

第8章 面向对象分析

- 面向对象分析概述
- 建立功能模型（用例模型）
- 建立静态模型（对象模型）
- 建立动态模型

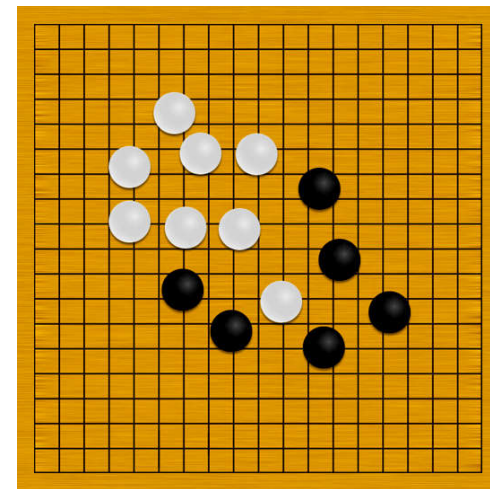
面向对象分析概述

SA、SD与OOA、OOD之对比

功能：从功能入手，分解功能为各函数。

过程：下棋的过程是通过对函数的调用来体现。

数据：棋子是作为函数（功能）的传递因素（参数），在棋盘落子来实现棋子的布局。



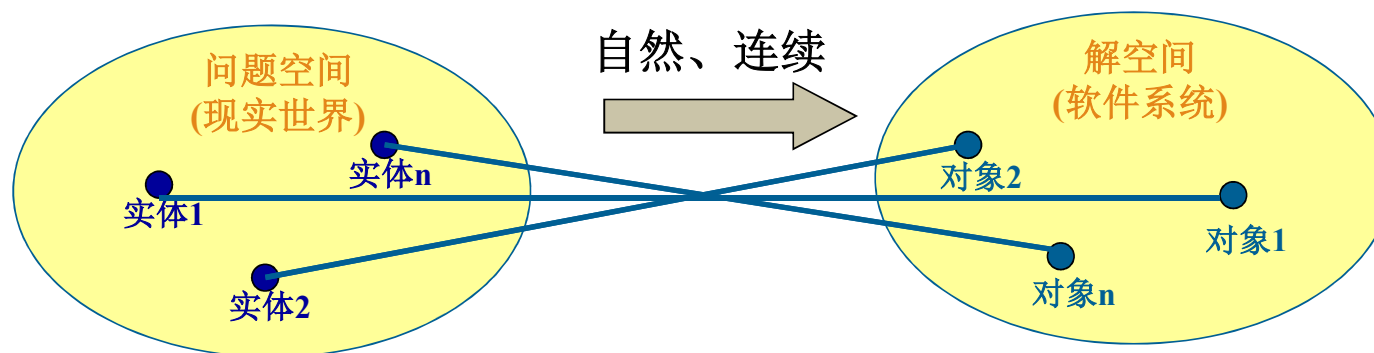
对象：在下棋过程中，涉及实体对象有棋子、棋盘，以及抽象对象“规则”。

过程：下棋过程是通过对象间消息传递来完成。例如：每次落子之后，均把“棋子”对象发送给“规则”对象，由“规则”结合“棋盘”对象去判断合理（落子是否有效）、胜负等关系。

扩展：如果修改规则，不会影响“棋子”和“棋盘”。不修改结构，不影响对象间消息的传递。

面向对象分析概述

面向对象的本质



在OOA过程中:

- 现实世界的实体直接对应信息世界的对象;
- 信息世界的对象再抽象、聚合、新生、去除。

面向对象分析概述

面向对象分析的基本过程

OOA是以**类**和**对象**为基础，以面向对象方法学为指导，分析用户需求，并最终建立问题域的准确模型的过程。和结构化需求分析一样，**OOA**也从问题定义入手，获取用户需求，建立功能模型、静态模型和动态模型。

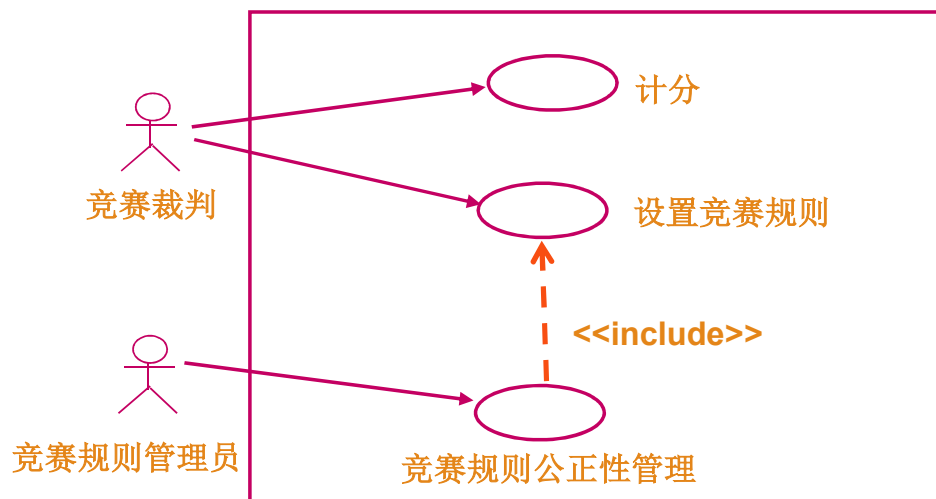
- 通过对用户需求陈述的分析，**建立功能模型**以体现用户将要实现的功能。
- 通过提取问题定义中的实体得到类和对象，并**建立静态模型**。
- 结合功能模型和静态模型，定义类和对象的内部表示和外在联系，**建立动态模型**。
- 将用自然语言描述的用户需求转换为功能模型、静态模型和动态模型共同刻画的结构、功能定义和性能描述。

建立功能模型

功能模型通过识别需求中的用例来描述用户功能需求，目的是分析和建立用户功能的需求信息，因此也称为用例模型。

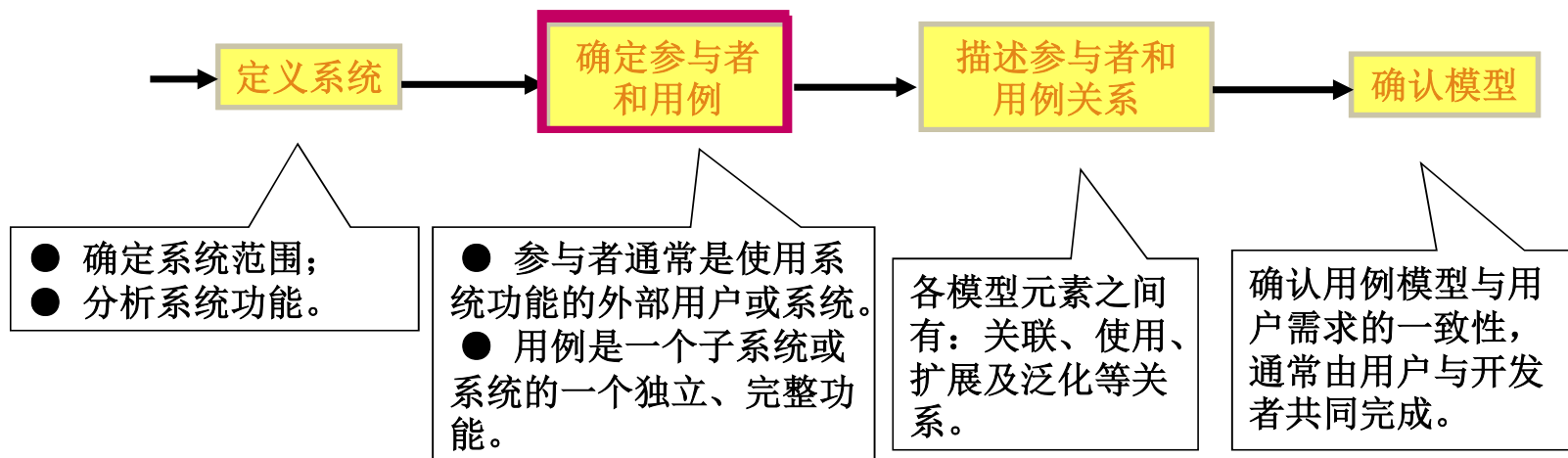
- 用例建模技术用于描述系统与用户的交互和功能需求。通过对典型用例的分析，使开发者能够有效地了解用户的需求。
- 用例模型由**若干个**用例图构成。
- 用例图描述参与者、用例以及它们之间的关系。

竞赛计分子系统



建立功能模型

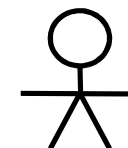
建立系统用例模型的过程就是对系统进行功能需求分析的过程。



建立功能模型

一、确定参与者(Actor)

执行者是指用户在系统中所扮演的参与者执行者用类似人的图形来表示，但执行者可以是人，也可以是一个外界系统。

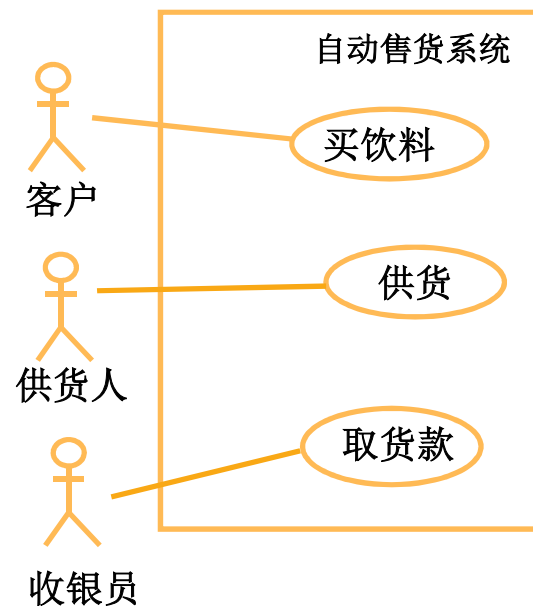
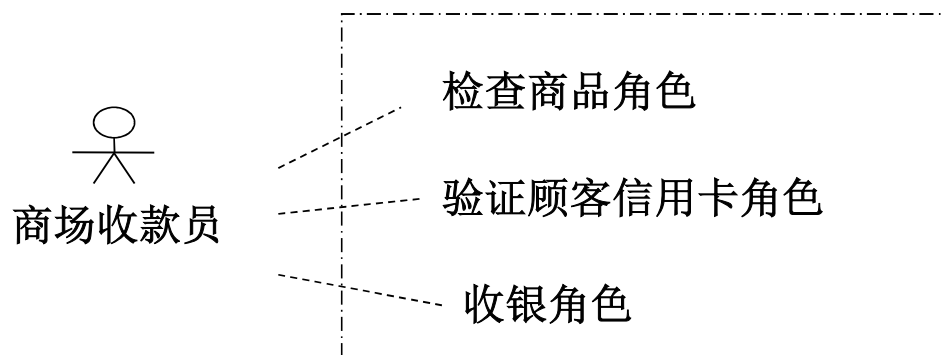


如何确定参与者：

- 1、谁使用系统的主要功能(主执行者)?
- 2、谁需要从系统获得对日常工作的支持和服务?
- 3、需要谁维护管理系统的日常运行(副执行者)?
- 4、系统需要控制哪些硬件设备(外部执行者)?
- 5、系统需要与其它哪些系统交互?
- 6、谁需要使用系统产生的结果(值)?

建立功能模型

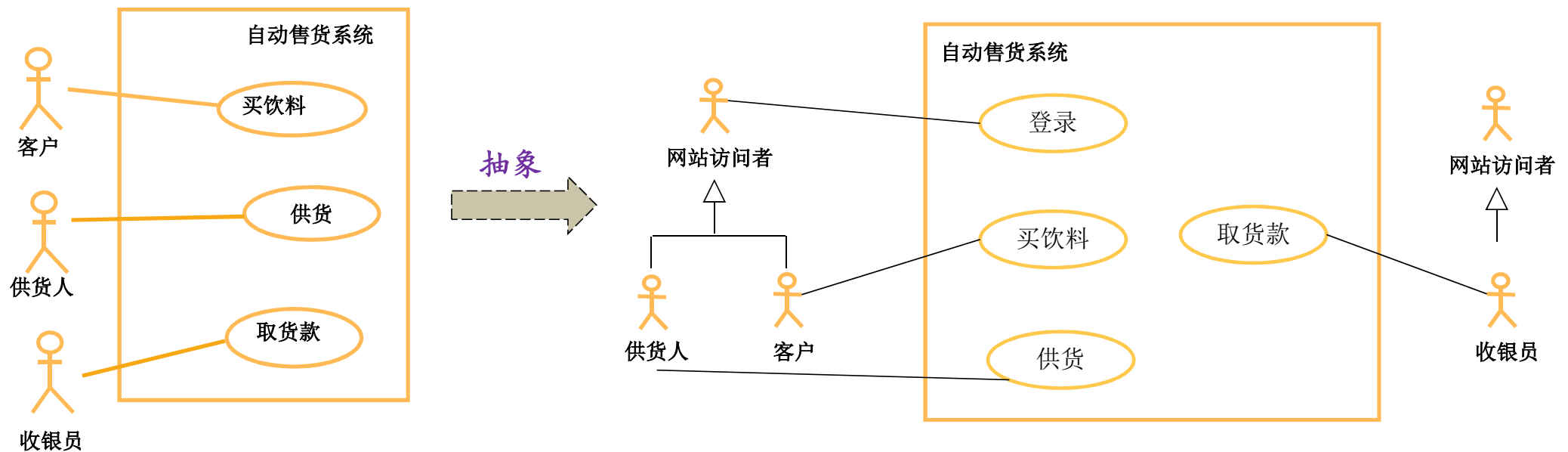
参与者定义：一个**参与者**定义了一组在功能上密切相关的**使用角色**。当一个事物与系统交互时，该事物可以扮演这样的**角色**。



- **参与者**可以请求系统提供服务，也可以接受系统的要求，并做出响应。
- **参与者**不是系统的一部分，它们位于系统之外。

建立功能模型

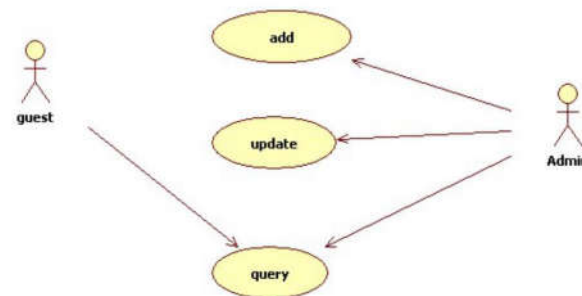
参与者**继承关系**定义：如果一组参与者有共同的性质，把这些性质抽取在一个参与者中，这组参与者再从中继承。



表示：客户、供货人、收银员对系统的请求有部分相同用例。

建立功能模型

二、建立用例(use case)



从本质上讲, 一个用例是用户与计算机之间的一次典型交互。在UML中, 用例被定义成系统执行的一系列动作（功能），但不分解动作。

用例有以下特点:

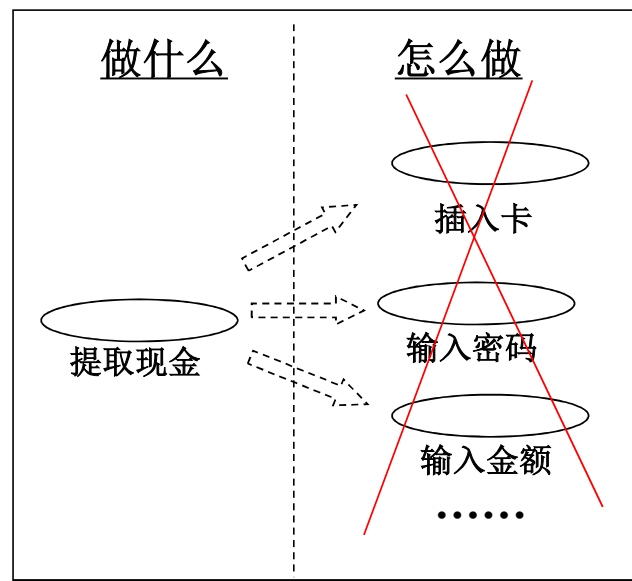
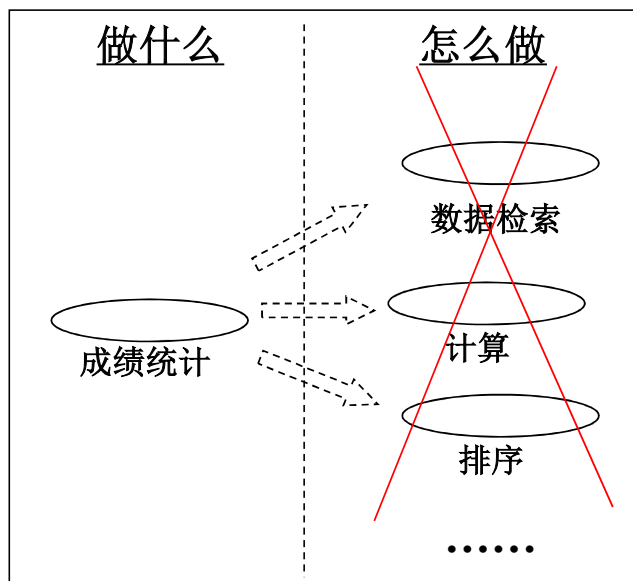
- 用例实现一个具体的用户目标;
- 用例由执行者激活, 并将结果值反馈给执行者;
- 用例必须具有功能上的完整描述。

建立功能模型

建立Use Case，从系统边界入手

系统边界，是指系统内部成分与系统以外事物的分界。因此，建立Use Case图主要是反映与将来系统进行交互的人员、设备或其他系统。

用例描述系统做什么（功能需求），而不描述怎么做。



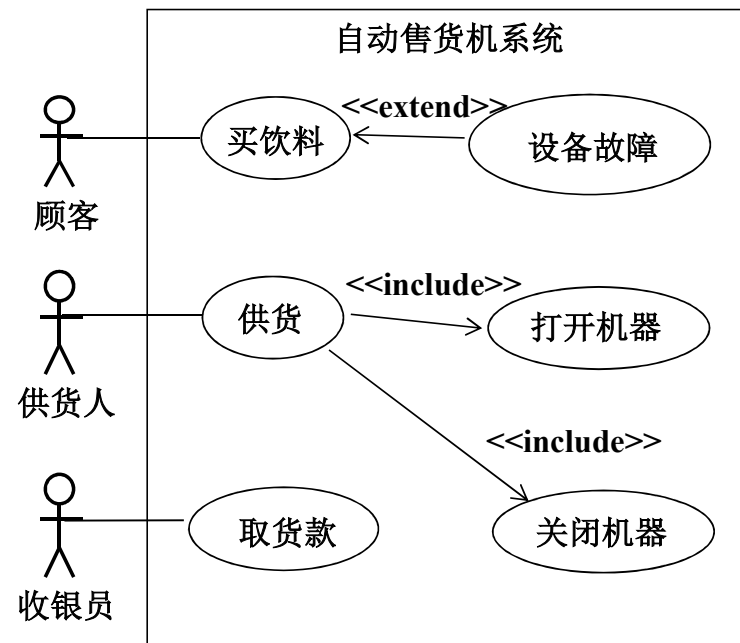
建立功能模型

三、用例之间的关系

参与者与用例之间通常是一种关联。

用例之间的主要关联：

- <<extend>>：通过向被扩展的用例添加动作来扩展用例。
- <<include>>：表示一个用例的行为包含了另一个用例的行为。



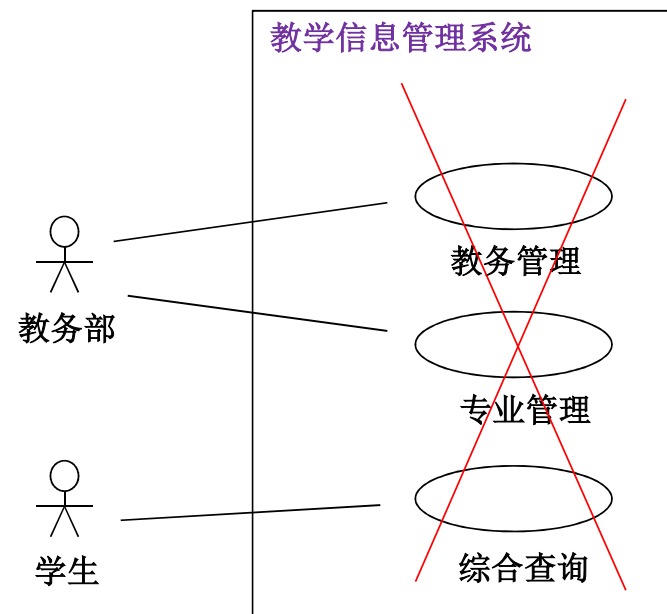
含有使用和扩展关系的用例图

建立功能模型

捕获用例的原则

1) 一个用例只描述一个功能，但用例功能不能太笼统，且缺乏与外部的交互。

一般不宜进行太抽象（笼统）的系统功能划分

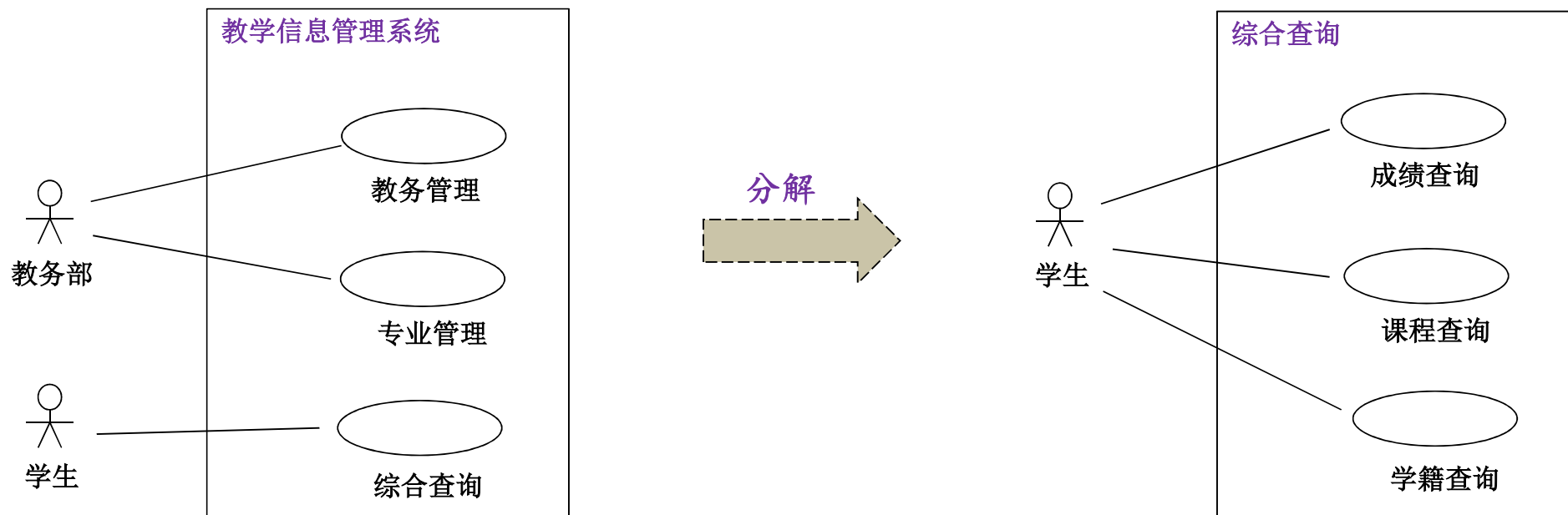


建立功能模型

捕获用例的原则

1) 一个用例只描述一个功能，但用例功能不能太笼统，且缺乏与外部的交互。

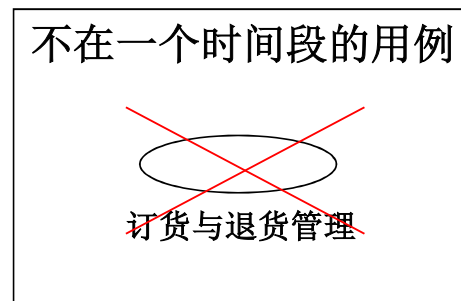
但可以对太（抽线）笼统的系统进行功能分级划分



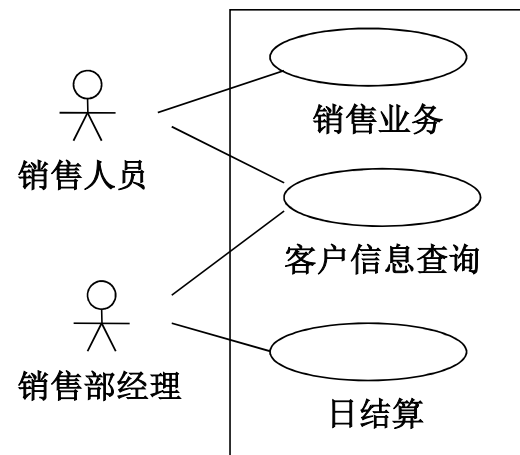
建立功能模型

捕获用例的原则

2) 一个用例是在一个相对完整的时间段中发生的，应尽量避免一个用例涉及多个时间段。



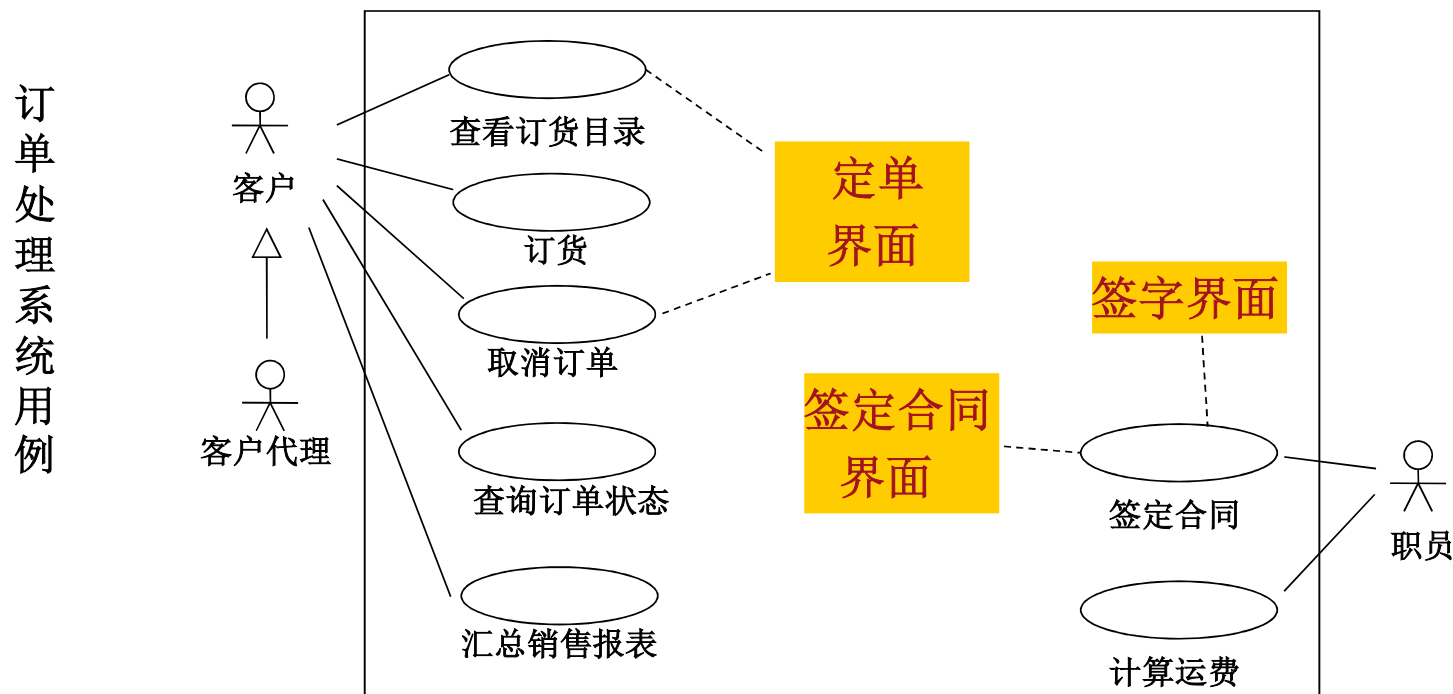
3) 一个参与者可以对应多个用例，一个用例也可以对应多个参与者。



建立功能模型

捕获用例的原则

4) 用例不是界面，界面也不是用例。一个用例可以对应多个界面，一个界面也可能对应多个用例。

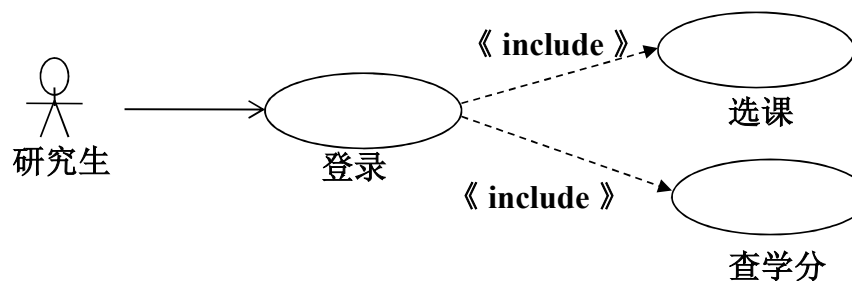


建立功能模型

用例图分析：研究生教务系统——同一用例的不同用例图分析

对登录、选课、查学分等功能，其用例描述的四种不同表示，分别对应四种不同的工作方式。

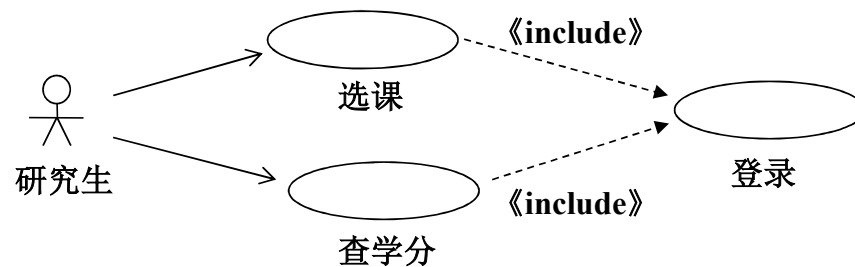
用例图1：



上图说明研究生在登录时，需要包括“选课”和“查学分”的功能。两个功能作为登录主程序的从属功能，并且都是必须要执行的功能。不符合一般处理逻辑的理解。

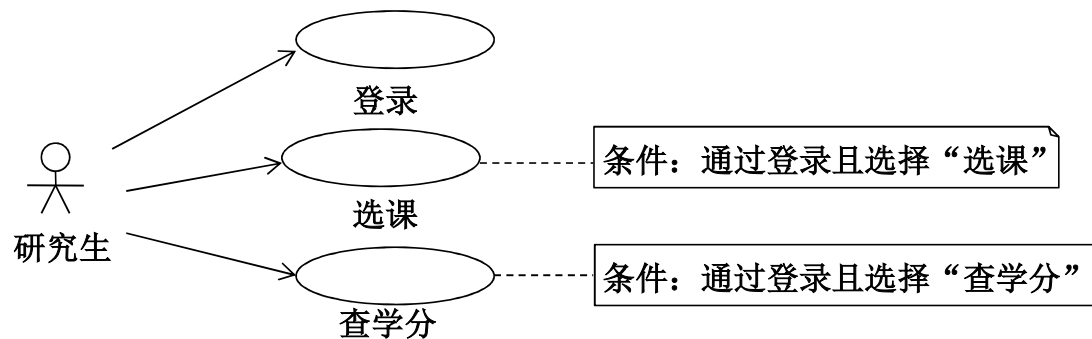
建立功能模型

用例图2:



说明研究生可选择两个用例，当进入每个用例时，都必须登录。

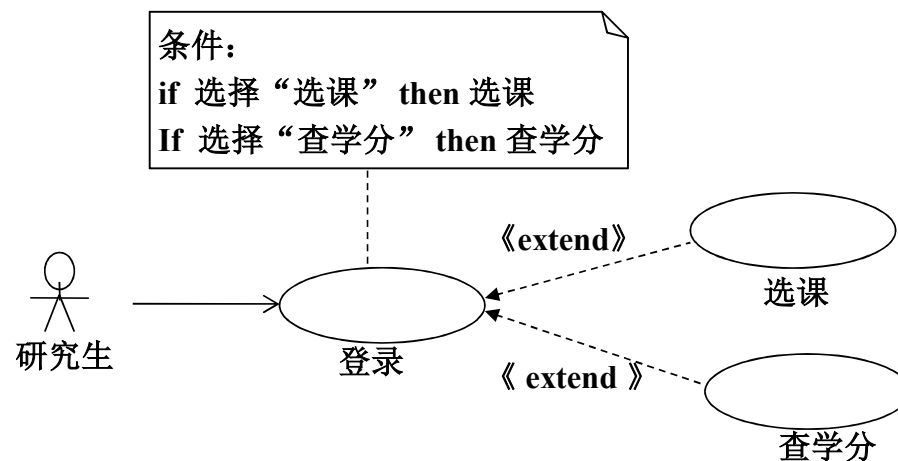
用例图3:



研究生有三个用例，它们之间不体现包含关系。

建立功能模型

用例图4:



说明研究生在“登录”后，可以选择执行“选课”或“查学分”。

但“选课”、“查学分”是研究生的主要用例，并且它们与“登录”用例不存在包含关系，因此不作为登录的扩展用例。

建立功能模型

用例图文档说明框架

用例图综述：研究生通过选课，确定下学期的选修课课程。

参与者：研究生

用例名称：选课

基本事件流：

1. 研究生浏览可选课程列表；
2. 研究生从课程列表中选择课程；
3. 系统保存选课信息。

扩展事件流：

1. 研究生在选课后24小时之内可以退课；
2. 研究生可以在第一次课后，选择退课。

关系描述：“选课”用例的扩展事件，可通过“退课”用例来支持。

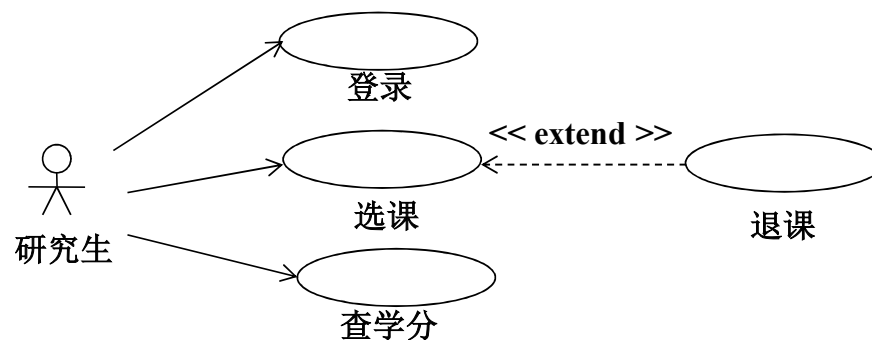
前置条件：研究生需要预先有账号（账号是在后端，由学校建立）。

后置条件：无。

异常：无。

限制：对“选课”用例：每学期选修课不能超过2门；

对“退课”用例：每学期只能退课一次。



建立功能模型

课堂练习

良乡校区理科教学楼一层有自动饮料售货机，出售N种不同类型的饮料，售货机上的N个按钮，分别对应不同的饮料。学生付款后通过按按钮来选择所要的饮料。每个按钮旁有一个指示灯，用来表明是否还有这种饮料可售。售货机有一个纸币口和硬币槽，用于收款和退款。

课后有一位同学购买饮料，请给出该同学购买饮料的用例图。

建立静态模型

- 用例模型分别从参与者及基于领域的软件系统角度描述用户需求，依据用例模型导出用例模型（功能模型）。
- 静态模型（对象模型、领域模型）通过建立类图及关系来反映领域概念，在面向对象设计阶段还需再次精化类图，但在面向对象设计阶段对类的抽象和分解程度不同。
- 静态建模的任务是构建问题域的概念模型，把问题域中的实体转变为信息域类与对象，并描述它们之间的关系。

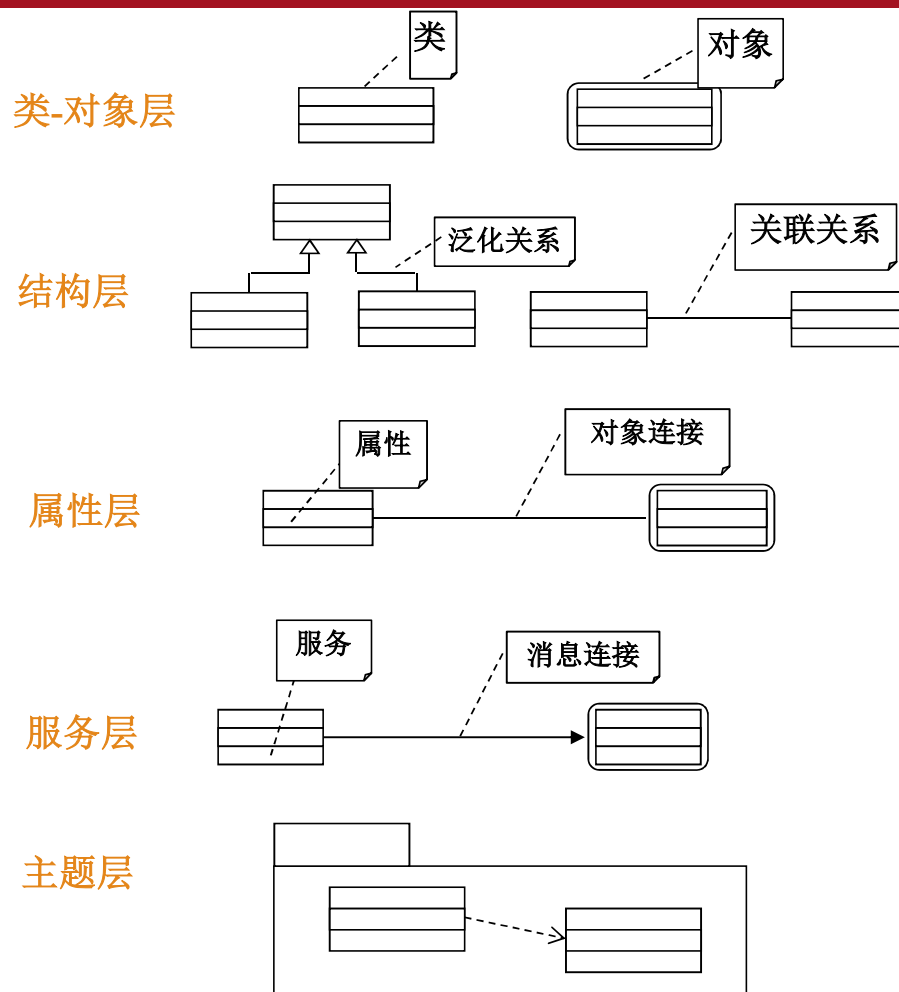
建立静态模型

静态模型的5个层次

Coad和Yourdon提出，对于复杂的大型软件系统，应建立分析问题域的静态模型。

该模型由5个层次为：

- 类 - 对象层
- 结构层
- 属性层
- 服务层
- 主题层



建立静态模型

建立静态模型的过程

➤ 识别类与对象

- 找出候选的类与对象
- 审查与筛选
- 类的命名

➤ 确定属性

➤ 确定服务

➤ 确定结构

- 确定关联关系
- 确定泛化关系
- 确定依赖关系
- 确定实现关系

➤ 划分主题

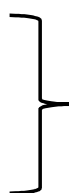
➤ 类图 / 包图的文档描述

建立静态模型

类的识别

类的识别是面向对象方法的一个难点，但又是建模的关键。

常用的方法有：

- 1. 名词识别法
 - 2. 系统实体识别法
 - 3. 从用例中识别类
 - 4. 利用分解与抽象
- 

建立静态模型

1. 类的识别——名词识别法

识别问题域中的实体，实体的描述通常用名词、名词短语、名词性代词的形式出现。

- 用指定语言对系统进行描述；
- 从系统描述中标识名词、名词短语、名词性代词；
- 识别确定（取、舍）类。

描述过程应与领域专家共同合作完成，并遵循问题域中的概念和命名。

defontology 国家本体

{

.....	
属性: 人口总数	属性: 陆地面积
类型 整数	类型 实数
值型 单值	值型 单值
时变性 连续变化	单位 平方公里
属性: 男性人口	时变性 稳定
类型 整数	属性: 水域面积
值型 单值	类型 实数
时变性 连续变化	值型 单值
属性: 女性人口	单位 平方公里
类型 整数	时变性 稳定
值型 单值	时变性 连续变化
.....	

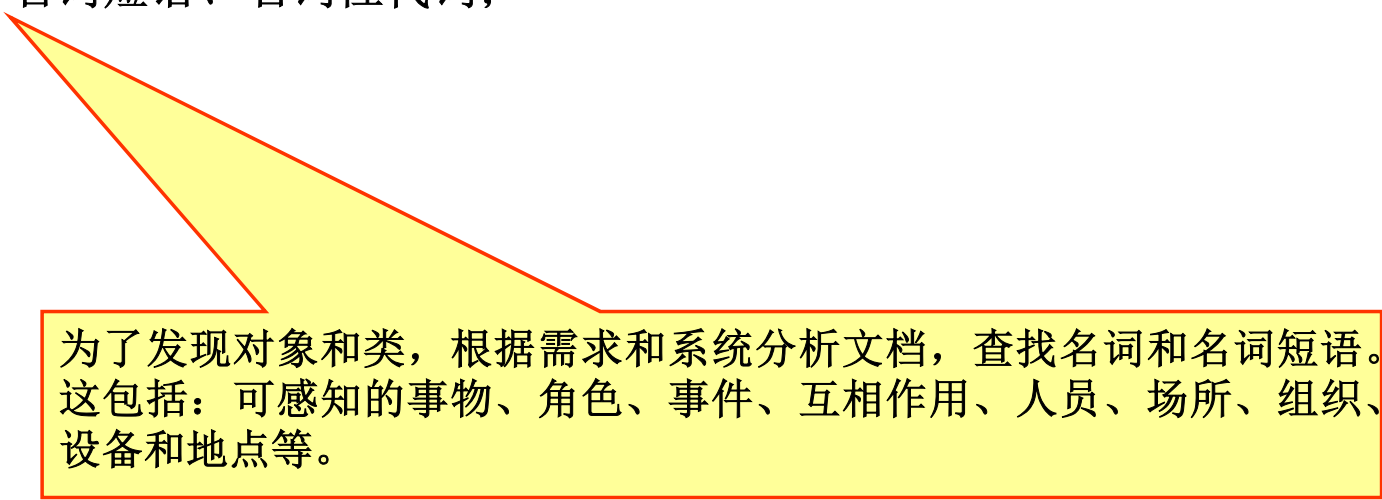
}

建立静态模型

1. 类的识别——名词识别法

识别问题域中的实体，实体的描述通常用名词、名词短语、名词性代词的形式出现。

- 用指定语言对系统进行描述；
- 从系统描述中标识名词、名词短语、名词性代词；
- 识别确定（取、舍）类。



为了发现对象和类，根据需求和系统分析文档，查找名词和名词短语。这包括：可感知的事物、角色、事件、互相作用、人员、场所、组织、设备和地点等。

建立静态模型

类识别举例——识别类

有一个购物超市，顾客可在货架上自由挑选商品，由收款机收款，收款机读取商品上的条形码标签，并计算商品价格。收款机全天开机，每台收款机有多个收款员，收款机记录销售事件，每个收款员交接班要记录帐册，记录前班节余，上交款，和本班节余等信息。收款机根据条形码进行收款，系统按商品一览表，对商品架上数量进行修改，商品中分特价商品和计量商品两种，供货员保证架上商品在下线以上，上线以下。每天收款机与上级系统相连，计帐。完成发送消息，查帐，报帐，价格更新，商品种类增删等操作。保留所有交易的记录，以备账务复查、清理货存及汇总使用。

建立静态模型

类识别举例——识别类

有一个购物超市，顾客可在货架上自由挑选商品，由收款机收款，收款机读取商品上的条形码标签，并计算商品价格。收款机全天开机，每台收款机有多个收款员，收款机记录销售事件，每个收款员交接班要记录帐册，记录前班节余，上交款，和本班节余等信息。收款机根据条形码进行收款，系统按商品一览表，对商品架上数量进行修改，商品中分特价商品和计量商品两种，供货员保证架上商品在下线以上，上线以下。每天收款机与上级系统相连，计帐。完成发送消息，查帐，报帐，价格更新，商品种类增删等操作。保留所有交易的记录，以备账务复查、清理货存及汇总使用。

通过分析问题的陈述，用名词分析法确定以下类：

购物超市 顾客 货架 收款机 商品 条形码 价格 收款员 销售事件
帐册 节余 上级系统 商品一览表 特价商品 计量商品 供货员

针对初步识别的类，首先要去掉冗余类、不相干类、模糊类、独立性不强的类以及描述的操作的类。以此得出购物超市的基本类图（概念层）。

建立静态模型

1. 类的识别——名词识别法

识别问题域中的实体，实体的描述通常用名词、名词短语、名词性代词的形式出现。

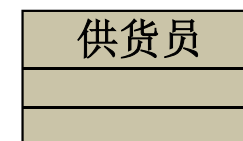
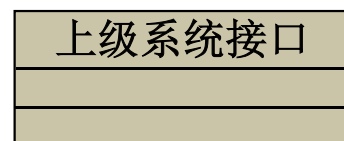
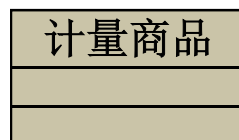
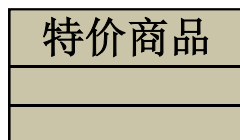
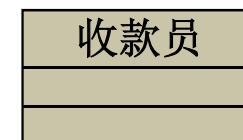
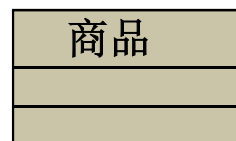
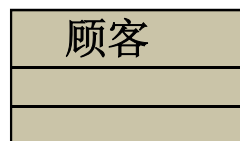
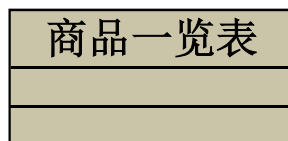
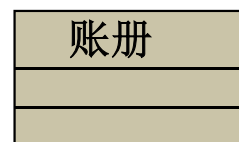
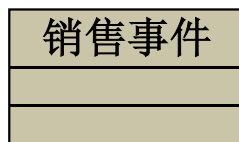
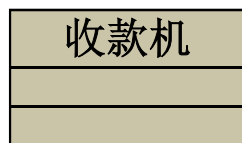
- 用指定语言对系统进行描述；
- 从系统描述中标识名词、名词短语、名词性代词；
- 识别确定（取、舍）类。

根据下述原则进一步确定类：

- ① 去掉冗余类：如两个类表述同一信息，应保留最具有描述能力的类。
- ② 去掉不相干的类：删除与问题无关或关系不大的类。
- ③ 删除模糊的类或性质独立性不强的类：有些初始类边界定义不确切，或范围太广，应该删除。
- ④ 所描述的操作不适宜作为对象类，并被其自身所操纵，所描述的只是实现过程中的暂时的对象，应删去。

建立静态模型

超市销售管理系统(类图，概念层)



建立静态模型

2. 类属性与操作识别

- (1) **属性 (Attribute)** 属性用来描述类的特征，表示需要处理的数据。
- (2) **方法 (Method)** 对数据的具体处理过程的描述放在方法部分，方法说明了该类能做些什么工作, 完成什么任务。

建立静态模型

2.1 类识别举例——识别属性与方法

有一个购物超市，顾客可在货架上自由挑选商品，由收款机收款，收款机读取商品上的条形码标签，并计算商品价格。收款机全天开机，每台收款机有多个收款员，收款机记录销售事件，每个收款员交接班要记录帐册，记录前班节余，上交款，和本班节余等信息。收款机根据条形码进行收款，系统按商品一览表，对商品架上数量进行修改，商品中分特价商品和计量商品两种，供货员保证架上商品在下线以上，上线以下。每天收款机与上级系统相连，计帐。完成发送消息，查帐，报帐，价格更新，商品种类增、删等操作。保留所有交易的记录，以备账务复查、清理货存及汇总使用。

建立静态模型

2.1 类识别举例——识别属性与方法

有一个购物超市，顾客可在货架上自由挑选商品，由收款机收款，收款机读取商品上的条形码标签，并计算商品价格。收款机全天开机，每台收款机有多个收款员，收款机记录销售事件，每个收款员交接班要记录帐册，记录前班节余，上交款，和本班节余等信息。收款机根据条形码进行收款，系统按商品一览表，对商品架上数量进行修改，商品中分特价商品和计量商品两种，供货员保证架上商品在下线以上，上线以下。每天收款机与上级系统相连，计帐。完成发送消息，查帐，报帐，价格更新，商品种类增、删等操作。保留所有交易的记录，以备账务复查、清理货存及汇总使用。

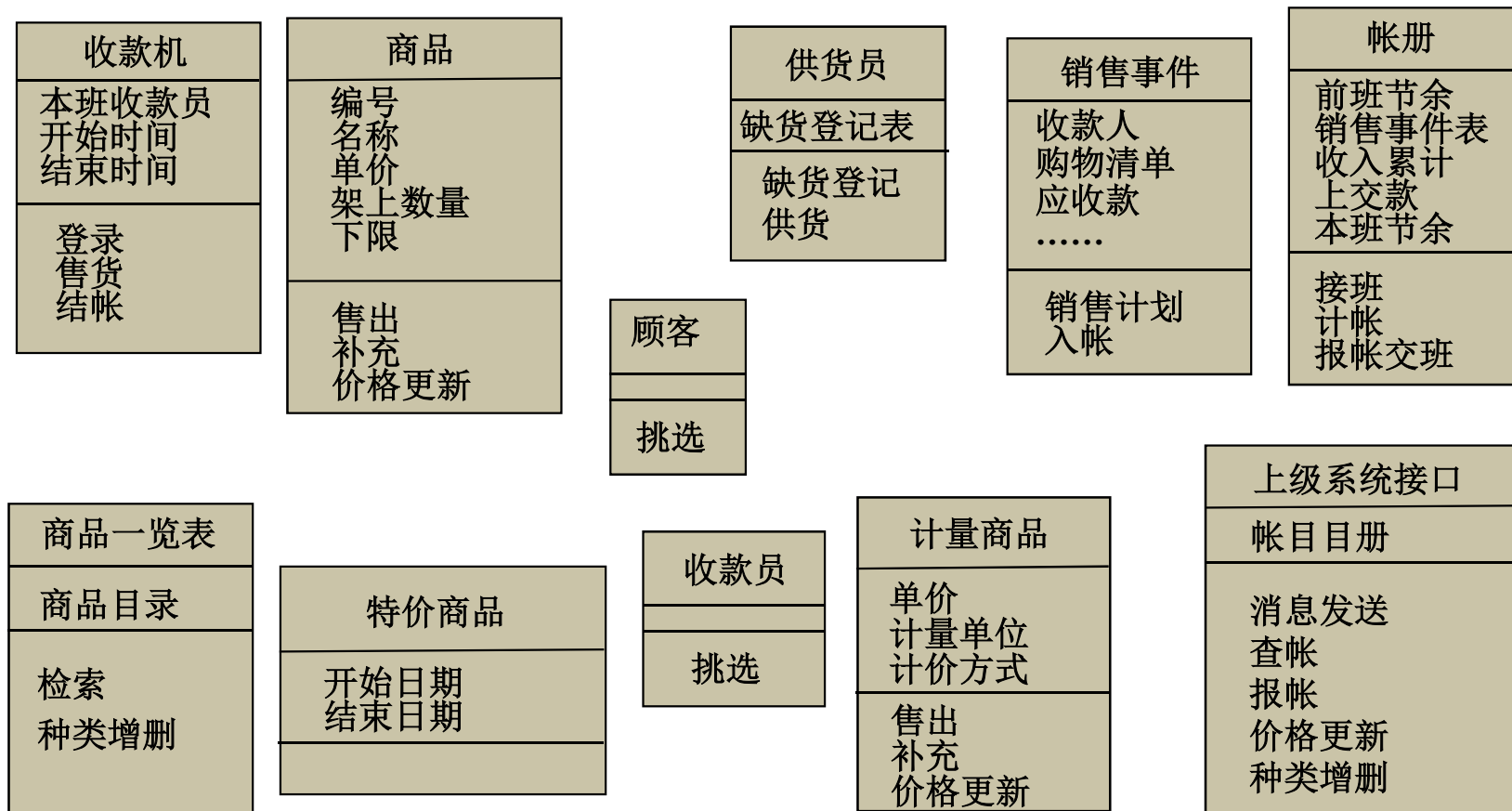
通过分析问题的陈述，用动词或动宾结构分析法确定以下方法：

挑选 收款 读取 计算 记录 交接班 收款 修改 记账 发送消息 查账
报账 更新 增 删 保留 复查 清理 汇总

注意：根据面向对象方法的要求，属性与方法要归纳到识别出的类中，不应独立于类之外。

建立静态模型

2.2 超市销售管理系统(类图, 设计层)



建立静态模型

3. 类图的抽象层次和细化关系

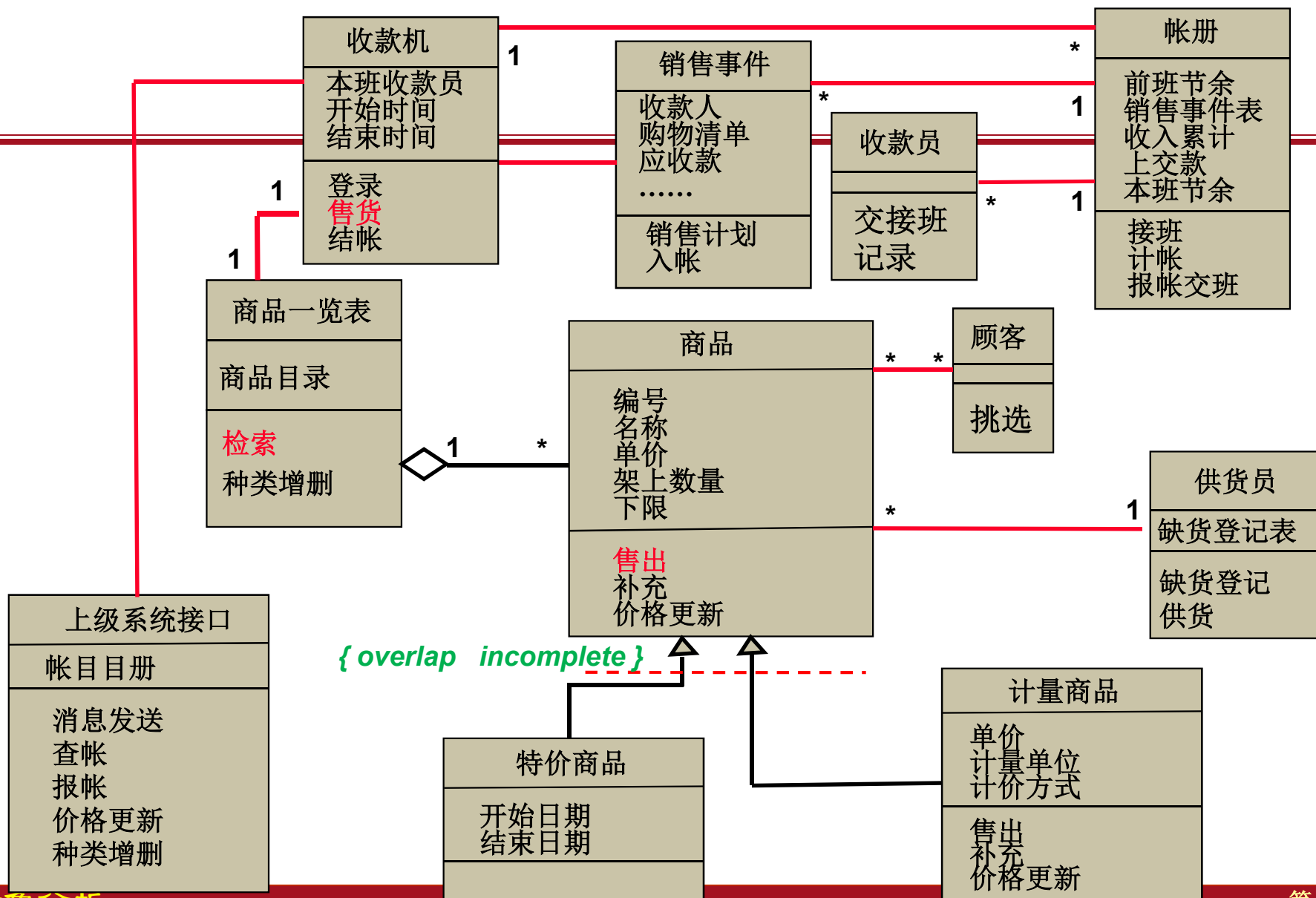
在软件开发的阶段都使用类图, 但这些类图表示了不同层次的抽象。

- 在需求分析阶段, 类图是研究领域的概念;
- 在设计阶段, 类图描述类与类之间的接口;
- 而在实现阶段, 类图描述软件系统中类的实现。

类图分为三个层次:

1. 概念层(**Conceptual**)类图描述应用领域中的概念。
2. 说明层(**Specification**)类图描述软件的接口部分, 而不是软件的实现部分。
3. 只有在实现层(**Implementation**)才真正有类的概念, 并且揭示软件的实现部分。

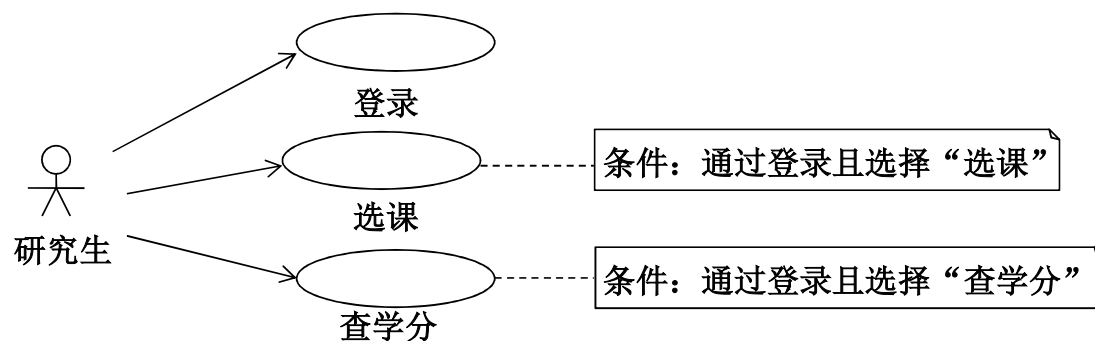
关系层完整的类图



建立静态模型

4. 边界类

一般情况下，交互接口、界面并不显然地出现在用户定义、需求描述中，而用例图中的每个交互用例都需体现用户与系统信息的交换。因此，需要考虑是否设计边界类与之对应。边界类不仅与系统内部用例有关，还需要考虑系统的外部环境。

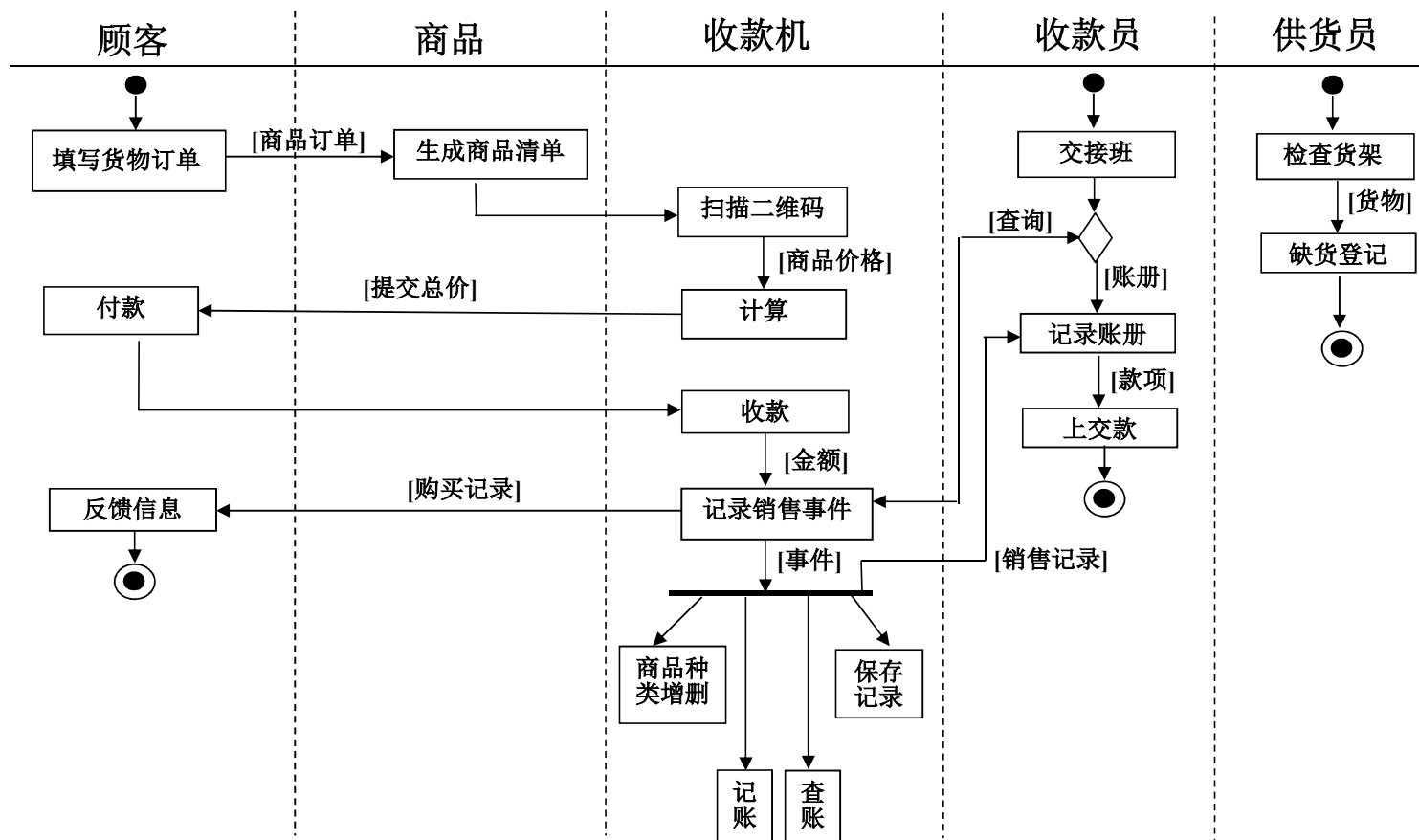


建立动态模型

- 通过对对象的组织 and 交互，有助于进一步发现对象活动，更能增进类间关系的发现和确认。
- 补充用例模型、静态模型中，关于操作、顺序、流程等与过程相关的建模。
- 对于大型复杂系统来说，通过对对象间的动态分析，更有助于明确类的方法的分组、类间关系的完善。

建立动态模型

活动图——超市销售管理系统



建立动态模型

活动图——俄罗斯方块状态图对应的活动图描述

