

第10章 面向对象分析

10.1 面向对象分析的基本过程

10.2 需求陈述

10.3 建立对象模型

10.4 建立动态模型

10.5 建立功能模型

10.6 定义服务

10.1 面向对象分析的基本过程

10.1.1 概述

面向对象分析过程：

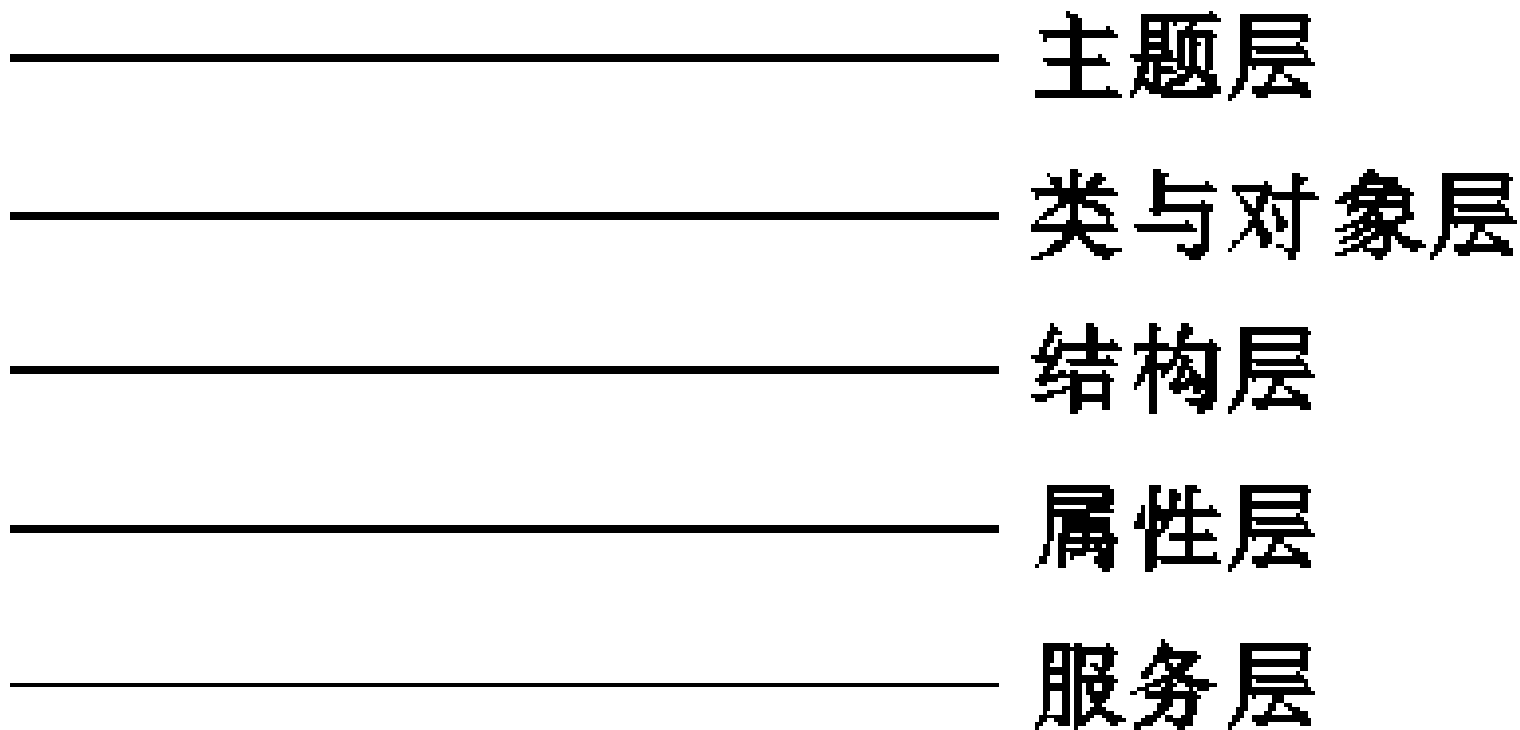
- 分析陈述用户需求的文件。
- 发现和改正原始陈述中的二义性和不一致性，补充遗漏的内容。
- 深入理解用户需求，抽象出目标系统的本质属性，并用模型准确地表示出来。

10.1.2 3个子模型与5个层次

3个子模型:

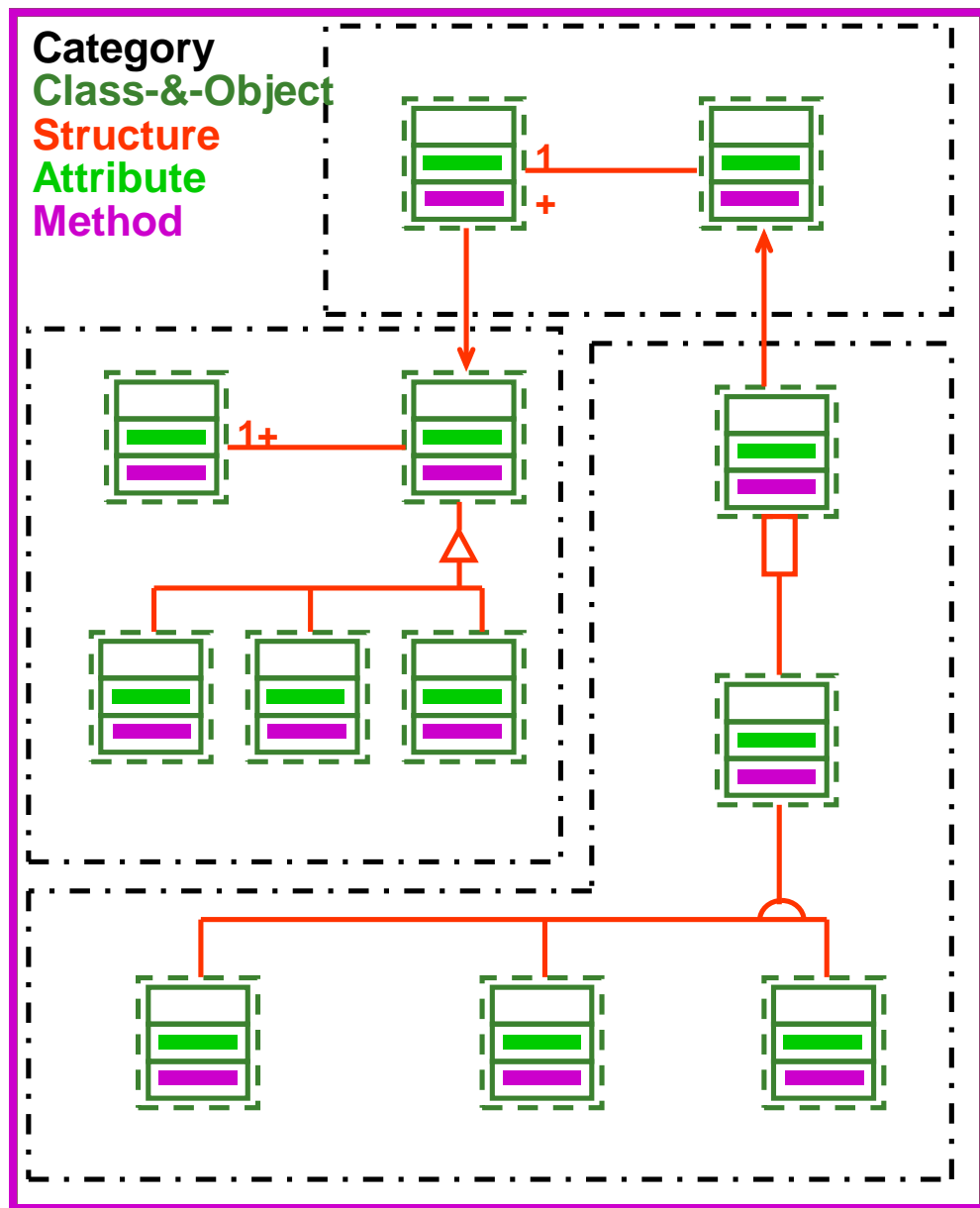
- 对象模型——抽象出客观世界实体及实体间相互关系;
- 动态模型——涉及交互作用和时序问题;
- 功能模型——解决运算量很大的问题。

5个层次:



5个层次对应的 5项主要活动:

- 找出类与对象
- 识别结构
- 识别主题
- 定义属性
- 定义服务



面向对象分析大体顺序：

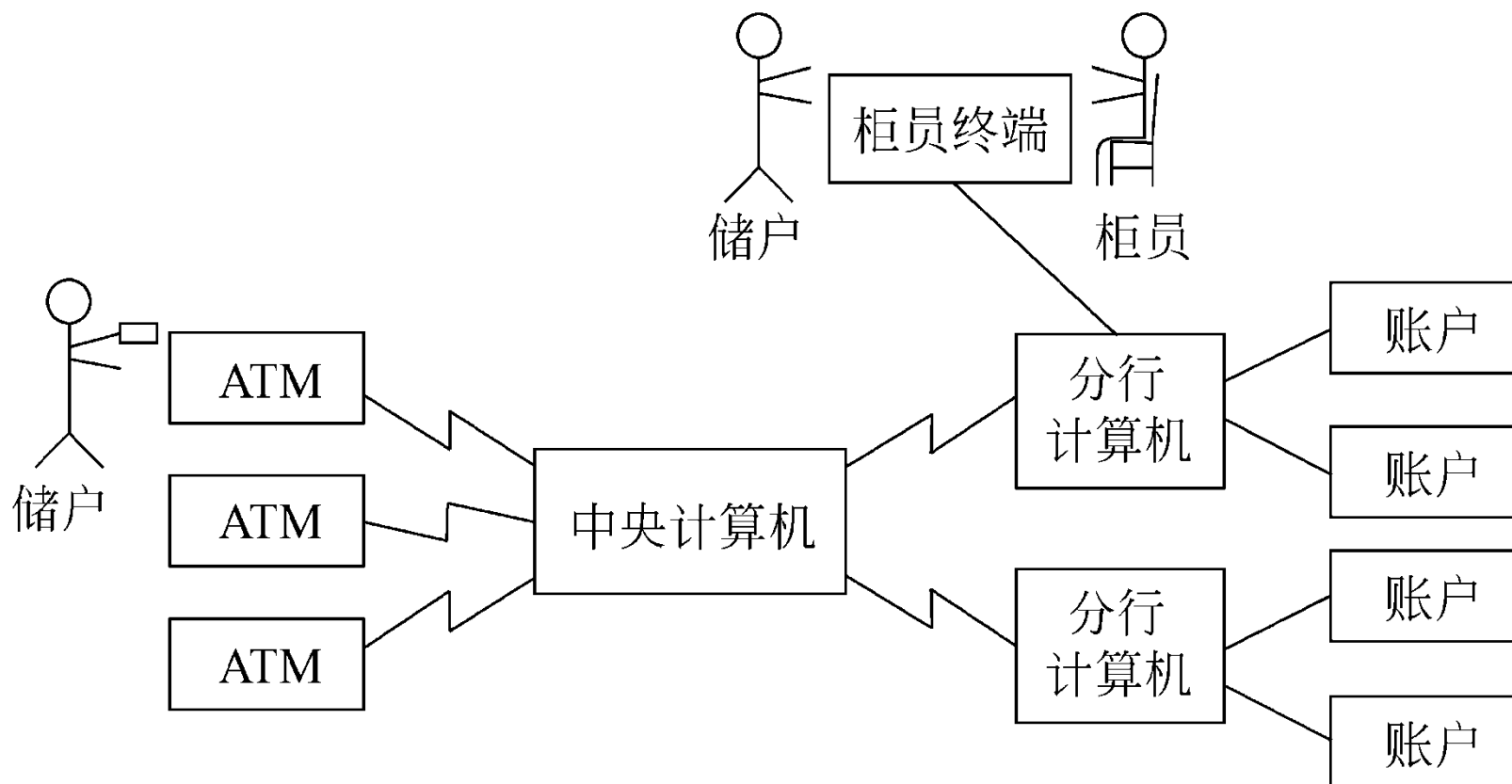
- 寻找类与对象
- 识别结构
- 识别主题
- 定义属性
- 建立动态模型
- 建立功能模型
- 定义服务

10.2 需求陈述

10.2.1 书写要点

- 需求陈述的内容包括：问题范围，功能需求，性能需求，应用环境及假设条件等。
- 书写需求陈述时，要尽力做到语法正确，而且应该慎重选用名词、动词、形容词和同义词。
- 需求陈述仅仅是理解用户需求的出发点，它并不是一成不变的文档。
- 系统分析员必须与用户及领域专家密切配合协同工作，共同提炼和整理用户需求。

10.2.2 例子



需求陈述，见P234

10.3 建立对象模型

典型的工作步骤：

- 首先确定对象类和关联，对于大型复杂问题还要进一步划分出若干个主题；
- 然后给类和关联增添属性，以进一步描述；
- 利用适当的继承关系进一步合并和组织类；
- 对类中操作的最后确定，则需等到建立了动态模型和功能模型之后。

10.3.1 确定类与对象

1. 找出候选的类与对象

- 对象是对问题域中有意义的事物的抽象，它们既可能是物理实体，也可能是抽象概念。
- 大多数客观事物可分为下述5类：
 - 可感知的物理实体
 - 人或组织的角色
 - 应该记忆的事件
 - 两个或多个对象的相互作用
 - 需要说明的概念

非正式分析：

- 用自然语言书写的需求陈述为依据，
 - 把名词作为类与对象的候选者；
 - 把形容词作为确定属性的线索；
 - 把动词作为服务(操作)的候选者。
- 用这种简单方法确定的候选者是非常不准确的，其中往往包含大量不正确的或不必要的事物，还必须经过更进一步的严格筛选。

ATM系统类与对象的初步的候选者:

银行	软件	支票	副本
自动取款机	成本	账户	信息
系统	市	事务	密码
中央计算机	街道	现金兑换卡	类型
分行计算机	营业厅	余额	取款额
柜员终端	储蓄所	磁卡	账单
网络	柜员	分行代码	访问
总行	储户	卡号	通信链路(隐含)
分行	现金	用户	事务日志(隐含)

2. 筛选出正确的类与对象

筛选标准:

- 冗余 两个类表达了同样的信息
- 无关 删掉与当前要解决的问题无关的
- 笼统 笼统的、泛指的名词
- 属性 实际上描述的是其他对象的属性
- 操作 既可作为名词，又可作为动词的词
- 实现 去掉仅和实现有关的候选的类与对象

第2步：去粗取精

例：从ATM需求分析中提出的名词集合

~~银行~~、ATM、~~系统~~、中央计算机、分行计算机、柜员终端、~~网络~~、总行、分行、~~软件~~、~~成本~~、~~市~~、~~街道~~、~~营业厅~~、~~储蓄所~~、柜员、储户、~~现金~~、支票、~~账户~~、事务、现金兑换卡、~~余额~~、~~磁卡~~、~~分行代码~~、~~卡号~~、~~用户~~、~~副本~~、~~信息~~、~~密码~~、~~类型~~、~~取款额~~、~~账单~~、~~访问~~、通信链路、事务日志

筛选时依下列标准删除：

✱ **冗余** ✱ **无关** ✱ **笼统** ✱ **属性** ✱ **操作** ✱ **实现**

10.3.2 确定关联

1. 初步确定关联

- 在需求陈述中使用的描述性动词或动词词组，通常表示关联关系。
- 通过分析需求陈述，还能发现一些在陈述中隐含的关联。
- 最后，分析员还应该与用户及领域专家讨论问题域实体间的相互依赖、相互作用关系，根据领域知识再进一步补充一些关联。

2、确定关联 ⇒ 结构层

第1步：收集 ——

① 需求陈述中涉及objects的动词短语：

- ATM、中央计算机、分行计算机及柜员终端组成网络
- 总行拥有多台ATM
- ATM设在主要街道上
- 分行提供分行计算机和柜员终端
- 柜员终端设在分行营业厅及储蓄所内
- 分行分摊软件开发成本
- 储户拥有账户
- 分行计算机处理针对账户的事务
- 分行计算机维护账户
- 柜员终端与分行计算机通信
- 柜员输入针对账户的事务
- ATM与中央计算机交换关于事务的信息
- 中央计算机确定事务与分行的对应关系
- ATM读现金兑换卡
- ATM与用户交互
- ATM吐出现金
- ATM打印账单
- 系统处理并发的访问

② 需求陈述中隐含的关联

- 总行由各个分行组成
- 分行保管账户
- 总行拥有中央计算机
- 系统维护事务日志
- 系统提供必要的安全性
- 储户拥有现金兑换卡

③ 根据问题域知识得出的关联

- 现金兑换卡访问账户
- 分行雇用柜员

2. 筛选

- 已删去的类之间的关联。
- 与问题无关的或应在实现阶段考虑的关联。
- 瞬时事件 关联应该描述问题域的静态结构，而不应该是一个瞬时事件。
- 三元关联 三个或三个以上对象之间的关联，大多可以分解为二元关联或用词组描述成限定的关联。
- 派生关联 应该去掉那些可以用其他关联定义的冗余关联。

第2步：筛选删除

① 与已删去的object 有关的关联

- ~~ATM、中央计算机、分行计算机及柜员终端组成网络~~
- ~~总行拥有多台ATM~~
- ~~ATM设在主要街道上~~
- 分行提供分行计算机和柜员终端
- ~~柜员终端设在分行营业厅及储蓄所内~~
- ~~分行分摊软件开发成本~~
- 储户拥有账户
- 总行由各个分行组成
- 现金兑换卡访问账户
- 分行保管账户
- 总行拥有中央计算机
- 分行雇用柜员

- 分行计算机处理针对账户的事务
- 分行计算机维护账户
- 柜员终端与分行计算机通信
- 柜员输入针对账户的事务
- ATM与中央计算机交换关于事务的信息
- 中央计算机确定事务与分行的对应关系
- ATM读现金兑换卡
- ATM与用户交互
- ~~ATM吐出现金~~
- ~~ATM打印账单~~
- ~~系统处理并发的访问~~
- ~~系统维护事务日志~~
- ~~系统提供必要的安全性~~
- 储户拥有现金兑换卡

② 与问题无关的或应在实现阶段考虑的关联

③瞬时事件：注意关联描述静态结构

- ~~ATM、中央计算机、分行计算机及柜员终端组成网络~~
- 总行拥有多台ATM
- ~~ATM设在主要街道上~~
- 分行提供分行计算机和柜员终端
- ~~柜员终端设在分行营业厅及储蓄所内~~
- ~~分行分摊软件开发成本~~
- 储户拥有账户
- 总行由各个分行组成
- 现金兑换卡访问账户
- 分行保管账户
- 总行拥有中央计算机
- **分行雇用柜员**
- 分行计算机处理针对账户的事务
- 分行计算机维护账户
- 柜员终端与分行计算机通信
- 柜员输入针对账户的事务
- ATM与中央计算机交换关于事务的信息
- 中央计算机确定事务与分行的对应关系
- ~~ATM读现金兑换卡~~
- ~~ATM与用户交互~~
- ~~ATM吐出现金~~
- ~~ATM打印账单~~
- ~~系统处理并发的访问~~
- ~~系统维护事务日志~~
- ~~系统提供必要的安全性~~
- 储户拥有现金兑换卡

④三元关联：分解为二元关联或限定关联

- 柜员输入针对账户的事务
= 柜员输入事务 + 事务修改账户
- 分行计算机处理针对账户的事务
= 分行计算机处理事务 + 事务修改账户
- ATM与中央计算机交换关于事务的信息
= ATM与中央计算机通信 + 在ATM上输入事务

注：识别链属性 —— 某实体仅用于描述另两个实体的关系，且该实体本身不包含属性。例如“公司付给员工工资”可分解为“公司雇用员工”，附链属性“工资”。

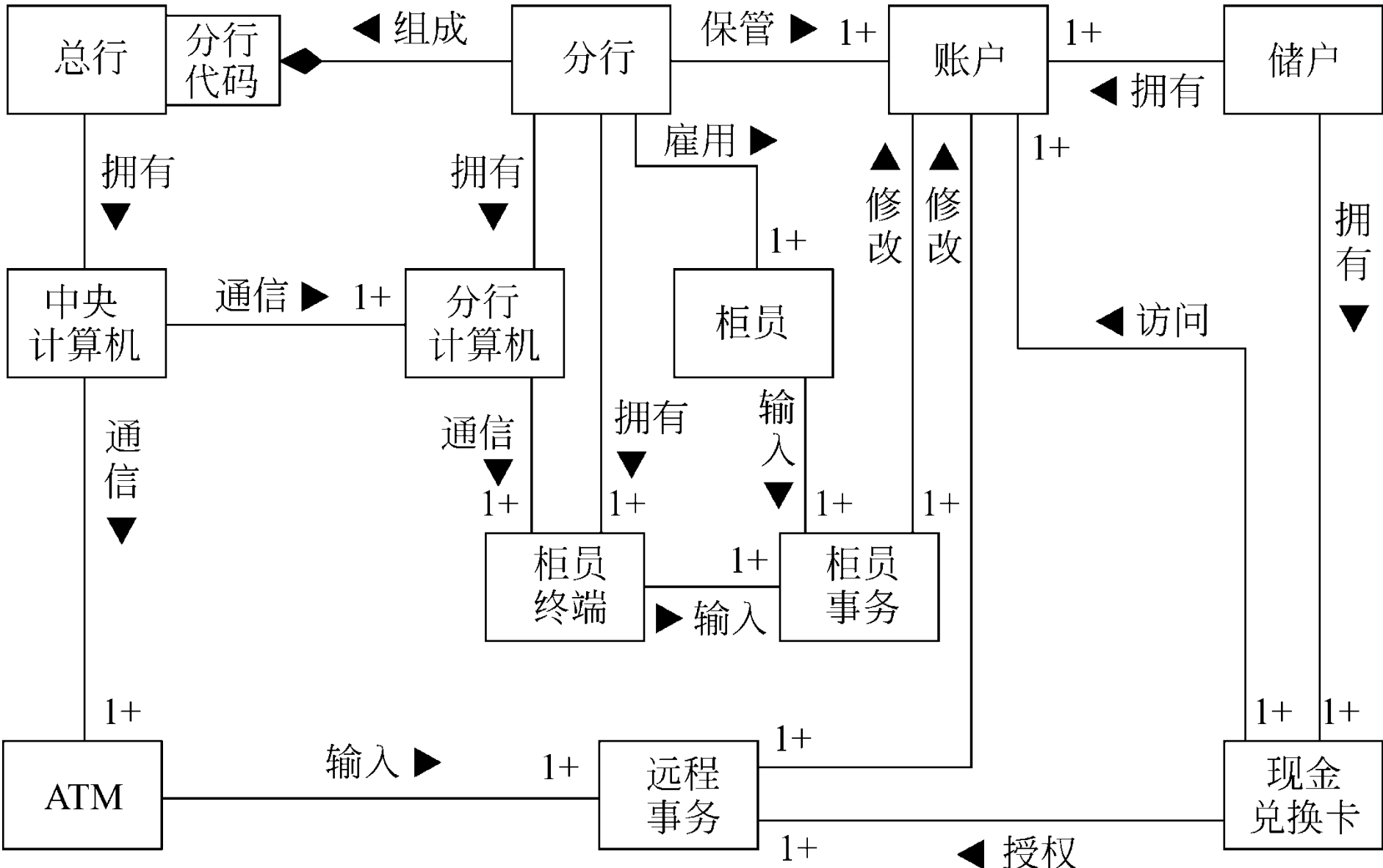
⑤派生关联：即可用其它关联定义的冗余关联

- ~~总行拥有多台ATM~~
- 分行提供分行计算机和柜员终端
- 储户拥有账户
- 总行由各个分行组成
- ~~分行保管账户~~
- ~~总行拥有中央计算机~~
- 现金兑换卡访问账户
- 储户拥有现金兑换卡
- 分行雇用柜员
- 分行计算机处理事务
- ~~分行计算机维护账户~~
- 柜员终端与分行计算机通信
- 柜员输入事务
- ~~事务修改账户~~
- ~~ATM与中央计算机通信~~
- 在ATM上输入事务
- 中央计算机确定事务与分行的对应关系

3. 进一步完善

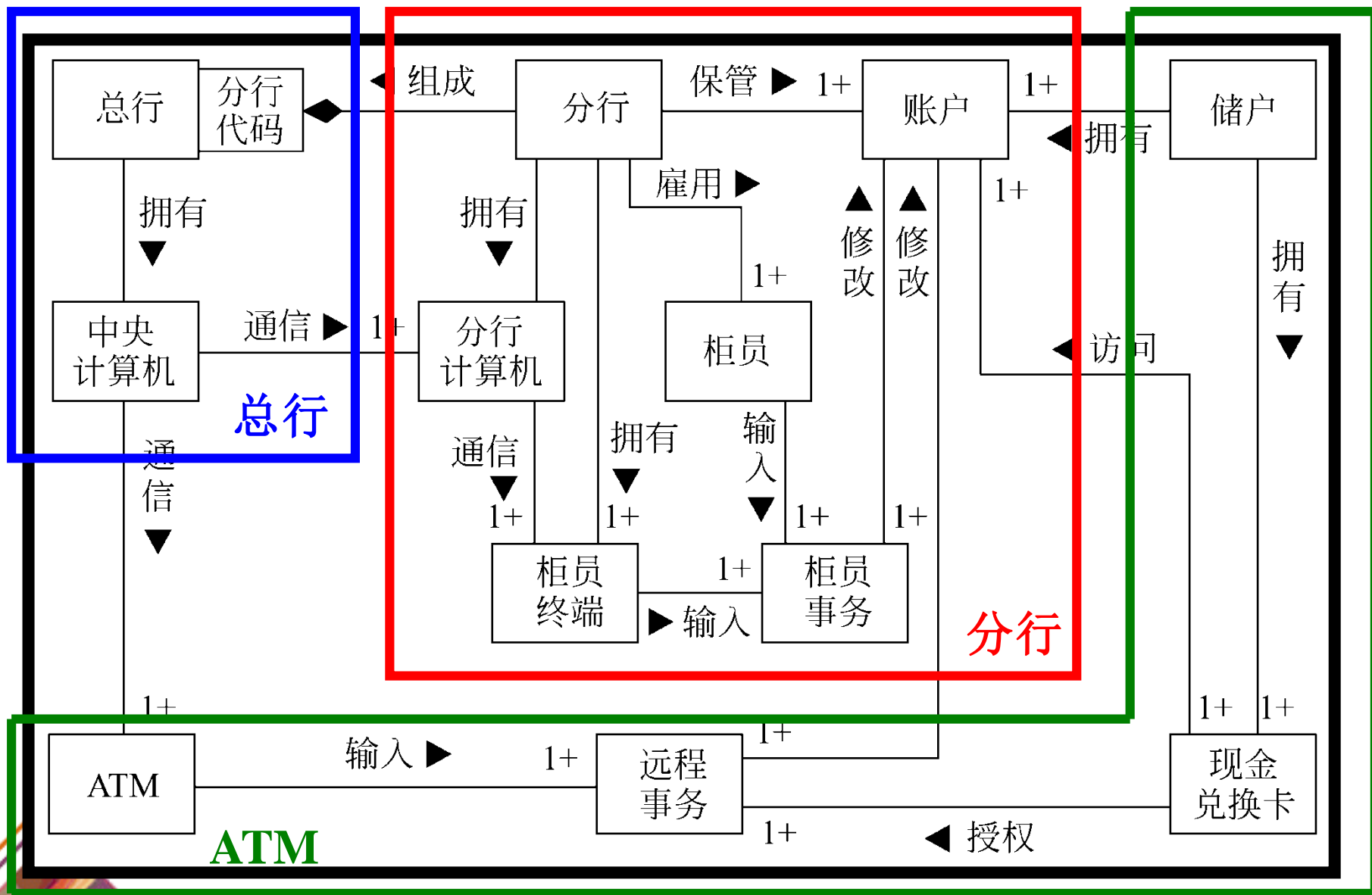
- **正名** 好的名字是帮助读者理解的关键因素之一。因此，应该仔细选择含义更明确的名字作为关联名。
- **分解** 为了能够适用于不同的关联，必要时应该分解以前确定的类与对象。
- **补充** 发现了遗漏的关联就应该及时补上。
- **标明重数** 应该初步判定各个关联的类型，并粗略地确定关联的重数。

- 分行提供分行计算机和柜员终端。
- 储户拥有账户。
- 分行计算机处理**事务**。事务修改账户。
- 柜员终端与分行计算机通信。
- **柜员输入事务**。事务修改账户。
- **ATM**与中央计算机通信。在**ATM**上输入**事务**。
- 中央计算机与分行通信。
- 总行由各个分行组成。
- 分行保管账户。
- 总行拥有中央计算机。
- 储户拥有现金兑换卡。
- 现金兑换卡访问账户。
- 分行雇用柜员。



10.3.3 划分主题

- 开发大型、复杂系统的过程中，为了降低复杂程度，人们习惯于把系统进一步划分成几个不同的主题。
 - 在开发很小的系统时，可能根本无须引入主题层；
 - 对于含有较多对象的系统，则往往先识别出类与对象和关联，然后划分主题；
 - 对于规模极大的系统，则首先由高级分析员粗略地识别对象和关联，然后初步划分主题，对系统结构有更深入的了解之后，再进一步修改和精炼主题。
- 应该按问题领域而不是用功能分解方法来确定主题。此外，应该按照使不同主题内的对象相互间依赖和交互最少的原则来确定主题。



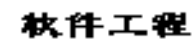
10.3.4 确定属性

1. 分析

- 通常，在需求陈述中用**名词词组**表示属性，用**形容词**表示可枚举的具体属性。
- 分析员还必须借助于领域知识和常识才能分析得出需要的属性。
- 应该仅考虑与具体应用直接相关的属性，不要考虑那些超出所要解决的问题范围的属性。

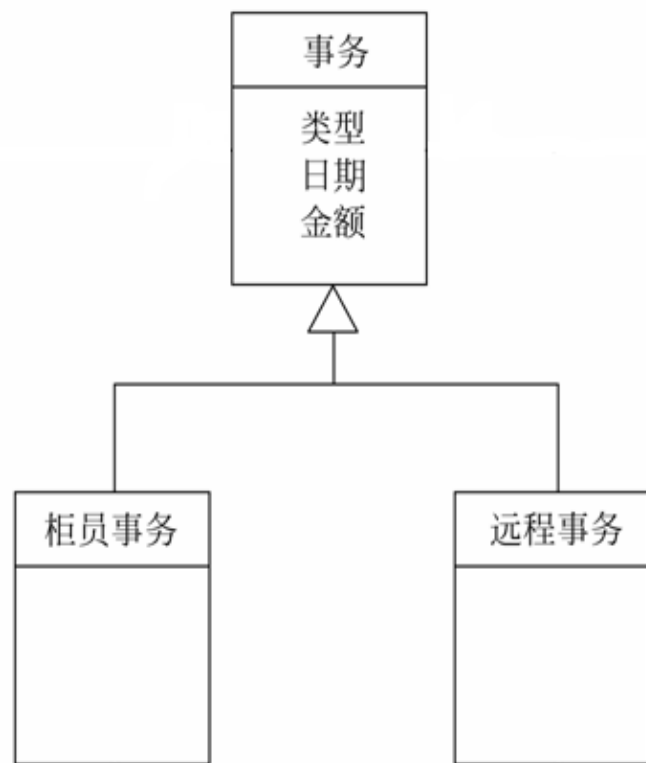
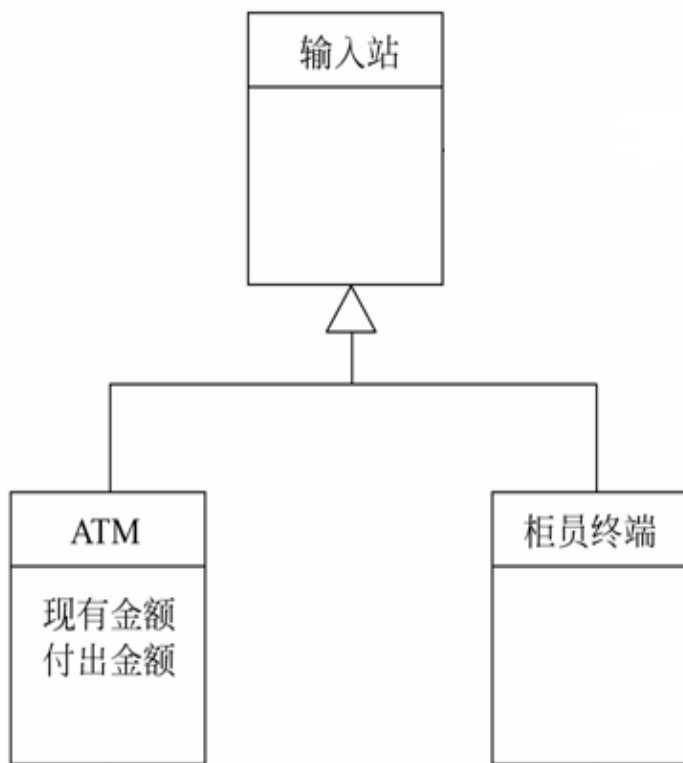
2. 选择

- 误把对象当作属性
- 误把关联类的属性当作一般对象的属性
- 把限定误当成属性
- 误把内部状态当成了属性
- 过于细化
- 存在不一致的属性



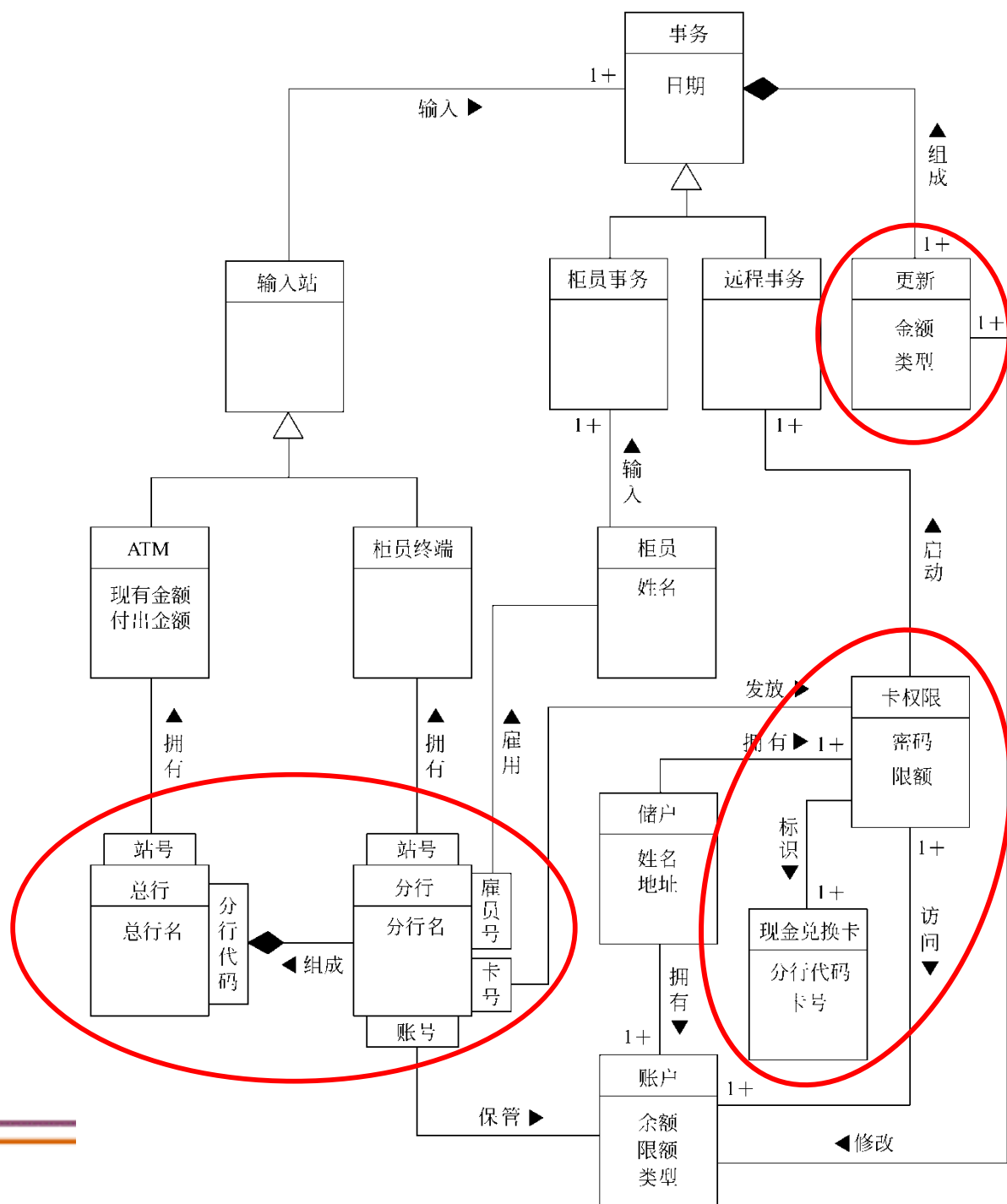
10.3.5 识别继承关系

- 确定了类中应该定义的属性之后，就可以利用继承机制共享公共性质，并对系统中众多的类加以组织。
- 两种方式建立继承关系：
 - 自底向上：抽象出现有类的共同性质泛化出父类，这个过程实质上模拟了人类归纳思维过程。
 - 自顶向下：把现有类细化成更具体的子类，这模拟了人类的演绎思维过程。



10.3.6 反复修改

- 仅仅经过一次建模过程很难得到完全正确的对象模型。
- 软件开发过程就是一个多次反复修改、逐步完善的过程。
- 面向对象的概念和符号在整个开发过程中都是一致的，更易实现反复修改、逐步完善。
- ATM系统可能做的修改：
 - 分解“现金兑换卡”类
 - “事务”由“更新”组成
 - 把“分行”与“分行计算机”合并



10.4 建立动态模型

- 第一步，是编写典型交互行为的脚本。
- 第二步，从脚本中提取出事件，确定触发每个事件的动作对象以及接受事件的目标对象。
- 第三步，排列事件发生的次序，确定每个对象可能有的状态及状态间的转换关系，并用状态图描绘它们。
- 最后，比较各个对象的状态图，检查它们之间的一致性，确保事件之间的匹配。

10.4.1 编写脚本

- 脚本是指系统在某一执行期间内出现的一系列事件。脚本描述用户与目标系统之间的一个或多个典型的交互过程。
- 编写脚本的目的，是保证不遗漏重要的交互步骤，它有助于确保整个交互过程的正确性的和清晰性。
- 编写脚本的过程，实质上就是分析用户对系统交互行为的要求的过程。
- 编写脚本时，首先编写正常情况的脚本，然后考虑特殊情况，最后考虑出错情况。

例：ATM系统的正常、异常情况脚本

ATM系统的正常情况脚本

- **ATM**请储户插卡；储户插入一张现金兑换卡。
- **ATM**接受该卡并读它上面的分行代码和卡号。
- **ATM**要求储户输入密码；储户输入自己的密码“1234”等数字。
- **ATM**请求总行验证卡号和密码；总行要求“39”号分行核对储户密码，然后通知**ATM**说这张卡有效。
- **ATM**要求储户选择事务类型(取款、转帐、查询等)；储户选择“取款”。
- **ATM**要求储户输入取款额；储户输入“880”。
- **ATM**确认取款额在预先规定的限额内，然后要求总行处理这个事务；总行把请求转给分行，该分行成功地处理完这项事务并返回该帐户的新余额。
- **ATM**吐出现金并请储户拿走这些现金；储户拿走现金。
- **ATM**问储户是否继续这项事务；储户回答“不”。
- **ATM**打印帐单，退出现金兑换卡，请储户拿走它们；储户取走帐单和卡。
- **ATM**请储户插卡

ATM系统的异常情况脚本

- **ATM**请储户插卡；储户插入一张现金兑换卡。
- **ATM**接受该卡并读它上面的分行代码和卡号。
- **ATM**要求储户输入密码；储户误输入“**8888**”。
- **ATM**请求总行验证卡号和密码；总行要求“**39**”号分行核对储户密码，然后通知**ATM**拒绝这张卡。
- **ATM**显示“密码错”，并请储户重新输入密码；储户输入“**1234**”；**ATM**请总行验证后知此次输入的密码正确。
- **ATM**要求储户选择事务类型(取款、转帐、查询等)；储户选择“取款”。
- **ATM**要求储户输入取款额；储户改变主意不想取款了，敲“取消”键。
- **ATM**退出现金兑换卡，请储户拿走它；储户取走他的卡。
- **ATM**请储户插卡

10.4.2 设想用户界面

- 大多数交互行为都可以分为应用逻辑和用户界面两部分。应用逻辑是内在的、本质的内容，用户界面是外在的表现形式。
- 用户界面的美观程度、方便程度、易学程度以及效率等等，是用户使用系统时最先感受到的，用户界面的好坏往往对用户是否喜欢、是否接受一个系统起很重要的作用。
- 软件开发人员往往快速地建立起用户界面的原型，供用户试用与评价。

向储户显示的信息

0

1

2

3

4

5

6

7

8

9

ENTER

CLEAR

CANCEL

账单出口

现金出口

10.4.3 画事件跟踪图

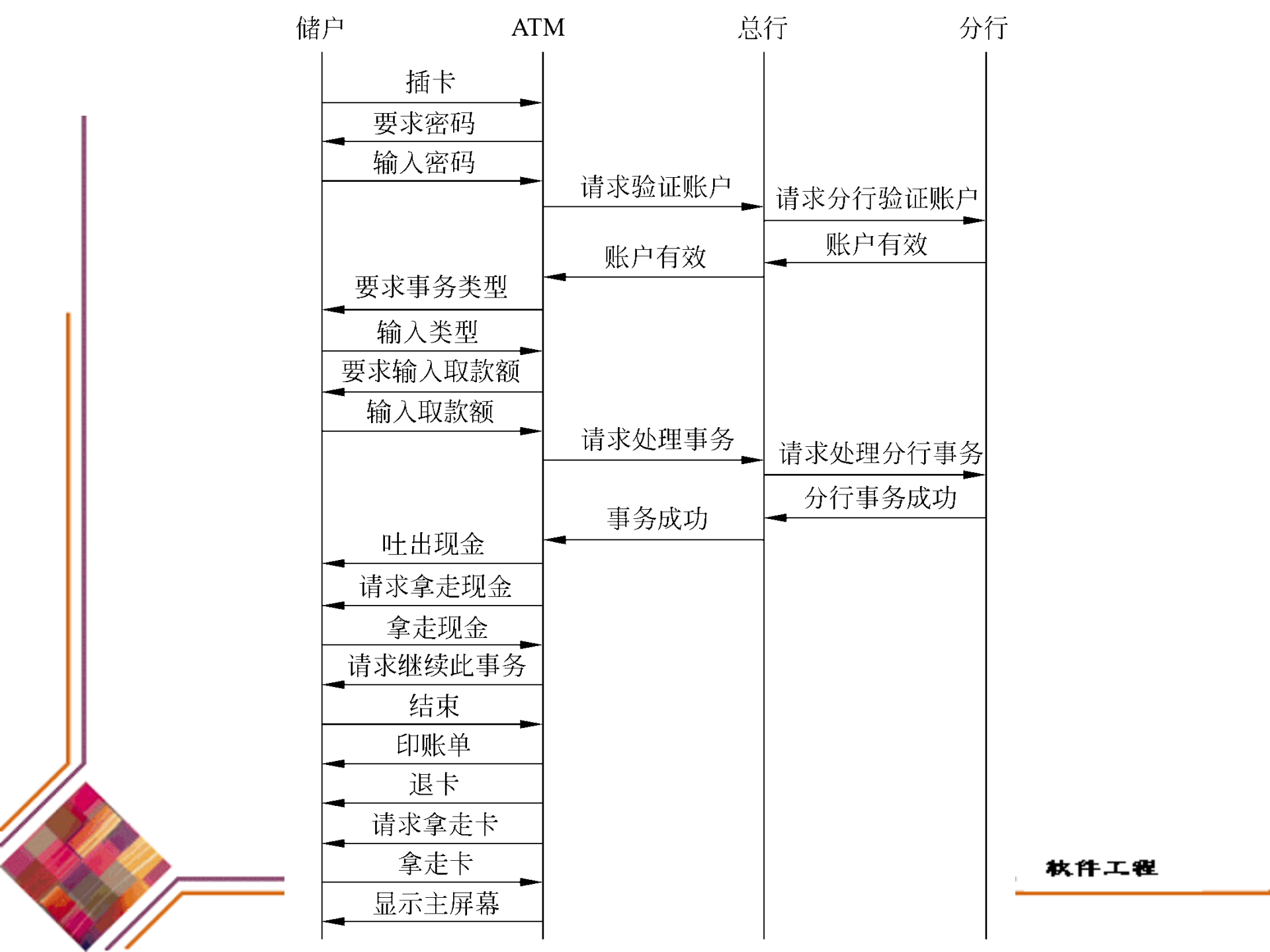
- 用自然语言书写的脚本往往不够简明，而且有时在阅读时会有二义性。
- 为了有助于建立动态模型，通常在画状态图之前先画出事件跟踪图。
- 为此首先需要进一步明确事件及事件与对象的关系。

1. 确定事件

- 应该仔细分析每个脚本，以便从中提取出所有外部事件。事件包括系统与用户交互的所有信号、输入、输出、中断、动作等等。
- 传递信息的对象的动作也是事件。大多数对象到对象的交互行为都对应着事件。
- 把对控制流产生相同效果的那些事件组合在一起作为一类事件，并给它们取一个惟一的名称。
- 把对控制流有不同影响的那些事件区分开来，不要误把它们组合在一起。
- 区分出每类事件的发送对象和接受对象。

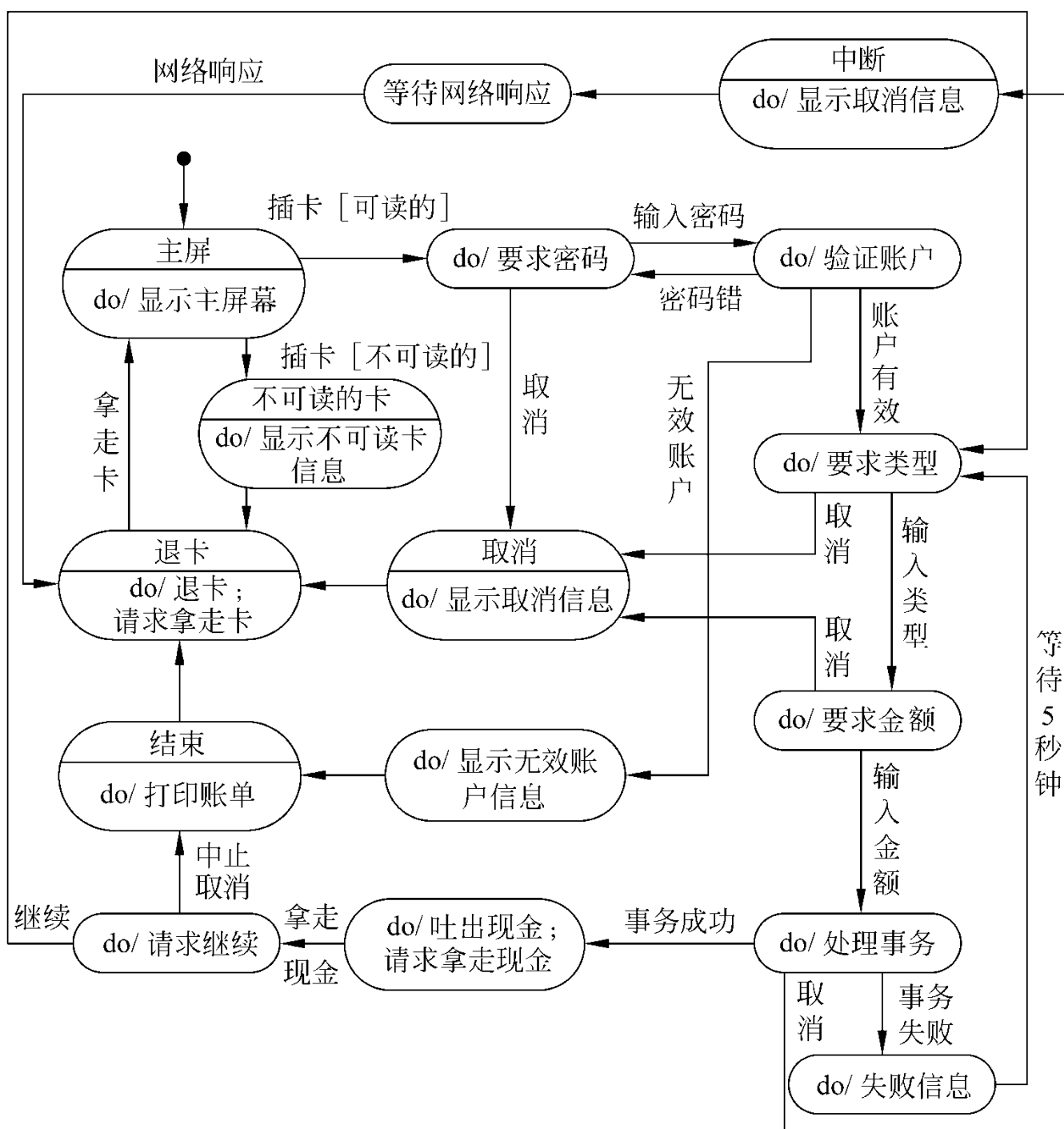
2. 画出事件跟踪图

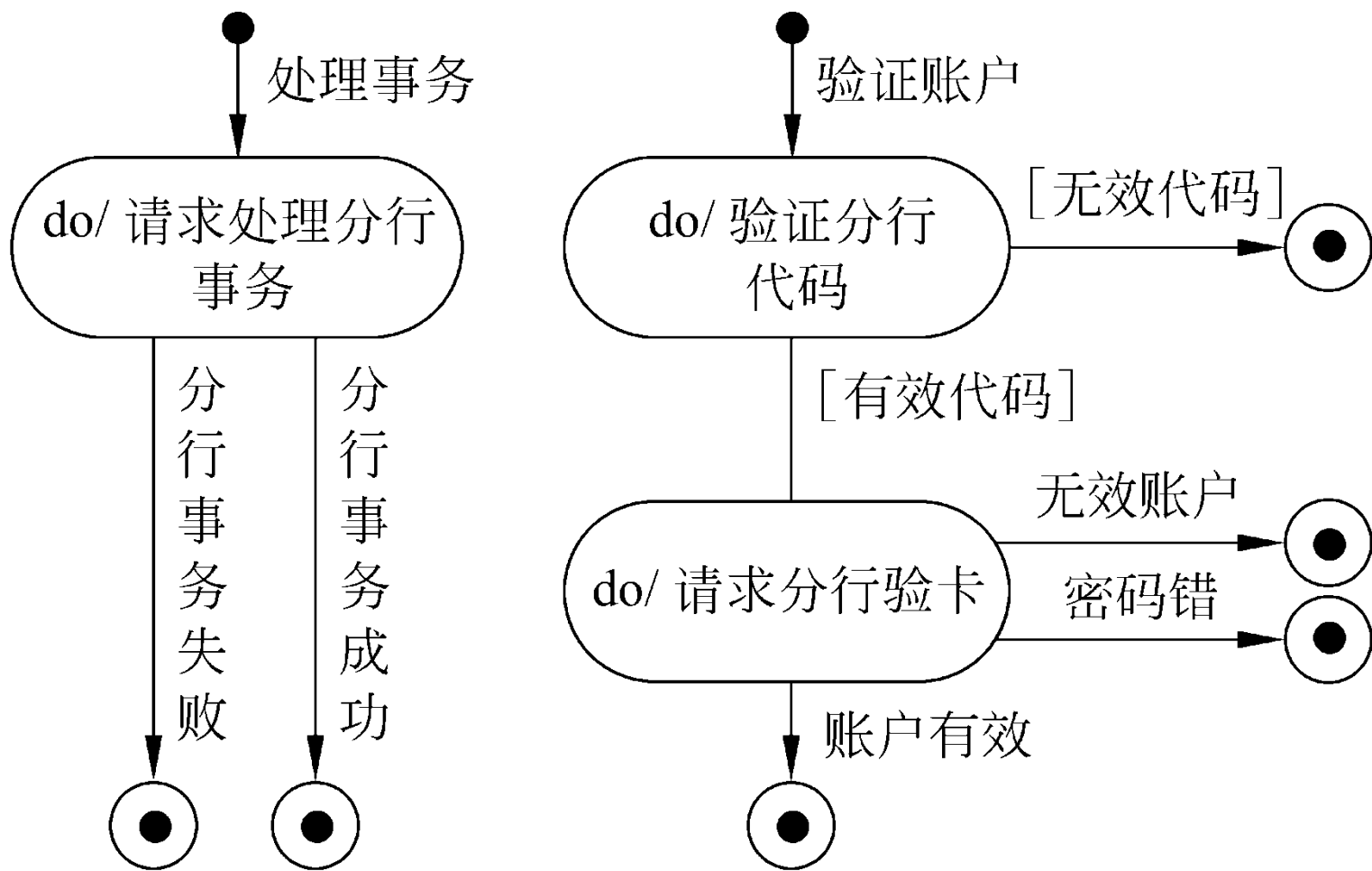
- 用事件跟踪图把事件序列以及事件与对象的关系，形象、清晰地表示出来。
- 一条竖线代表一个对象，每个事件用一条水平的箭头线表示，箭头方向从事件的发送对象指向接受对象。
- 时间从上向下递增。
- 箭头线之间的间距并没有具体含义。



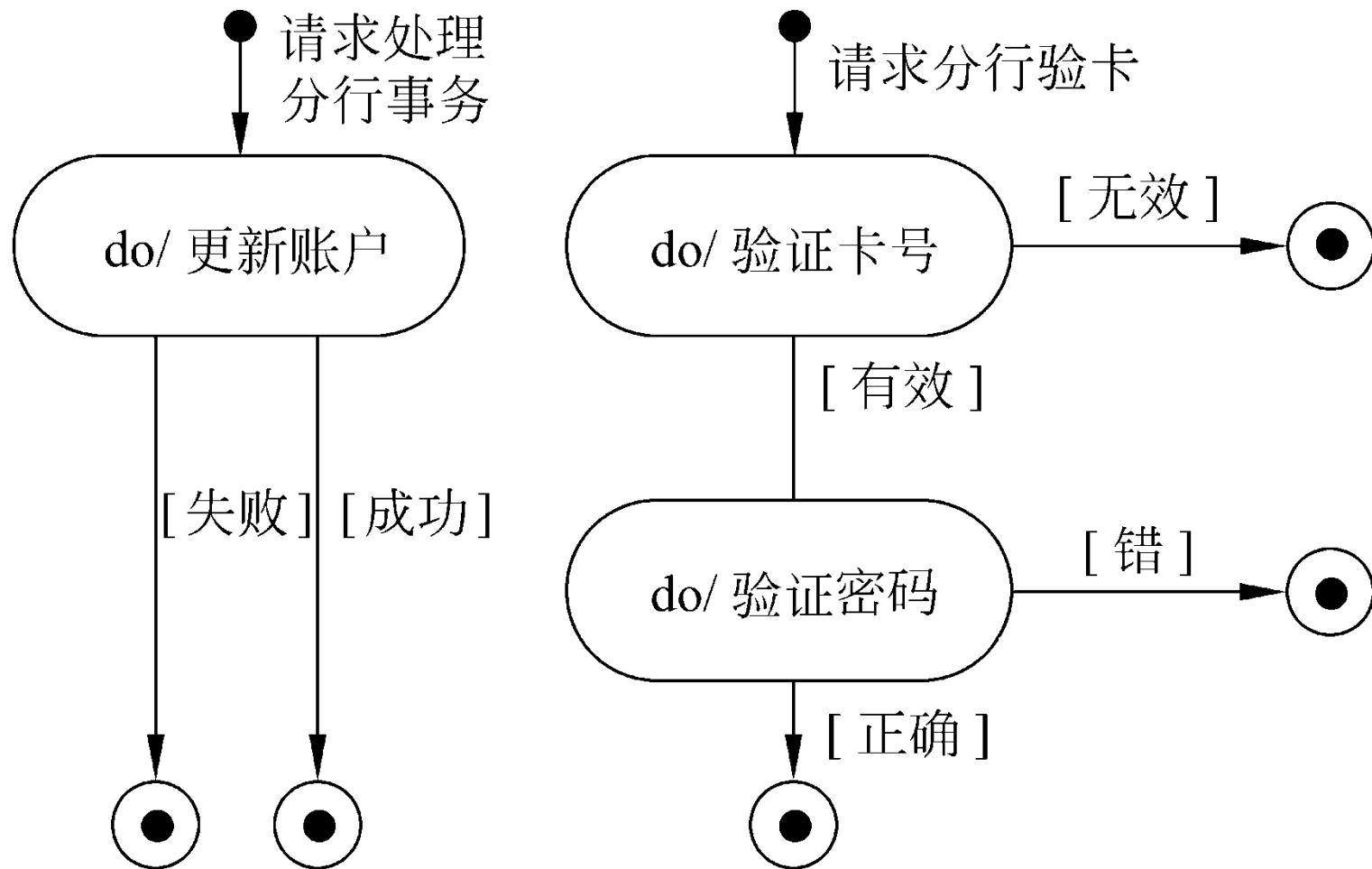
10.4.4 画状态图

- 状态图描绘事件与对象状态的关系。
- 通常，用一张状态图描绘一类对象的行为。
- 考虑完正常事件之后再考虑边界情况和特殊情况。
- 出错处理是不能省略的。





总行类的状态图



分行类的状态图

10.4.5 审查动态模型

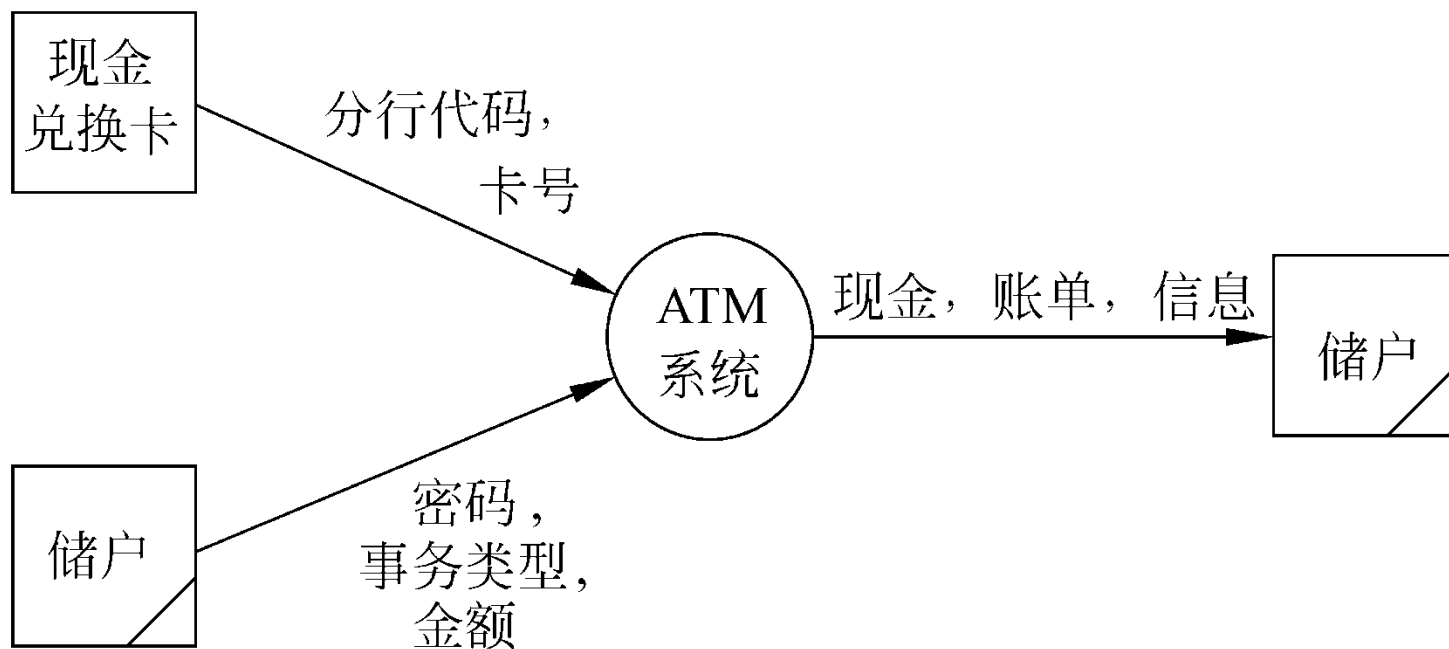
- 在完成了每个具有重要交互行为的类的状态图之后，应该检查系统级的完整性和一致性。
- 一般说来，每个事件都应该既有发送对象又有接受对象，当然，有时发送者和接受者是同一个对象。
- 应该审查每个事件，跟踪它对系统中各个对象所产生的效果，以保证它们与每个脚本都匹配。

10.5 建立功能模型

- 功能模型表明了系统中数据之间的依赖关系，以及有关的数据处理功能，它由一组数据流图组成。其中的处理功能可以用IPO图(或表)、伪码等多种方式进一步描述。
- 通常在建立了对象模型和动态模型之后再建立功能模型。

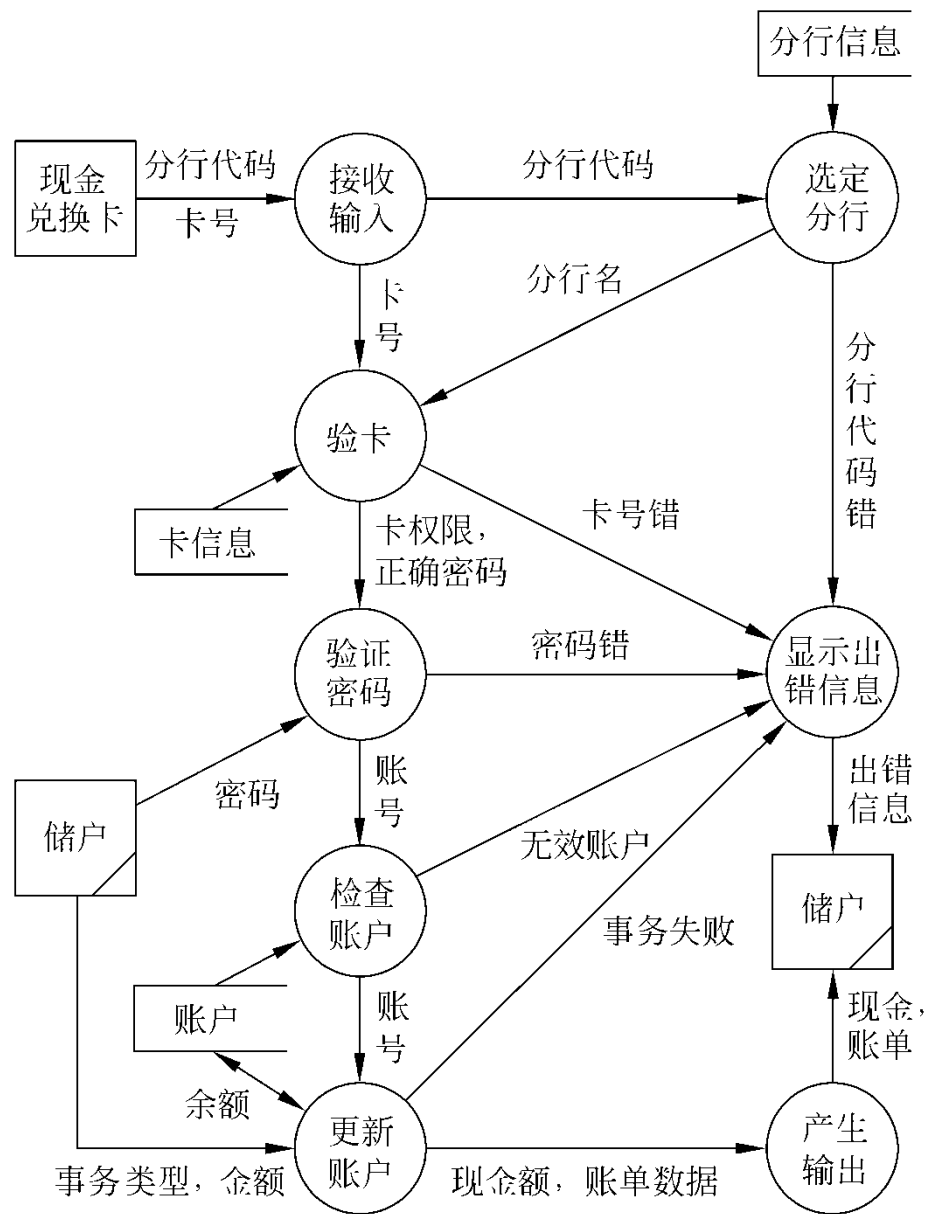
10.5.1 画出基本系统模型图

- 基本系统模型由若干个数据源点/终点，及一个处理框组成，这个处理框代表了系统加工、变换数据的整体功能。



10.5.2 画出功能级数据流图

- 把基本系统模型中单一的处理框分解成若干个处理框，以描述系统加工、变换数据的基本功能，就得到功能级数据流图。



10.5.3 描述处理框功能

- 把数据流图分解细化到一定程度之后，就应该描述图中各个处理框的功能。
- 描述既可以是说明性的，也可以是过程性的。
 - 说明性描述规定了输入值和输出值之间的关系，以及输出值应遵循的规律。
 - 过程性描述则通过算法说明“做什么”。

10.6 定义服务

- 建立了动态模型和功能模型之后，才能最终确定类中应有的服务，因为这两个子模型更明确地描述了每个类中应该提供哪些服务。
- 在确定类中应有的服务时，既要考虑该类实体的常规行为，又要考虑在本系统中特殊需要的服务。

1. 常规行为

- 在分析阶段可以认为，类中定义的每个属性都是可以访问的，也就是说，假设在每个类中都定义了读、写该类每个属性的操作。
- 通常无需在类图中显式表示这些常规操作。

2. 从事件导出的操作

- 状态图中发往对象的事件也就是该对象接收到的消息，因此该对象必须有由消息选择符指定的操作，这个操作修改对象状态(即属性值)并启动相应的服务。

3. 与数据流图中处理框对应的操作

- 数据流图中的每个处理框都与一个对象上的操作相对应。

4. 利用继承减少冗余操作

- 应该尽量利用继承机制以减少所需定义的服务数目。
- 尽量抽取出相似类的公共属性和操作，以建立这些类的新父类，并在类等级的不同层次中正确地定义各个服务。

小结

- 1、面向对象分析的基本过程
- 2、需求陈述
- 3、建立对象模型
- 4、建立动态模型
- 5、建立功能模型
- 6、定义服务