram (Cont.)

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 included use case	
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	
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	


Appointment System
Make appointment
Make payment arrangements
<<extend>>
Make old patient appt.
Make new patient appt.
<<extend>>
<<include>>
Update patient information
Create new patient
Patient
Produce schedule information
Management
New Patient
<<include>>
Record availability
Doctor
<<include>>
Manage schedule
Old Patient

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.

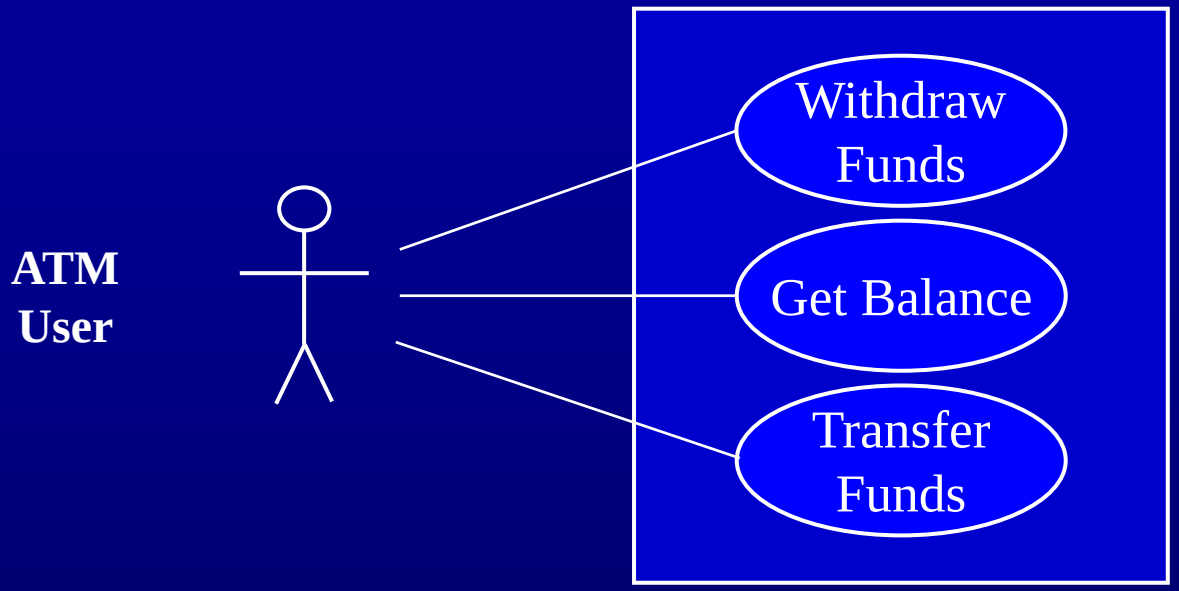
Confirm the Major Use Cases

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.

Simple Use Case Example



- **Actors : Humans or software components that use the software being modeled**
- **Use cases : Shown as ellipse**

Elaboration

- Use cases are commonly elaborated (or documented)
- Elaboration is first written textually
  - Details of operations
  - Alternatives model choices and conditions during execution

Elaboration of ATM Use Case

- **Use Case Name : Withdraw Funds**
- **Summary : Customer uses a valid card to withdraw funds from a valid bank account.**
- **Actor : ATM Customer**
- **Precondition : ATM is displaying the idle welcome message**
- **Description :**
  - **Customer inserts an ATM Card into the ATM Card Reader.**
  - **If the system can recognize the card, it reads the card number.**
  - **System prompts the customer for a PIN.**
  - **Customer enters his/her PIN.**
  - **System checks the card's expiration date and whether the card has been stolen or lost.**
  - **If the card is valid, the system checks if the entered PIN matches the card PIN.**
  - **If the PINs match, the system finds out what accounts the card can access.**
  - **System displays customer accounts and prompts the customer to choose a type of transaction. There are three types of transactions, Withdraw Funds, Get Balance and Transfer Funds. (The previous eight steps are part of all three use cases; the following steps are unique to the Withdraw Funds use case.)**

Elaboration of ATM Use Case – (2/3)

- **Description (continued) :**
  - Customer selects Withdraw Funds, selects the account number, and enters the amount.
  - System checks that the account is valid, makes sure that customer has enough funds in the account, makes sure that the daily limit has not been exceeded, and checks that the ATM has enough funds.
  - If all four checks are successful, the system dispenses the cash.
  - System prints a receipt with a transaction number, the transaction type, the amount withdrawn, and the new account balance.
  - System ejects card.
  - System displays the idle welcome message.

Elaboration of ATM Use Case – (3/3)

- **Alternatives :**
  - If the system cannot recognize the card, it is ejected and the welcome message is displayed.
  - If the current date is past the card's expiration date, the card is confiscated and the welcome message is displayed.
  - If the card has been reported lost or stolen, it is confiscated and the welcome message is displayed.
  - If the customer entered PIN does not match the PIN for the card, the system prompts for a new PIN.
  - If the customer enters an incorrect PIN three times, the card is confiscated and the welcome message is displayed.
  - If the account number entered by the user is invalid, the system displays an error message, ejects the card and the welcome message is displayed.
  - If the request for withdraw exceeds the maximum allowable daily withdrawal amount, the system displays an apology message, ejects the card and the welcome message is displayed.
  - If the request for withdraw exceeds the amount of funds in the ATM, the system displays an apology message, ejects the card and the welcome message is displayed.
  - If the customer enters Cancel, the system cancels the transaction, ejects the card and the welcome message is displayed.
- **Postcondition :**
  - Funds have been withdrawn from the customer's account.

Use Cases to Activity Diagrams

- Activity diagrams indicate flow among activities
- Activities should model user level steps
- Two kinds of nodes:
  - Action states
  - Sequential branches
- Use case descriptions become action state nodes in the activity diagram
- Alternatives are sequential branch nodes
- Flow among steps are edges
- Activity diagrams usually have some helpful characteristics:
  - Few loops
  - conditions

ATM Withdraw Activity Graph