

# Object oriented Analysis &Design

## 面向对象分析与设计



### Lecture\_02 用例模型 Use Case & Requirements Text CH6

Presenter: **Xiaohong Chen**

陈小红

2020/10/13

# Use Cases

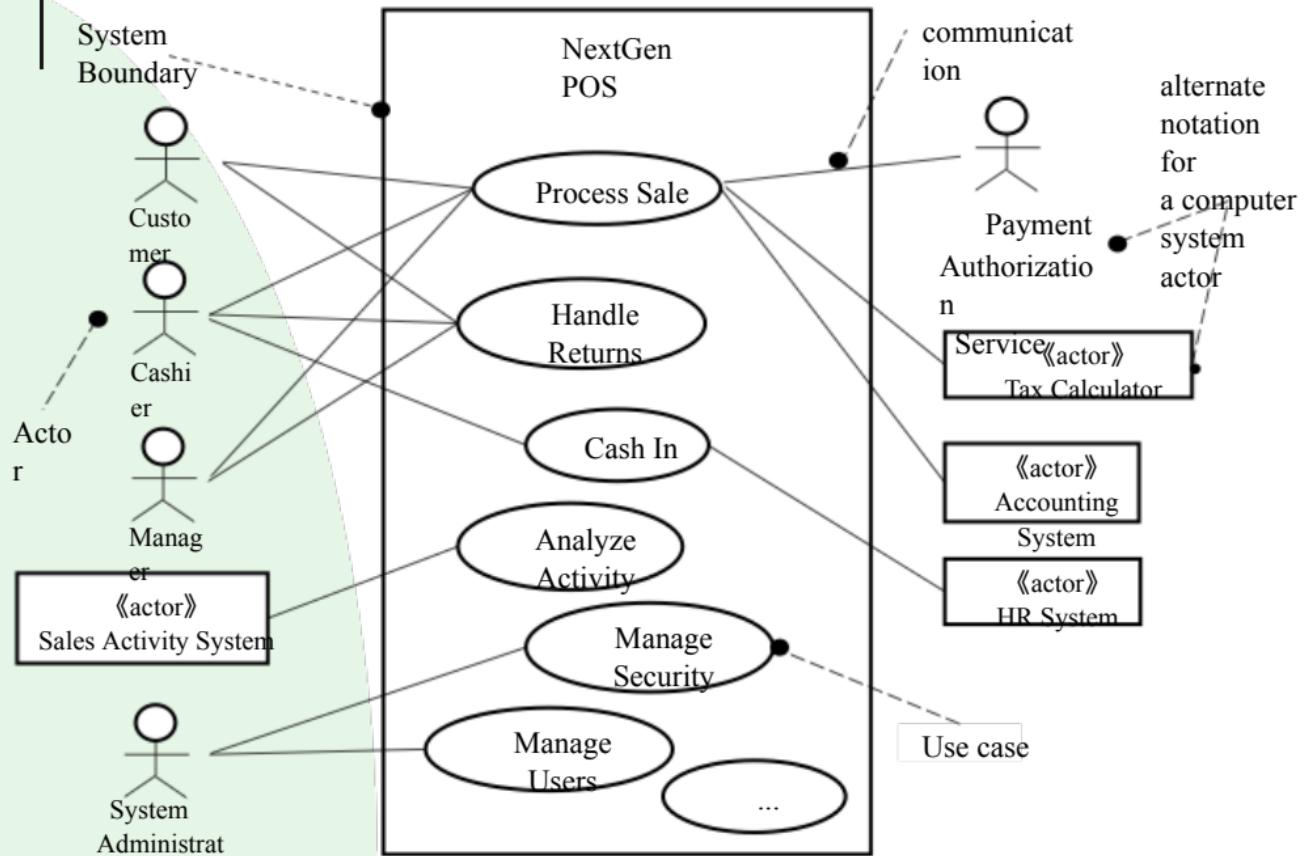
- Use Case is the main input of OOAD
- Target of the Lecture
  - Guide to identify and write use cases.
  - Evaluate use cases.
  - Use the brief, casual, and fully dressed formats, in an essential style.

# 用例 Use Case

- 根据用户对产品的功能的期望, 提取出产品外部功能的描述。
- 用例视图 (use case view), 描述产品外部功能的视图
  - 从软件产品的使用者的角度, 而不是开发者的角度
  - 讨论‘what’而不是‘how’
  - 从用户角度出发, 系统需要为用户提供的功能需求

- 用例模型= 用例图+用例描述

## 用例图



# Use Case Elements

- ## Actor

- Actors are usually individuals involved with the system defined according to their roles. The actor can be a human or other external system.

- ## Use Case

- A use case describes how actors use a system to accomplish a particular goal. Use cases are typically initiated by a user to fulfill **goals describing the activities** and variants involved in attaining the goal.

- ## Relationship

- The relationships between and among the actors and the use cases.

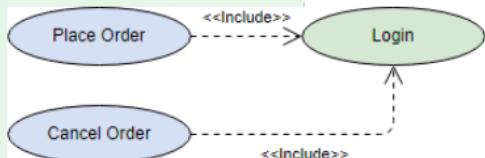
- ## System Boundary

- The system boundary defines the system of interest in relation to the world around it.

# Use Case relationship

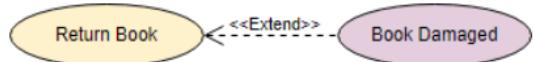
- <<include>> Use Case

The time to use the <<include>> relationship is after you have completed the first cut description of all your main Use Cases. You can now look at the Use Cases and identify common sequences of user-system interaction.

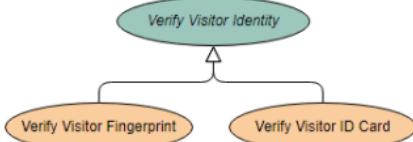


- <<extend>> Use Case

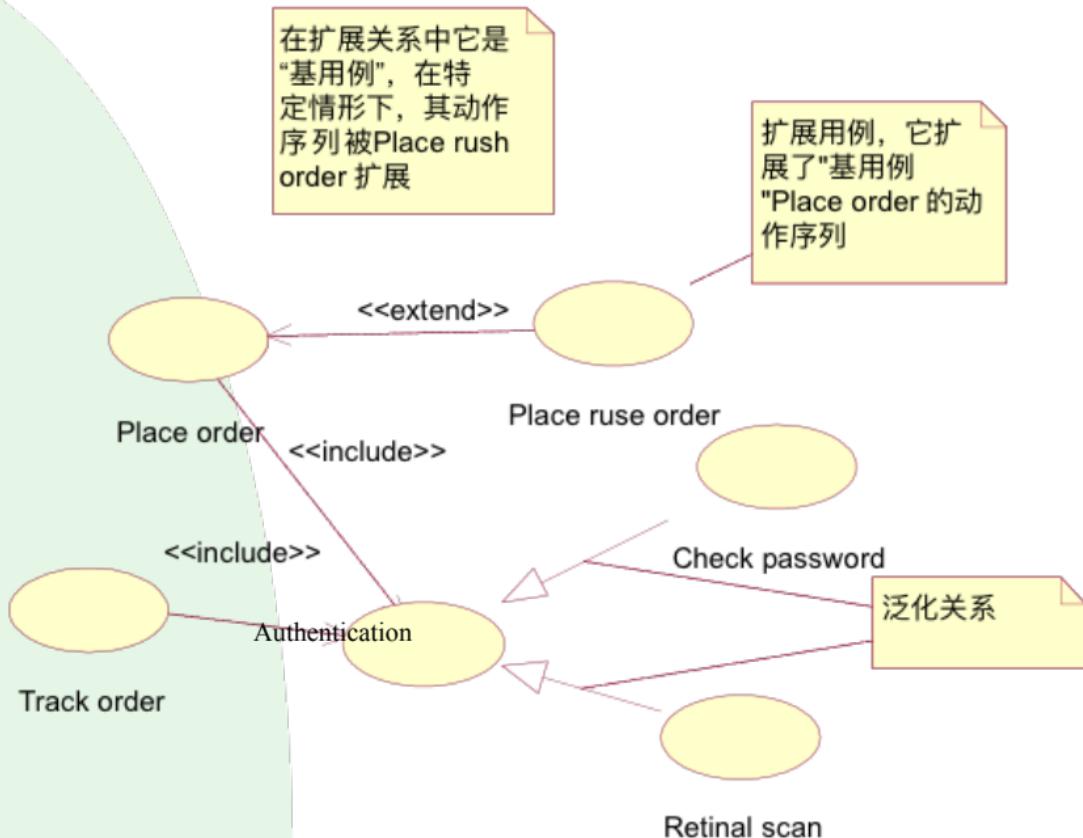
An extending use case is, effectively, an alternate course of the base use case. The <<extend>> use case accomplishes this by conceptually inserting additional action sequences into the base use-case sequence.



- Abstract and generalized Use Case



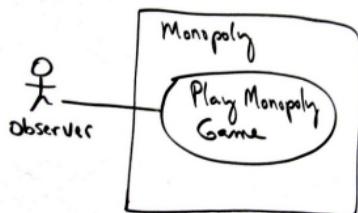
# Example: 泛化、包含和扩展



# Use Case description

- Scope: Monopoly application
- Level: user goal
- Primary Actor: Observer
- Stakeholders and Interests:
  - Observer: Wants to easily observe the output of the game simulation.
- **Main Success Scenario:**
  1. Observer requests new game initialization, enters number of players.
  2. Observer starts play.
  3. System displays game trace for next player move (see domain rules, and "game trace" in glossary for trace details).

Repeat step 3 until a winner or Observer cancels.
- **Extensions:**
  - \*a. At any time, System fails:  
(To support recovery, System logs after each completed move)
    1. Observer restarts System.
    2. System detects prior failure, reconstructs state, and prompts to continue.
    3. Observer chooses to continue (from last completed player turn).
- **Special Requirements:**
  - Provide both graphical and text trace modes.



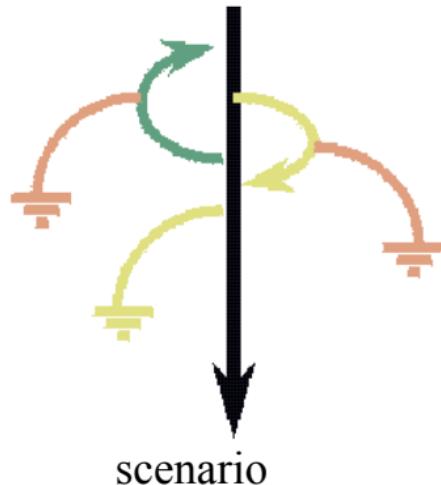
# 用例的描述

The two main parts of the flow of events :

- **The main flow of events**

should cover what "normally" happens when the use case is performed.

- **The alternative flows of events** cover behavior of optional or exceptional character in relation to the normal behavior.

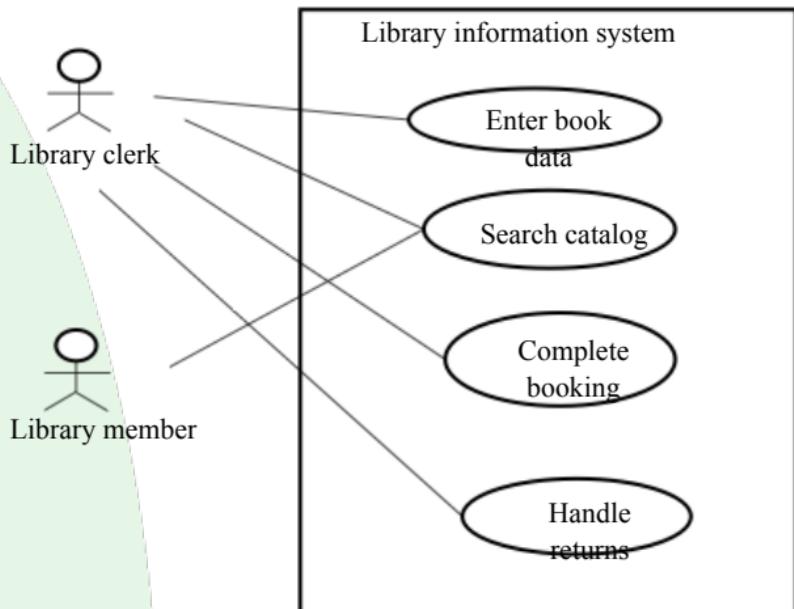


# Exercise: use case diagram

## A library information system

- System contains a catalog (database) with books and journals.
- Library clerks access the system
  - to enter book data (populate the catalog)
  - to search the catalog
  - to complete a booking
  - to handle returns
- Library members access the system
  - to search the catalog

# Reference



# 基本建模元素

- 对软件产品的需求分析首先要定义软件系统的边界，包括两个方面的内容
  - 分析软件产品与外界的联系
  - 确定软件产品与外界的联系时，包含的动态行为及其相互关系
- 在UML的用例图（use case diagram）中，下列建模元素为上述两个方面的内容提供支持
  - 系统参与者(actor)
  - 用例(use case)
  - 依赖、泛化以及关联关系



# 课堂练习：ATM机取款系统



SOFTWARE ENGINEERING INSTITUTE  
華東師範大學軟件學院

# 12306订票系统



# 基本内容：自学

# 1 系统参与者

## 系统参与者

- 代表位于系统之外并和系统进行交互的一类对象
- 用它可以对软件系统与外界发生的交互进行分析和描述



学生

<<Actor>>

数据库

- 系统参与者 代表的角色有：

- 人
- 硬件设备
- 系统



## 1 系统参与者

- 系统参与者除了可以代表作为人的软件使用者之外，还可以代表
- 直接和软件系统交互的软件系统赖以运行的**软/硬件平台**
- 与软件系统有信息交换的**计算机外部设备**
- 例如，**淘宝**  
**支付系统？**



## 1 how to find actor

根据下面的一些问题来寻找系统的参与者：

- ①谁使用系统？ Who or what uses the system
- ②谁安装系统、维护系统？ Who installs the system?  
Who maintains the system
- ③谁启动系统、关闭系统？ Who starts and stops the system
- ④谁从系统中获取信息，谁提供信息给系统？ Who gets and provides information to the system
- ⑤在系统交互中，谁扮演了什么角色？ What roles do they play in the interaction
- ⑥系统会与哪些其他系统进行交互？ What other systems interact with this system
- ⑦Does anything happen at a fixed time?

# 练习：查找 Actor

- 手机软件系统的Actor有：
  - 用户
  - 网络
  - SIM卡
- 学院管理系统的Actor有：
  - 用户/学生/教师/系统管理员
  - 数据库
  - 如果有子系统
    - 其它系统 可以作为一个 Actor
- 银行ATM机的Actor有：
  - 用户/**管理员**
  - 数据库

## 2 用例 (Use Case)

### 指定系统参与者以后

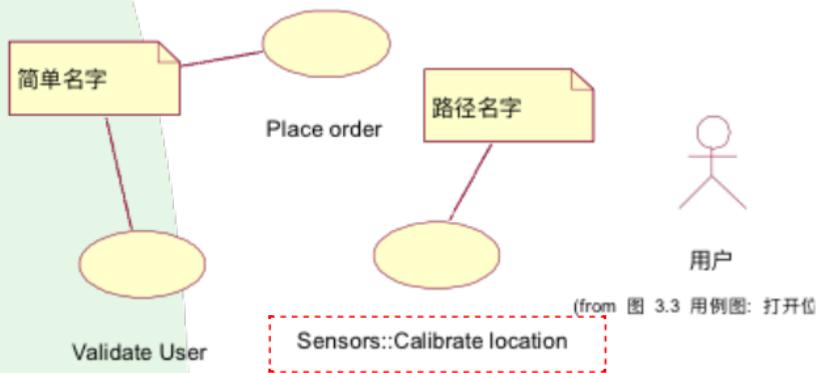
- 需要详细描述系统参与者和软件系统交互的具体内容
- 对交互所代表的软件系统的功能进行分类
- 对这些功能，详细指定其代表的软件系统的动态行为
- 在UML里，软件系统的功能和其代表的动态行为通过**用例**来建模的。
- **用例：**描述了当系统参与者和软件系统进行交互时，软件系统所执行的一系列的动作序列。

## | 2 用例

- 这些动作序列
  - 包含正常使用的各种动作序列
  - 包含对非正常使用时软件系统的动作序列的描述
- 一个用例代表系统参与者和系统的一次交互
  - 用例视图中用例的设置，就代表了软件系统的功能的划分
  - 仔细考虑每个用例代表的动态行为的内容，使得每个用例都能产生一个有价值的、可观察的结果。
  - 用例的划分应使得功能的分布较为均衡、易于理解、易于使用。

## 2.1 用例和系统参与者的绘制

- 用例的名字分为两种
  - 简单名字
  - 路径名字/全称名字(Qualified Name)
- 简单名字是一个不包含冒号的字串
- 路径名则是简单名字前面加上一个包含此用例的所在的模型包的名字，它们之间用两冒号隔开。





## 2.2 how to find use case

根据下面的一些问题来识别用例：

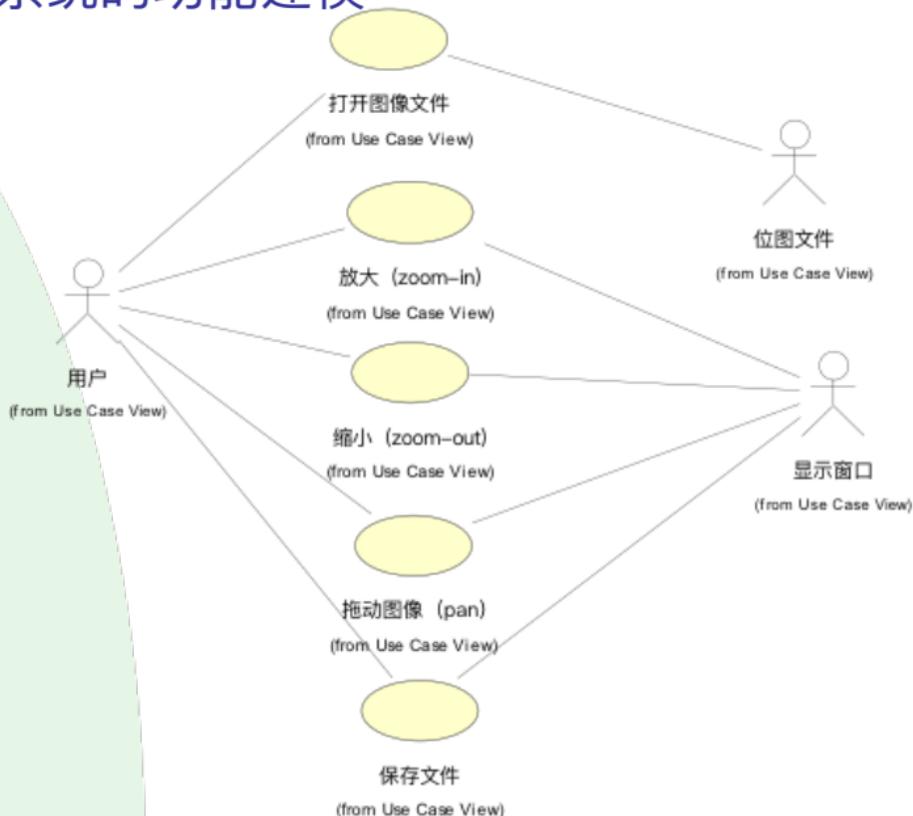
- ① 参与者希望系统提供什么功能； What functions will the actor want from the system?
- ② 系统是否存储和检索信息；
- ③ 当系统改变状态时，是否通知参与者； Are any actors notified when the system changes?
- ④ 是否存在影响系统的外部事件，是哪个参与者通知系统这些外部事件。 Are there external events that notify the system?
- ⑤ Start with actors, then identify what they want to do?
- ⑥ Which actors trigger activity?



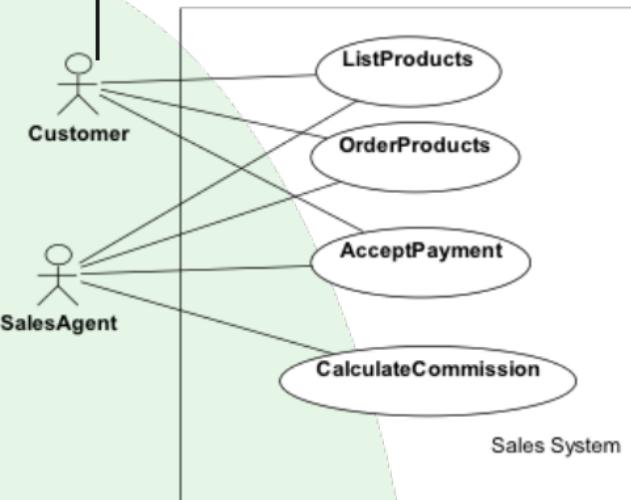
### 3 参与者和用例之间的关系：关联关系

- 例子：假定设计开发一个名为“位图观察器(bmpviewer)”的软件产品，为用户提供图像显示和浏览的功能
  - 经过对用户提出的功能需求的分析和整理，得出用户期望的产品功能如下：
    - 打开一个bitmap文件
    - 可对文件进行放大显示(Zoom-in)
    - 可对文件进行缩小显示(Zoom-out)
    - 可对文件进行浏览(Pan)
    - 可将文件保存到磁盘
  - Actor?
    - 用户
    - 位图文件
    - 显示窗口
  - 用例?
    - 把上列的每项功能都用一个用例表示
  - 把上述功能通过用例和参与者的概念表示在用例图上

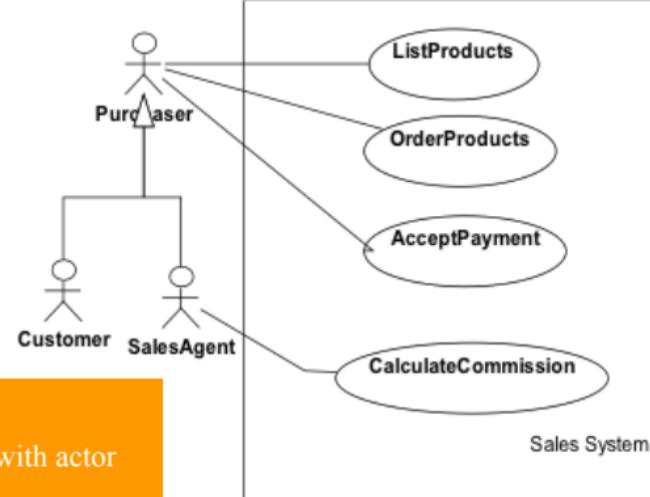
# 用例为软件系统的功能建模



### 3.1 参与者/参与者之间的关系：泛化关系



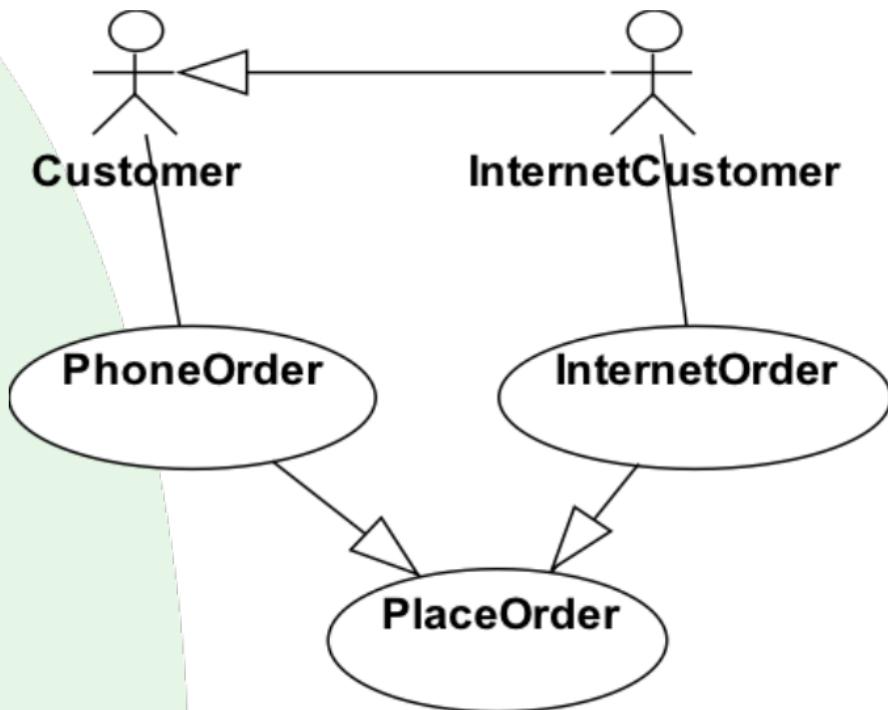
We can simplify with actor generalization!



## 3.2 用例之间的关系：泛化关系

- 考虑订单管理系统（Order Management System）的部分用例模型
  - It is useful to separate **ordinary Customer** from **Internet Customer**, since they have slightly different properties. However, since Internet Customer does exhibit all properties of a Customer, you can say that Internet Customer is a specialization of Customer, indicated with an actor-generalization.
  - The concrete use cases in this diagram are
    - **Phone Order** (initiated by the Customer actor) and **Internet Order** (initiated by Internet Customer).
    - These use cases are both variations of the more general **Place Order** usecase, which in this example is abstract

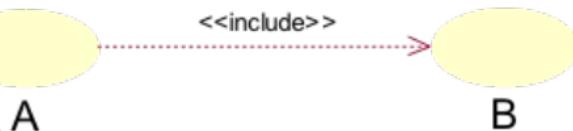
### 3.2 用例之间的关系：泛化关系



### 3.3 用例之间的关系：包含关系

- 在分析和整理系统参与者和系统用例的时候，除了使用泛化关系之外，还可能用到
  - 包含关系(include)
  - 扩展关系 (extend)
- **包含关系：**
  - 位于两个用例之间的**包含关系**意味着基用例显式地在其指定位置将另一个用例包含进来, 使其成为自己的行为的一部分。
    - 包含关系可用于提取公用的用例
    - 在具有包含关系的两个用例中
      - 被包含的那个用例不能单独存在, 它只能以实例的形式存在于包含它的用例之中。

### 3.3 用例之间的关系：包含关系

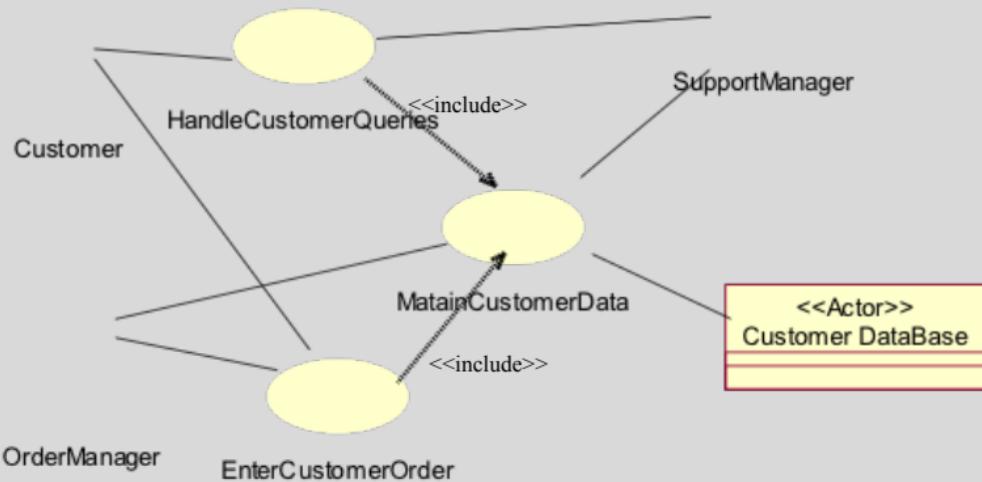


Reading:

B is included in A

- One step in A **calls** the use case B
- A is called *including use case*, B is called *included use case*
- in former versions of UML, you find the stereotype <<use>> instead of <<include>>
- <<include>> relationship is used to sort out common behavior across multiple use cases
- an included use case B for just one including

# Example: <<include>>



# Example : <<include>>

- **Use Case:** Handle Customer Queries

- **Brief Description:** The use case allows a Customer to submit requests.

- **Primary Actor:** Customer

- **Level:** User-goal

- **Main Success Scenario:**

1. The Customer calls the service center.
2. The System opens a customer service request (CSR) and logs date and time.
3. The System routes the call to the next available SupportManager and displays the CSR to them.
4. The SupportManager enters the Customer's identification number.
5. If no prior calls have been placed by this Customer, the SupportManager maintains customer data\*.

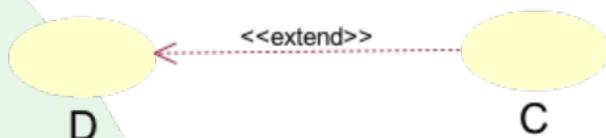
[The remainder of use case is omitted for brevity.]

\* -- the underlined text indicates that another use case is included (called).

## 3.4 用例之间的关系：扩展关系

- 扩展关系
- 两个用例之间的**扩展关系**，代表基用例可以**隐式地**包含另一个用例作为其行为的一部分，包含的位置间接地由另一个用例（扩展用例）确定。基用例可以独立于扩展用例单独存在。
- 当一个用例有多个子流程时，可以用扩展关系对其进行扩展，使得此基用例的不同子流程能在不同的情形下以扩展用例的形式被激活。
- UML把包含关系和扩展关系表示为依赖关系的构造在任何一种图形表示中，箭头所指的模型元素分别代表被包含的用例或被扩展用例(基用例)，而包含关系和扩展关系的构造标记分别是<<include>>和<<extend>>(下页图)

## &lt;&lt;extend&gt;&gt; Relationship

**Reading:**

C extends D in one of the extension points

- An extension point is a labeled position in D's event flow
- C is called *extending use case*, D is called *extended use case* or *base use case*
- <<extend>> is used to **add behavior** to base use case
- Extended use case D should be meaningful also without the extending use case C

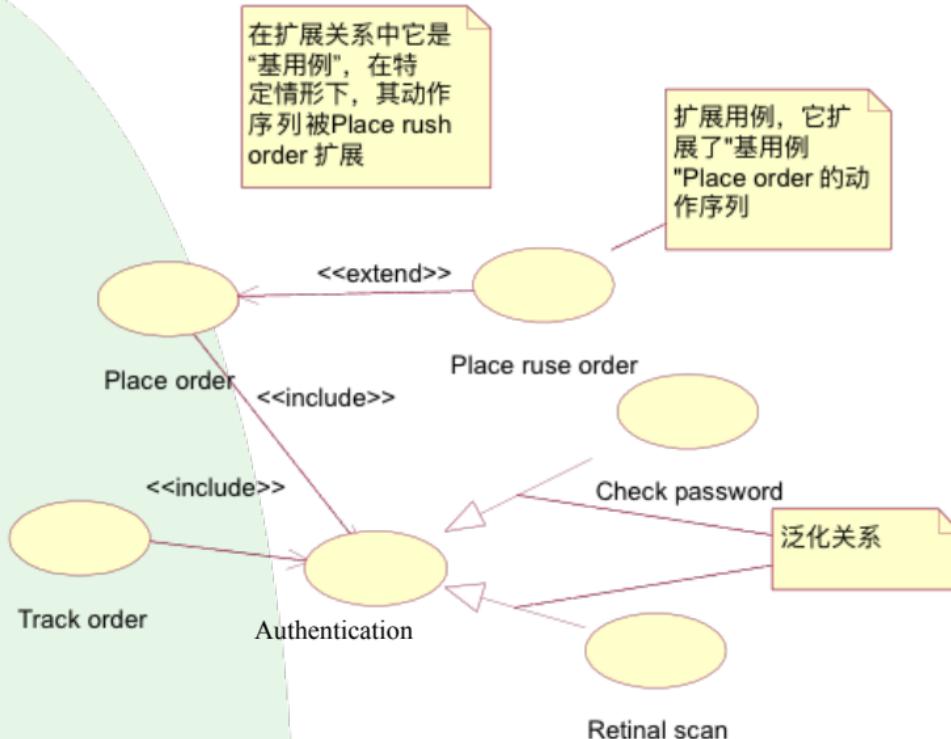
# Example : <<extend>>

- **Use Case:** Maintain Customer Data
- **Brief Description:** The Manager (determined by including use case) adds Customer information.
- **Level:** Subfunction
- **Main Success Scenario:**
  1. The Manager enters or changes the Customer information in any order.
  2. The Manager saves the additions or changes.
  3. System sends data to CustomerDatabase.
  4. The use case ends.
- **Extensions:**
  - 2a. The Manager aborts the additions or changes.
    - 2a.1. Use case continues at step 1.
  - 3a. CustomerDatabase is out of service:
    - 3a.1. Use case ends in failure.

- **Extensions:**

- 2a. The Manager aborts the additions or changes.
  - 2a.1. Use case continues at step 1.
- 3a. CustomerDatabase is out of service:
  - 3a.1. Use case ends in failure.

# Example: 泛化、包含和扩展



## 4 案例

### 考虑订单管理系统（Order Management System）的部分用例模型

It is useful to separate ordinary Customer from Internet Customer, since they have slightly different properties. However, since Internet Customer does exhibit all properties of a Customer, you can say that Internet Customer is a specialization of Customer, indicated with an actor-generalization.

The concrete use cases in this diagram are

Phone Order (initiated by the Customer actor) and  
Internet Order (initiated by Internet Customer).

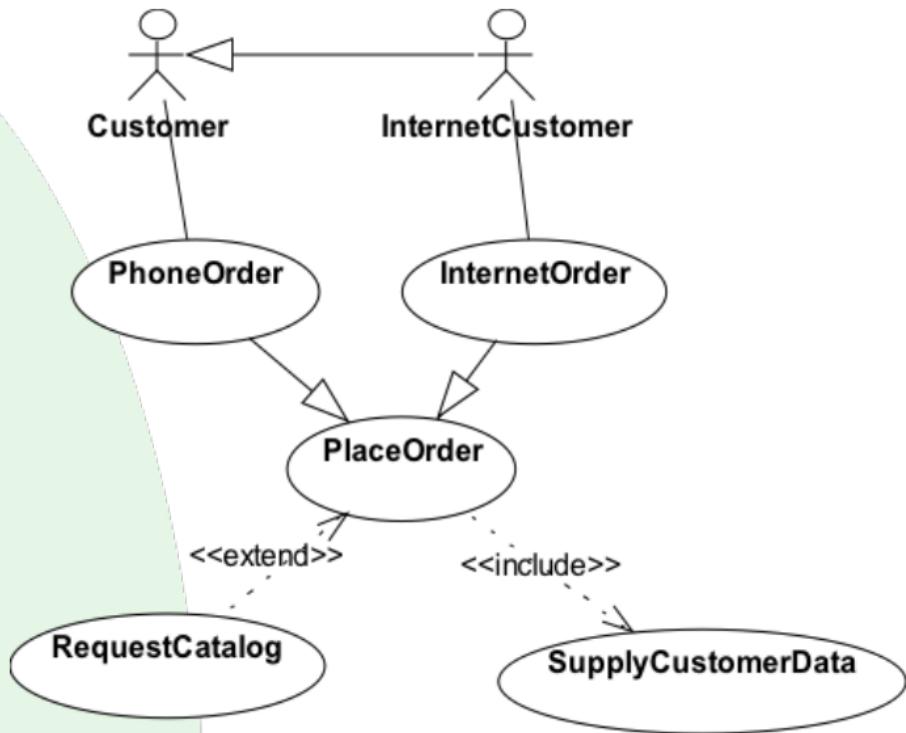
These use cases are both variations of the more general Place Order usecase, which in this example is abstract.

The **Request Catalog** use case represents an optional segment of behavior that is not part of the primary purpose of Place Order. It has been factored out to an abstract usecase to simplify the Place Order use case.

The **Supply Customer Data** use case represents a segment of behavior that was factored out since it is a separate function of which only the result is affecting the Place Order use case.

The Supply Customer Data use case can also be reused in other use cases.

Both RequestCatalog and Supply Customer Data are abstract in this example.





## 5 用例的描述

### 事件流的描述

用例代表系统和系统参与者之间发生的一系列的事件，这些事件流构成了用户对系统功能的一次使用。对事件流的描述包括四种形式，即：

形式文本

非形式文本

交互图

状态图

有相应的UML成员作为这些描述的载体

文本和正式文本可用**注标** (note) 表示，

交互图和活动图本身即是一个标准的UML成员。

- **主事件流** (main flow of events) 和**次要事件流** (alternative flow of events)。
  - 用来区分对系统功能的合法使用和非法使用
  - **主事件流** (main flow of events)
    - 合法使用，只有一个
  - **次要事件流** (alternative flow of events)
    - 可包含若干个。
- 在描绘事件流时，必须用足够清晰的语言以使得一个普通的用户能够理解。

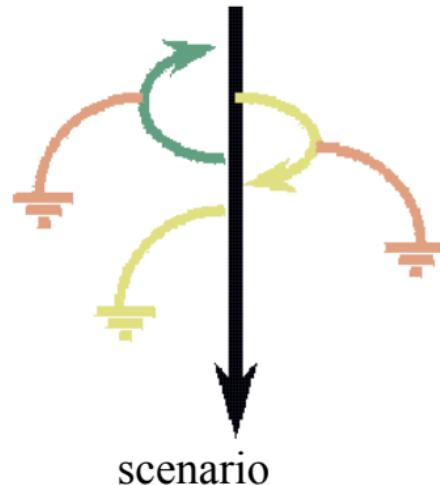
## 5 用例的描述

The two main parts of the flow of events :

- **The main flow of events**

should cover what "normally" happens when the use case is performed.

- **The alternative flows of events** cover behavior of optional or exceptional character in relation to the normal behavior.



# Scenario

- *Scenario* (use-case instance)
- A sequence of actions and interactions between actors and the system in the course of **particular** story, transaction, or use of the system
- it is also called a use case instance

# 5 用例的描述－格式1

Section title	Section content
<b>Summary</b>	Give the role, purpose and the summary of the use case.
<b>Actor List</b>	List the actor involved in the use case.
<b>Pre-condition</b>	Give the conditions that must be filled to activate the use case.
<b>Description</b>	Detail operations performed during the use case.
<b>Post-condition</b>	Give the modified states of the system, modified resources, etc ... .
<b>Exception</b>	Give the possible errors or exceptions that must be taken into account during the use case execution.

# 5 | 用例的描述－格式2

## *Login (use case)*

### Brief Description

This use case describes how a user logs into the Payroll System.

### Flow of Events

#### *Basic Flow*

This use case starts when the actor wishes to Login to the Payroll System.

The system requests that the actor enter his/her name and password

The actor enters his/her name and password.

The system validates the entered name and password and logs the actor into the system.

#### *Alternative Flows*

##### Invalid Name/Password

If in the **Basic Flow**, the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the **Basic Flow** or cancel the login, at which point the use case ends.

### Special Requirements

None.

### Pre-Conditions

None.

### Post-Conditions

If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.

### Extension Points

None.

- From our text book
- Basic definition
- use case/ identify?evaluation?
- guidelines

# Conclusion

- UML
- Use case for requirements

## 课后作业：订餐系统的用例模型

- 1.find actors
- 2.find use cases
- 3.draw use case diagram
- 4.specification use case by text format

# 作业要求

- 要求至少3个有效用例
- 10月20日提交，用例模型的word版
- 用例描述写2个，必须包含异常场景。
- 提交到大夏学堂。

- Next class: ch8~10,ch15
- Key topic located in ch9: Domain Model
- Ch10,Ch15: System sequence diagram.
  - You should already understand(!?)
    - Really?
    - Please review / preparation
- End