UML and Software Modeling

王翀

武汉大学计算机学院

Use Cases

Use Cases

"Use cases are a technique for capturing the functional requirements of a system.

Use cases work by describing the typical interactions between the users of a system and the system itself, providing a narrative of how the system is used."

Use Cases

Use case development process …

◦ Develop multiple scenarios

◦ Distill the scenarios into one or more use cases where each use case represents a functional requirement

◦ Establish associations between the use cases and actors

Use Cases

A *use case …*

◦ Specifies the behavior of a system or some subset of a <u>system</u>

◦ Is a set of scenarios tied together by a common user goal

◦ Does not indicative how the specified behavior is implemented, only what the behavior is.

◦ Performs a service for some *user* of the system.

   ◦ A user of the system is known as an *actor*.

◦ During the analysis phase, facilitates communication between the customer, users of the system and the developers of the system.

Use Cases

A *use case* …

◦ Represents a functional requirement of the system

  ◦ A *requirement* …

    ◦ Is a design feature, property, or behavior of a system

    ◦ States what needs to be done, but not how it is to be done

    ◦ Is a contract between the customer and the developer

    ◦ Can be expressed in various forms, including use cases

Use Cases

- Is graphically represented as an oval with the name of its functionality written inside.
  - Functionality is always expressed as a verb or a verb phrase.
- may exist in relationships with other use cases much in the same way as classes maintain relationships with other classes.

Use Case示例
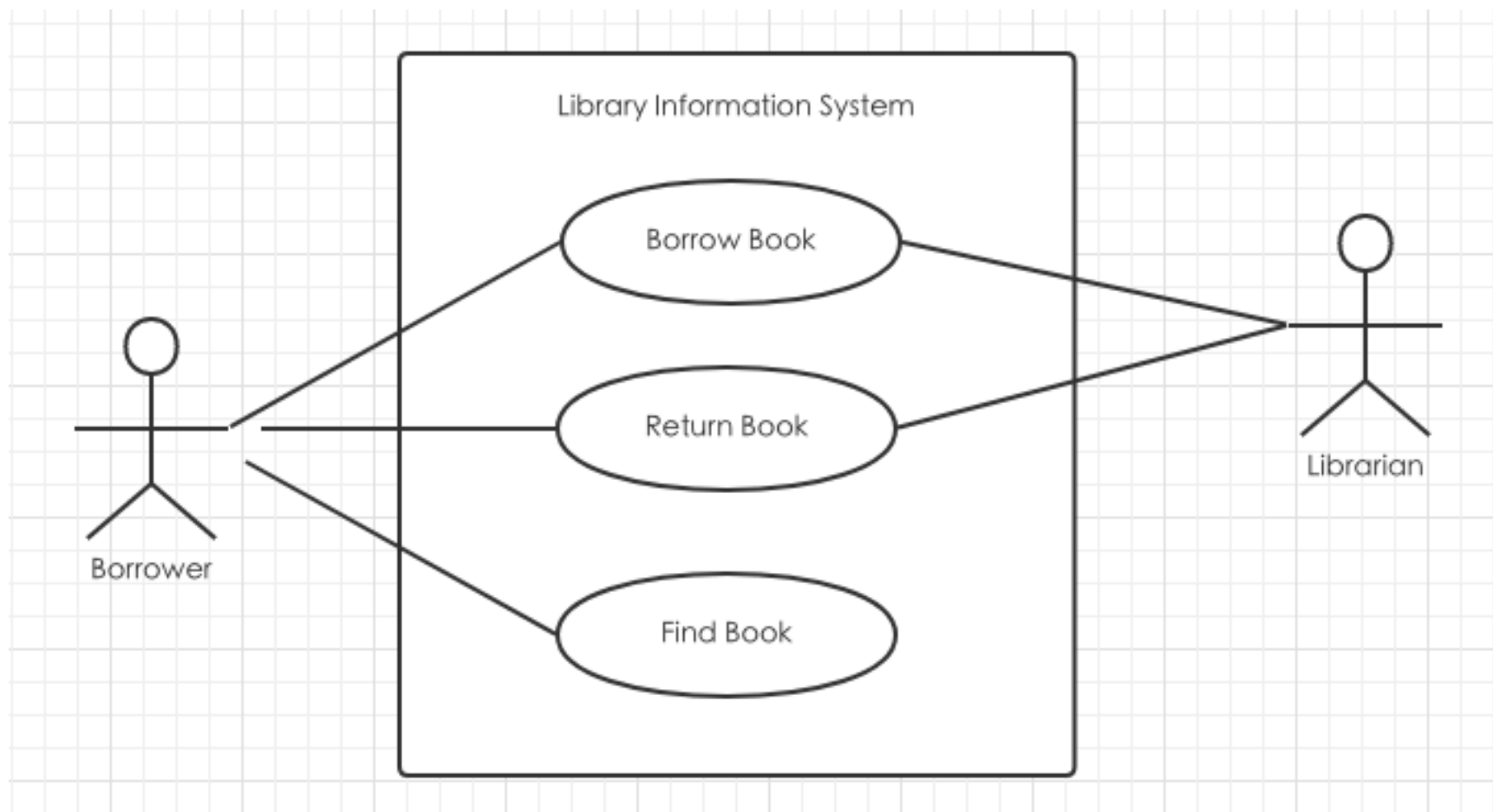
例: 字处理程序中,"置正文为黑体"
和"创建索引" 都可以是用例.

例: 在一个银行业务系统中可能
有如右的用例

置正文为黑体　　创建索引

➢ **浏览账户余额**
➢ **列出交易内容**
➢ **划拨资金**
➢ **……**

Actors

An *actor ...*

◦ is a role that the user plays with respect to the system

◦ is associated with one or more use cases

◦ does not have to be human

   ◦ Actors are roles played by people, organisations, other software and devices that interact with the application.

示例—图书馆

Actors

An *actor …*

- is a user of the system

- is most typically represented as a stick figure of a person labeled with its role name

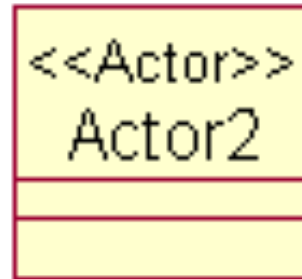 - Role names should be nouns

Actors

An actor …

- ◦ may be drawn as a stereotyped class or a graphical image of your own design

- ◦ are connected to use cases by associations
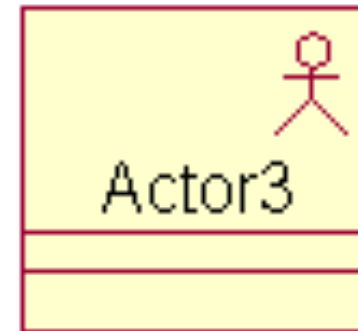
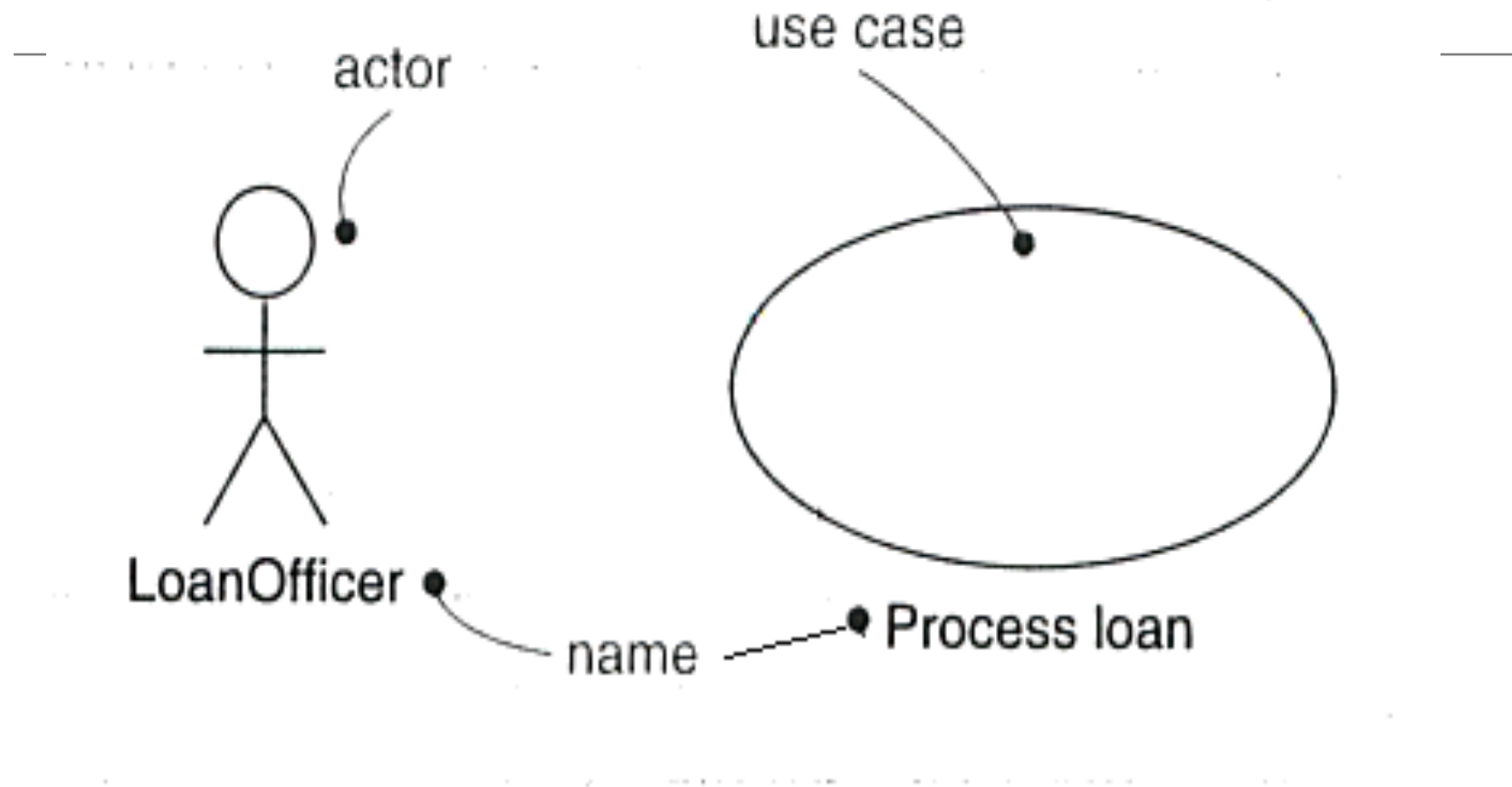- ◦ exist outside the system boundaries

参与者示例
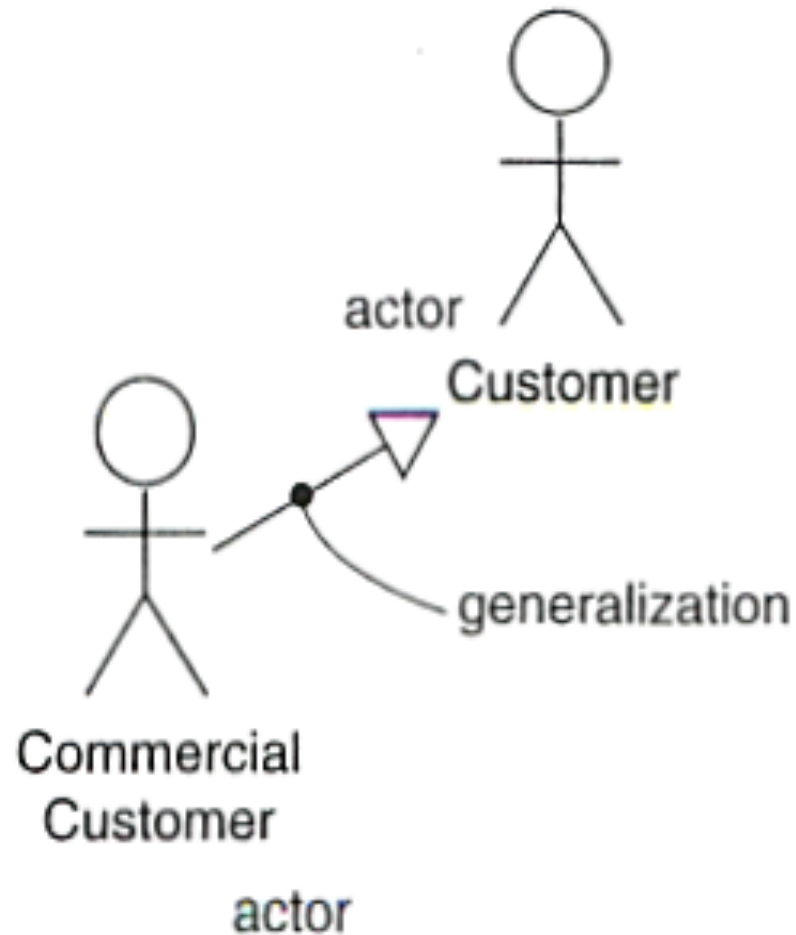

Icon形式

Label形式

Decoration形式

Actors and Use Cases

Graphical representation of an actor and a use case

Actors and Use Cases

Generalization between actors

场景(scenario)

● 在UML中指贯穿用例的一条单一路径， 用来显示用例中的某种特殊情况

● 每个用例有一系列场景，包括一个主要场景, 以及几个次要场景。

  ● 相对于主要场景，次要场景描述了执行路径中的异常或可选择的情况。

场景(scenario)

例：在"订货"用例中包括几个相关场景：

- 订货顺利进行的场景;

- 相关货源不足时的场景;

- 购货者的信用卡被拒绝时的场景;

- ……

Contents of a Use Case

Use cases and Flow of Events

- A use case, by itself, does not describe the flow of events needed to carry out the use case.
- Flow of events can be described using informal text, pseudocode, or activity diagrams.
  - May use a *note* to attach flow of events documentation to a use case.
  - Rose can link an activity diagram to a use case

Contents of a Use Case

Use cases and Flow of Events

◦ Be sure to address *exception handling* (error conditions) when describing flow of events.

◦ "The amount to detail you need in a use case depends on the amount of risk in that use case."

Use Case Diagrams

Organizing Use Cases

◦ A use case diagram is a graphical table of contents

◦ A cases may have a relationship with other use cases

   ◦ *Generalization* between use cases is used to extend the behavior of a parent use case.

   ◦ An <<include>> relationship between use cases means that the base use case explicitly incorporates the behavior of another use case at a location specified in the base.

   ◦ Sometimes the <<uses> stereotype is used instead of <<include>>

Use Case Diagrams

◦ An <<extend>> relationship between use cases means that the base use case implicitly incorporates the behavior of another use case at a location specified indirectly by the extending use case

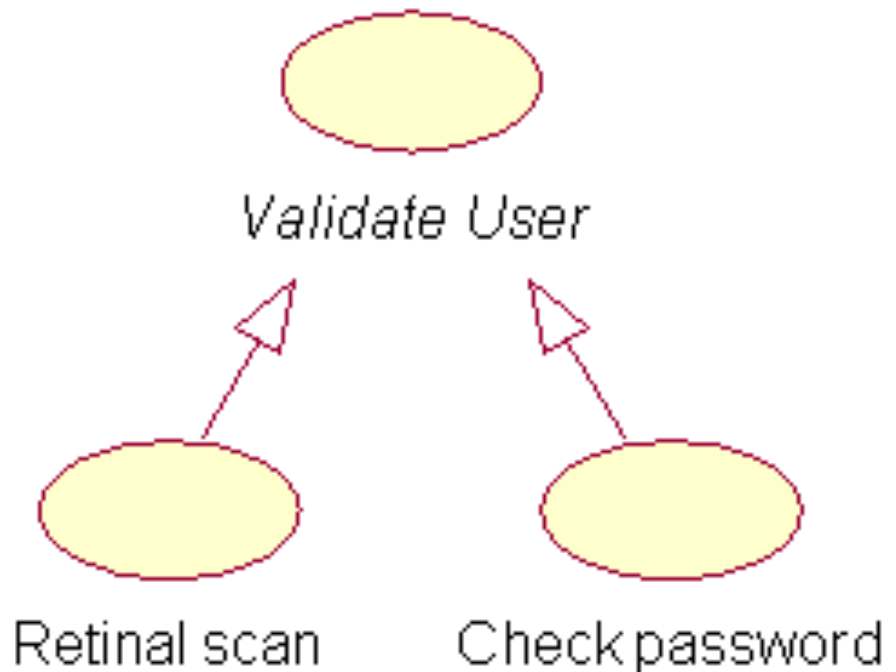◦ *Extended* behavior is optional behavior, while *included* behavior is required behavior

用例之间的关系

- 用例与参与者之间

  - 关联(association)关系

- 用例与用例之间的关系

  - 泛化(generalization)

  - 包含(include)

  - 扩展(extend)
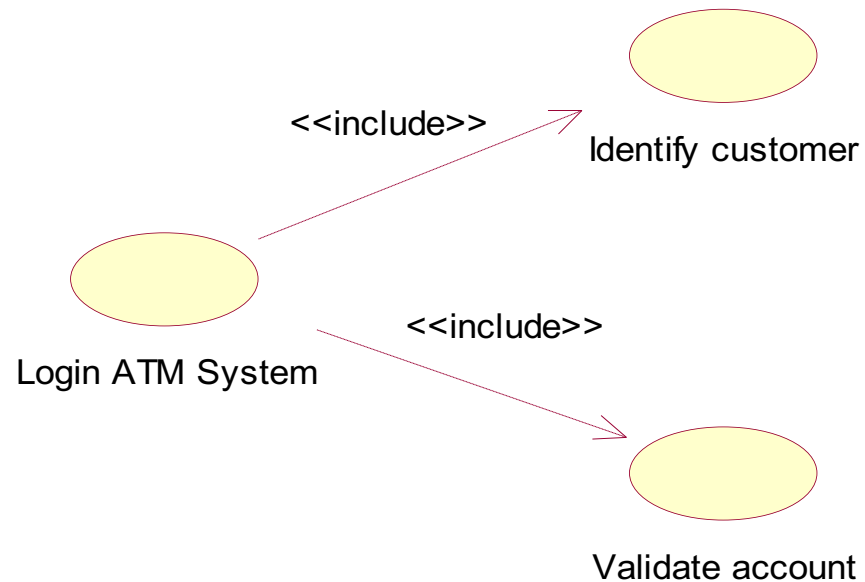
泛化关系

泛化关系代表一般与特殊的关系, 与继承类似.

在泛化关系中, 子用例继承了父用例的行为和含义, 子用例也可以增加新的行为和含义或覆盖父用例中的行为和含义.

包含关系

包含关系是指一个用例(基用例)的行为包含了另一个用例(被包含用例)的行为.

包含关系是依赖关系的版型, 但其含义更多.

<<include>> → Identify customer

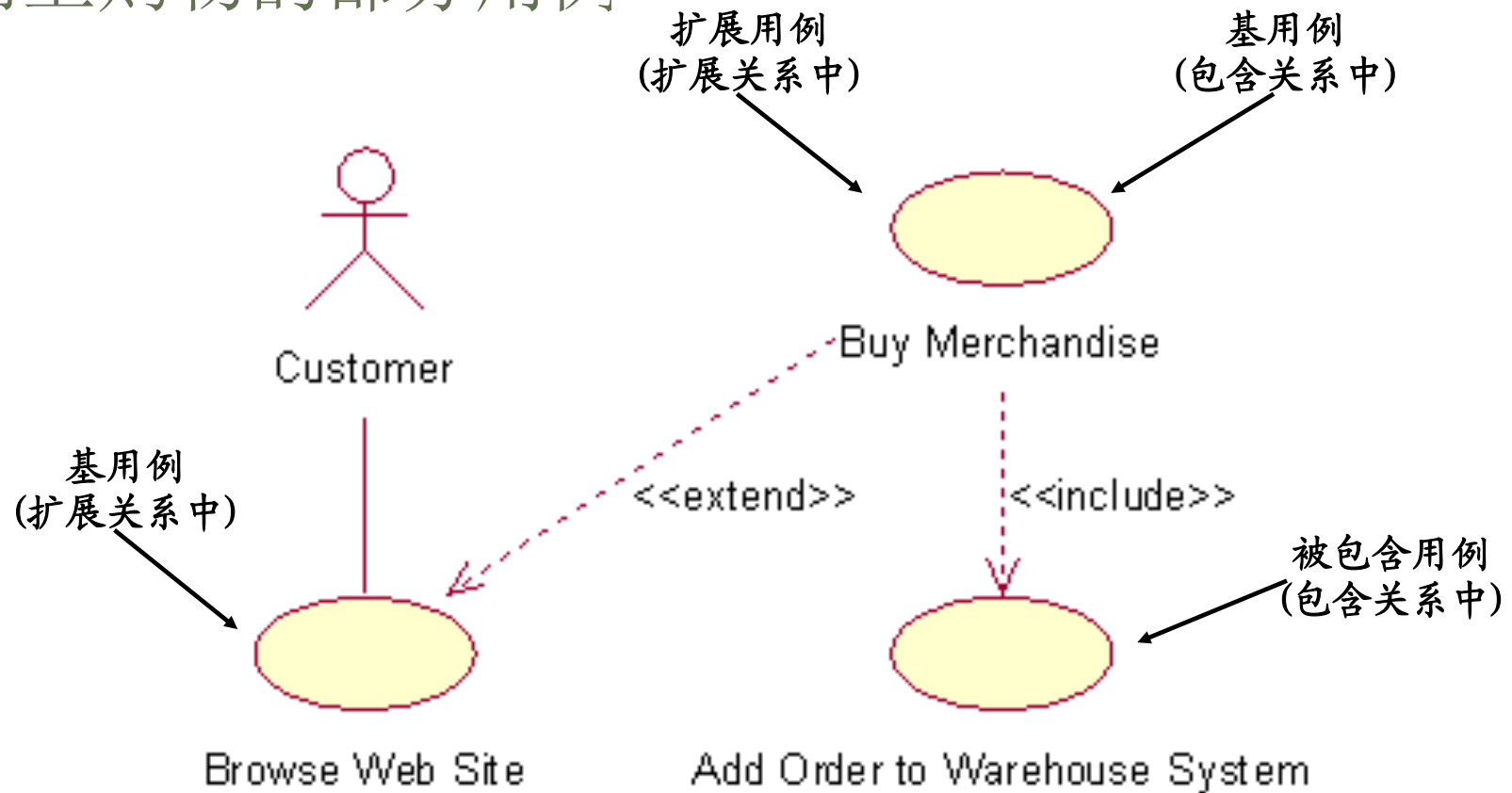Login ATM System

<<include>> → Validate account

扩展关系

扩展关系的基本含义与泛化关系类似，但对扩展用例有更多限制，即基本用例必须声明若干"扩展点"，扩展用例只能在扩展点上增加行为和含义。

- 基用例可以使用扩展用例的行为，但不是必须的；

- 扩展用例可以被基用例激活，进而将扩展用例的行为插入基用例中。

扩展关系（续）

网上购物的部分用例

扩展用例
（扩展关系中）

基用例
（包含关系中）

Customer

Buy Merchandise

基用例
（扩展关系中）

<<extend>>

<<include>>

被包含用例
（包含关系中）
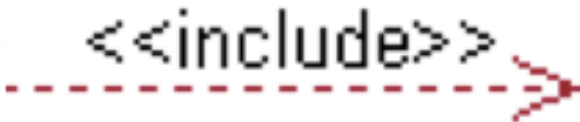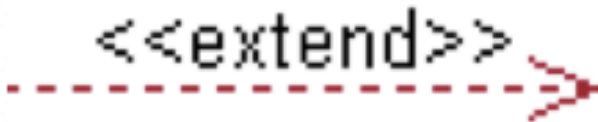
Browse Web Site

Add Order to Warehouse System

几种关系的比较

✓ 泛化和扩展表示用例之间的 "is a", 包含关系表示用例之间的 "has a".

✓ 扩展关系的基本用例是 well formed 的

  ✓ 一个基用例执行时, 可以执行或不执行扩展用例.

✓ 包含关系的基本用例可以不是或是 well formed 的

  ✓ 执行基用例时, 一定会执行被包含用例.

几种关系的比较（续）

✓需要重复处理两个或多个用例时,可以考虑包含关系.

✓处理正常行为的变型且只是偶而描述时,可以考虑只使用泛化关系.

✓处理正常行为的变型且希望采用更多控制方式时,可以在基本用例中设置扩展点,使用扩展关系.

关系类型	说明	表示符号
关联	actor与use case之间	——————————→
泛化	actor之间或use case之间	——————————▷
包含	use case之间	<<include>> - - - - - - →>
扩展	use case之间	<<extend>> - - - - - - →>

Use Case Diagrams
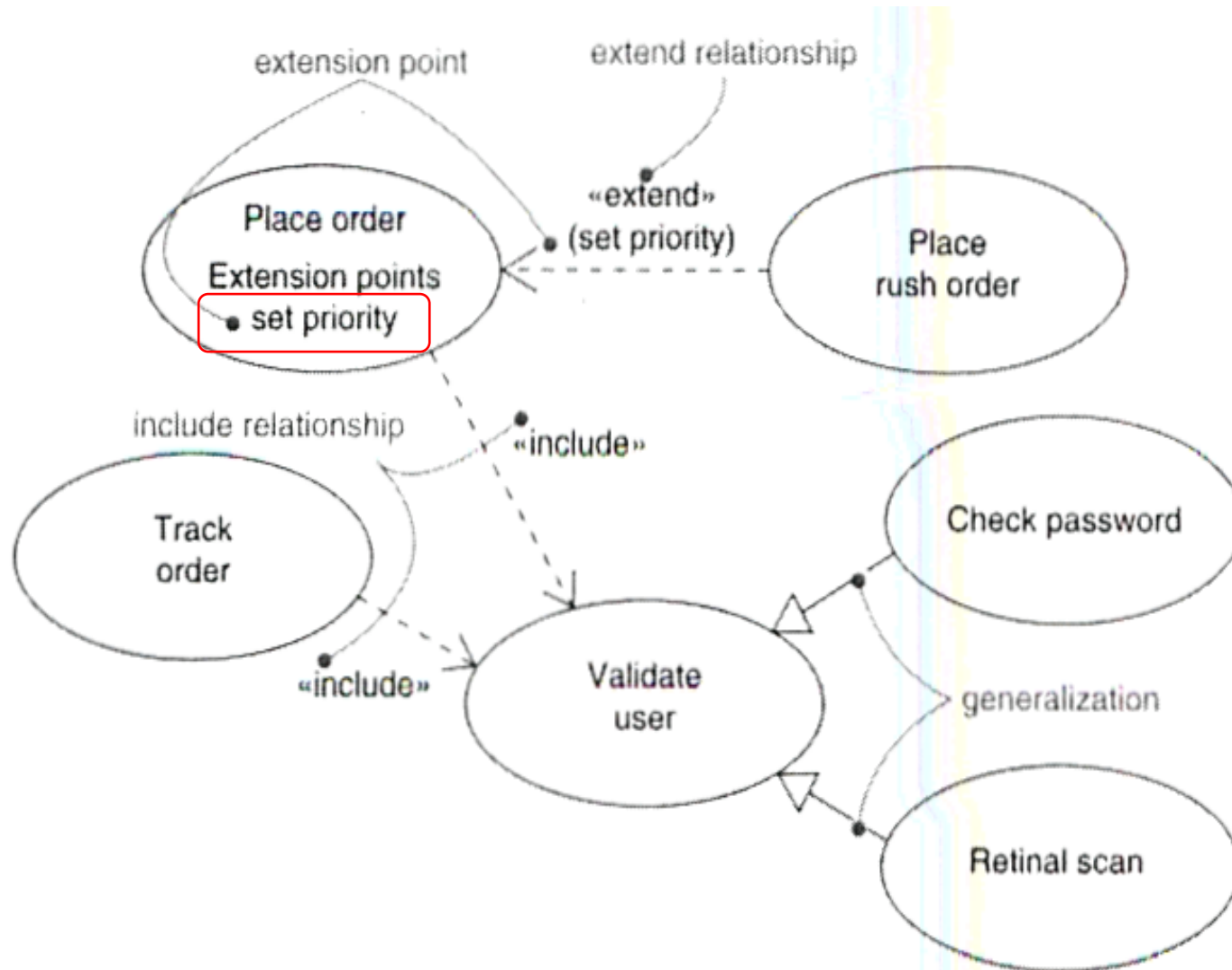
Use Case Diagrams
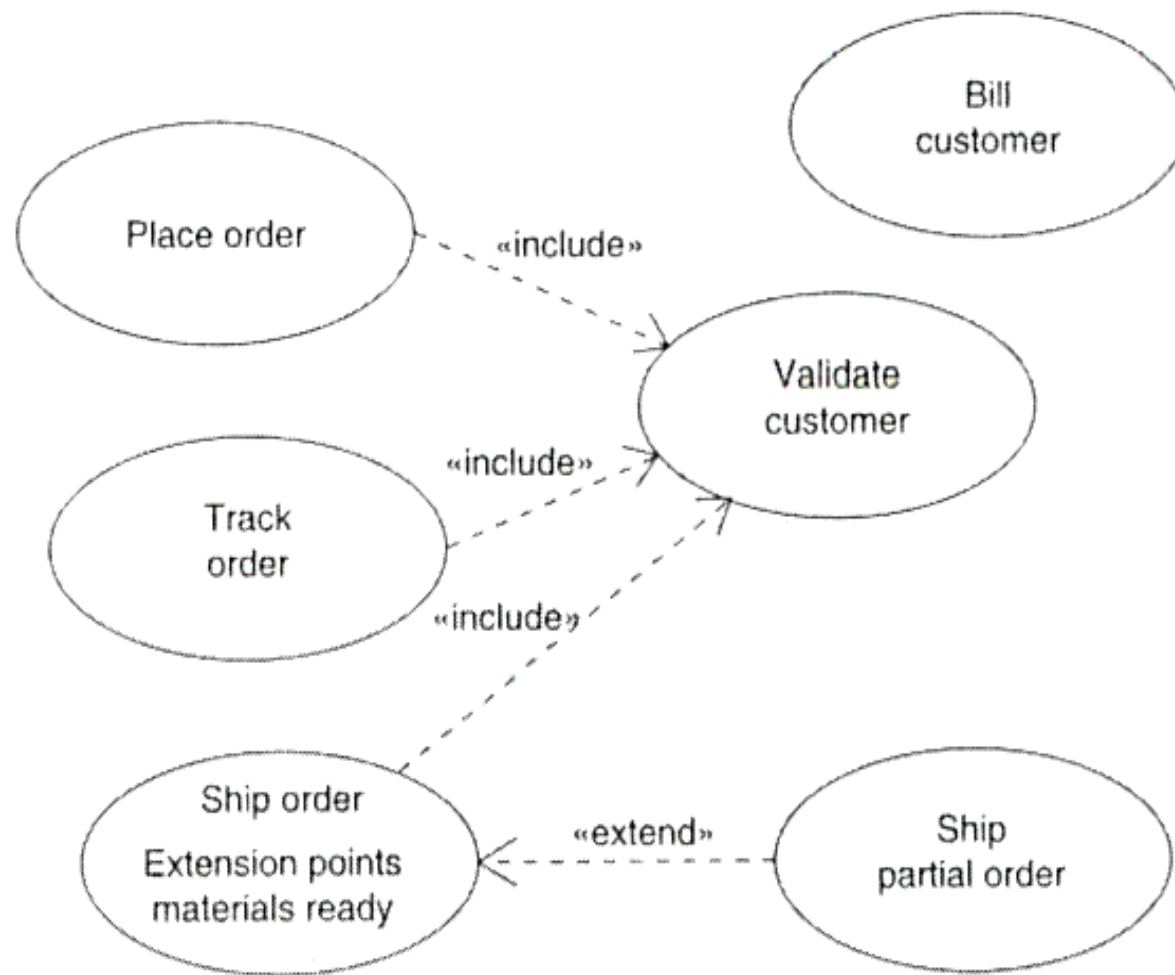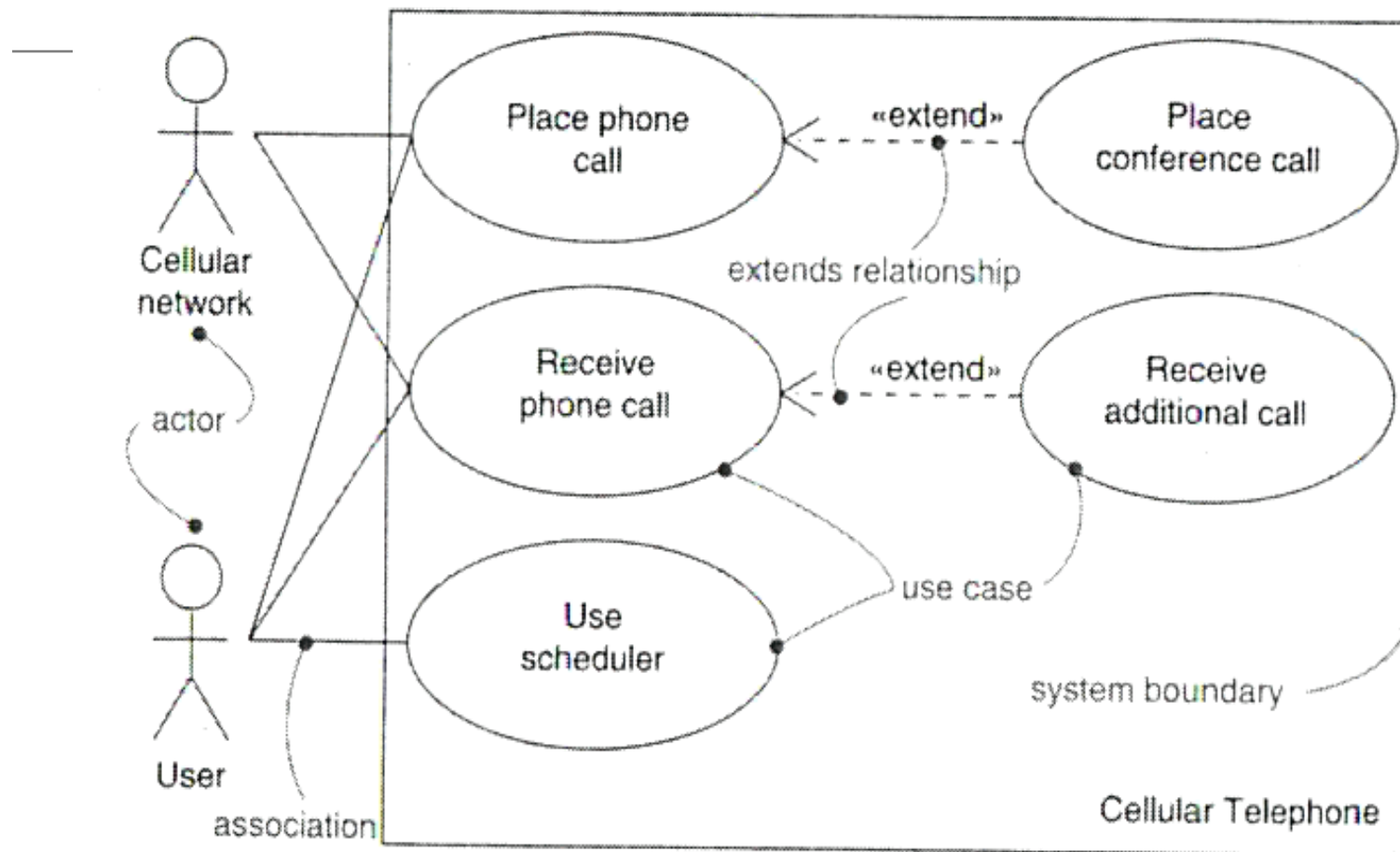
Use Case Diagram

Figure 17-1: A Use Case Diagram

Use Case Diagrams

Credit Card Validation System

Customer

Individual customer

Corporate customer

Perform card transaction

Process customer bill

Reconcile transactions

Manage customer account

Retail institution

Sponsoring financial institution
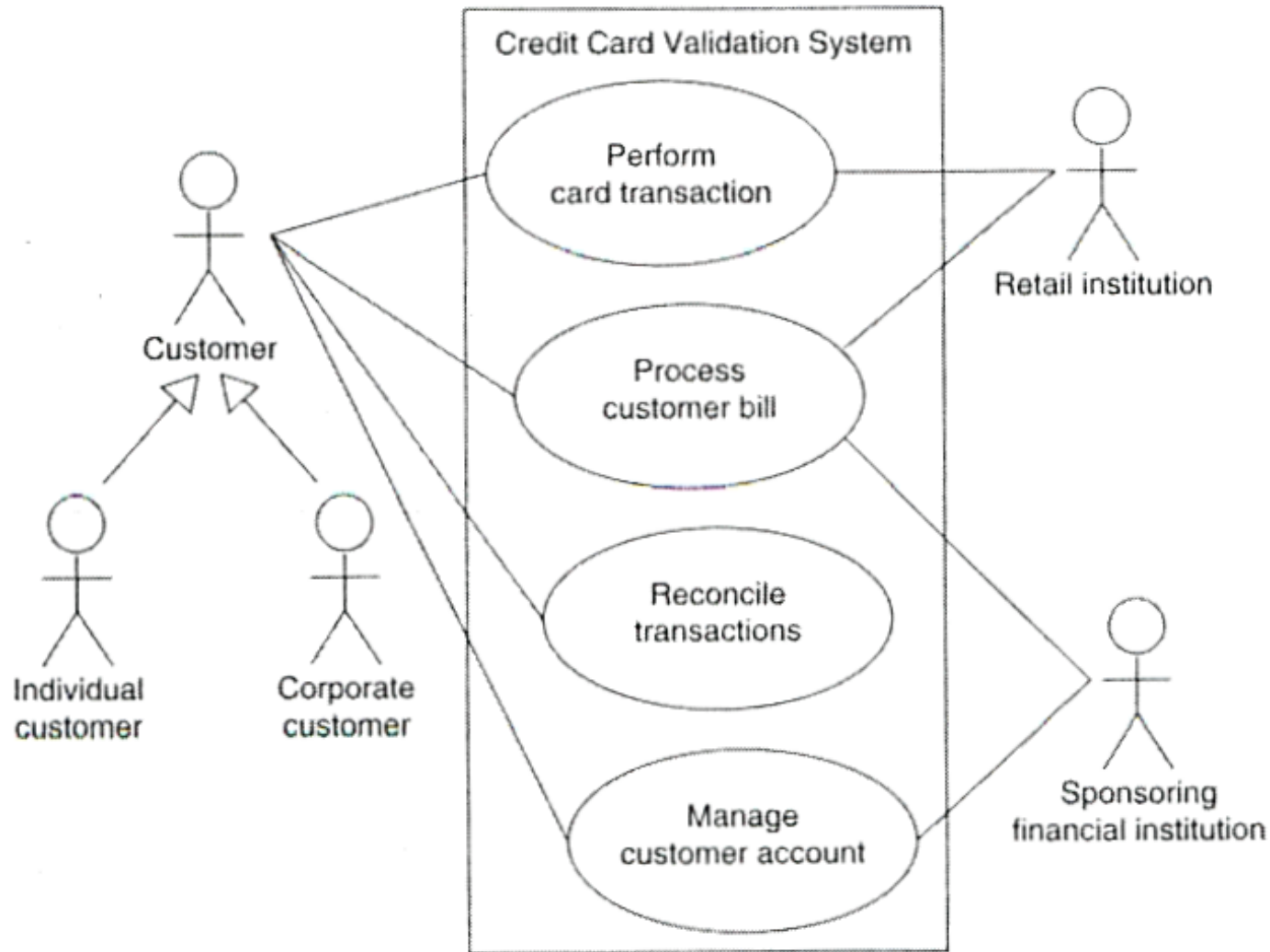
Modeling the Context of a System

用例的描述

用例的描述

用例描述是指对一个用例的功能进行的文字描述，是参与者与系统交互动作序列的说明。

> ⓘ 用例描述才是用例的主要部分，是后续的交互图分析和类图分析必不可少的部分.

用例采用自然语言描述参与者与系统的交互行为，要易于理解。

其读者是开发人员、用户、项目经理、测试人员等。

What are Three Common Use Case Formats?

Use cases can be written in different formats and levels of formality:

- **Brief Use Case** : Terse one-paragraph summary, usually of the main success scenario. Created during early requirements analysis, to get a quick sense of subject and scope. May take only a few minutes to create;

- **Casual Use Case** : Informal paragraph format. Multiple paragraphs that cover various scenarios. A refinement of a brief use case;

1.6.1 Defining Use Cases

Understanding the requirements includes understanding the domain processes and the external environment (external actors who participate in processes).

Use Case:		Play a Game

Actors:		Player

Description:	Player requests to roll the dice. System presents results. If the dice face value totals seven, player wins; otherwise player loses.

What are Three Common Use Case Formats?

Use cases can be written in different formats and levels of formality:

- ◦ **Fully Dress*e*d Use Case** : All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees. Cretaed after many use cases have been identified and written in a brief format, then during the first requirements workshop a few (such as 10%) of the architecturally significant and high-value use cases are written in this format.

用例的描述格式－－详述用例

描述项	说明
用例名称	表明用户的意图或用例的用途
标识符[可选]	惟一标识符, 便于引用该用例
用例描述	概述用例的几句话
参与者	与此用例相关的参与者
优先级	一个有序的排列, 1代表优先级最高
状态[可选]	用例状态, 可以是: 进行中, 等待审查, 通过审查, 未通过审查
前置条件	一个条件列表, 这些条件必须在访问用例前得到满足
后置条件	一个条件列表, 这些条件必须在用例完成之后得到满足
基本操作流程	描述用例中各项工作都顺利进行时用例的工作方式
可选操作流程	描述变异工作方式、出现异常或发生错误的情况下的路径

用例的描述格式－详述用例(续表)

描述项	说明
被泛化的用例	此用例所泛化的用例列表
被包含的用例	此用例所包含的用例列表
被扩展的用例	此用例所扩展的用例列表
修改历史记录[可选]	关于用例的修改时间、修改原因、修改人的详细信息
问题[可选]	与此用例的开发有关的问题列表
决策[可选]	关键决策的列表, 将这些决策信息记录下来以便维护时使用
频率[可选]	参与者访问此用例的频率, 如: 每日一次/每月一次等

例: 用例 "请假" 的描述

Example: Process Sale, Fully Dressed Style

ROUGHLY 10% OF THE USE CASES SHOULD BE WRITTEN IN THIS STYLE DURING THE INCEPTION PHASE.

用例描述时易出现的错误

➢ 只描述系统的行为，没有描述参与者的行为

➢ 只描述参与者的行为，没有描述系统的行为

➢ 在用例描述中就设定了对用户界面的设计的要求

➢ 描述过于冗长

ATM系统"取款"用例的两个错误描述：

Use case: Withdraw cash

Actor: customer

主事件流：

(1) 储户插入ATM卡,并输入密码

(2) 储户按"取款"按钮,并输入取款数目

(3) 储户取走现金/ATM卡/收据

(4) 储户离开

只描述了**actor**的行为

Use case: Withdraw cash

Actor: customer

主事件流：

- ATM系统获得ATM卡和密码
- 设置交易类型为"取款"
- ATM系统获得取款金额
- 输出现金、收据和ATM卡
- 系统复位

只描述了**System**的行为

ATM系统"取款"用例的正确描述：

Use case: Withdraw cash

Actor: customer

主事件流：

- 储户通过读卡机插入ATM卡
- ATM系统从卡上读取银行ID、账号、加密密码，并通过主银行系统验证银行ID和账号
- 储户输入密码，ATM系统根据加密密码对输入密码进行验证
- 储户按"取款"按钮，并输入取款数目，该数目应该为$5的倍数
- ATM系统通知主银行系统，传递账号和金额，并接收返回的确认信息和账户余额
- ATM系统输出现金、ATM卡和收据
- ATM系统记录交易到日志文件