

# 软件工程：第九章 面向对象分析

---

## 导学目标

- 了解面向对象分析的目标、关键要素及过程
- 掌握建立对象模型、动态模型、功能模型的步骤
- 了解服务的相关知识

## 第一节 面向对象分析概述

1、无论我们采用何种方式开发软件，分析的过程都是提取需求的过程。分析工作主要包括3项内容，理解、表达、验证。

- **理解**：系统分析员通过与用户及领域专家交谈，力求完全理解用户需求和该应用领域内关键性的背景知识
- **表达**：用某种无二义性的方式把这种理解表达成文档资料
- **验证**：由于问题复杂，交流过程中可能出现偏差，理解可能达不到预期的效果，因此我们需要去验证文档资料的正确性，若有问题，需要重新进行修改

### 2、面向对象分析(OOA)的关键

- 识别问题域内的类与对象，并分析它们之间的关系，建立起针对于问题域的正确模型

### 3、面向对象分析(OOA)的定义

- 抽取和整理用户需求并建立问题域精确模型的过程

### 4、面向对象分析(OOA)的过程

1. 获取用户需求
2. 建立模型(需要向领域专家学习并自信研究类似的问题域)
3. 书写需求规格说明书
4. 复审

### 5、面向对象建模中的三个要素

- 对象模型：几乎在解决任何一个问题
- 动态模型：当问题设计到交互和时序时
- 功能模型：解决运算量很大的问题

### 6、复杂问题的对象模型：(5个层次)

- 主题层
- 类与对象层
- 结构层
- 属性层
- 服务层

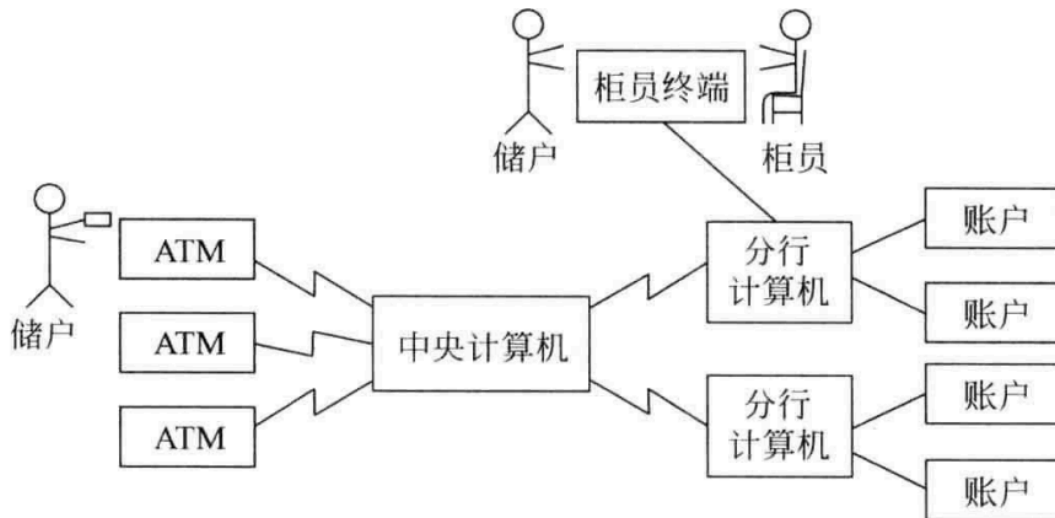
7、需求陈述的内容包括问题的范围、功能需求、性能需求、应用环境及假设条件等。需求陈述说明的是做什么，而不是怎样去做

### 8、书写需求陈述的要点

- 要尽力做到语法准确
- 不要把实际需求和设计决策混为一谈
- 需求陈述可繁可简
- 需求陈述要具体无二义性、完整性和一致性

## 需求陈述案例

### 银行ATM取款机



#### 1、需求陈述

- 某银行拟开发一个自动取款机系统，它是一个由自动取款机、中央计算机、分行计算机及柜员终端组成的网络系统。ATM和中央计算机由总行投资购买【ATM系统的组成】
- 总行拥有多台ATM，分别设在全市各主要街道上。分行负责提供分行计算机和柜员终端。柜员终端设在分行营业厅及分行下属的各个储蓄所内。该系统的软件开发成本由各个分行分摊【总行ATM以及分行ATM的功能和应用领域】
- 银行柜员使用柜员终端处理储户提交的储蓄事务。储户可以用现金或支票向自己拥有的某个账户内存款或开新账户。储户也可以从自己的账户中取款。通常，一个储户可能拥有多个账户【储户和ATM系统和账户之间的关系】
- 柜员负责把储户提交的存款或取款事务输进柜员终端，接收储户交来的现金或支票，或付给储户现金。柜员终端与相应的分行计算机通信，分行计算机具体处理针对某个账户的事务并且维护账户【储户、柜员、分行计算机、账户之间的关系】
- 拥有银行账户的储户有权申请领取现金兑换卡。使用现金兑换卡可以通过ATM访问自己的账户。目前仅限于用现金兑换卡在ATM上提取现金(即取款)，或查询有关自己账户的信息(例如，某个指定账户上的余额)。将来可能还要求使用ATM办理转账、存款等事务
- 所谓现金兑换卡就是一张特制的磁卡，上面有分行代码和卡号。分行代码唯一标识总行下属的一个分行，卡号确定了这张卡可以访问哪些账户。通常，一张卡可以访问储户的若干个账户，但是不一定能访问这个储户的全部账户
- 每张现金兑换卡仅属于一个储户所有，但是，同一张卡可能有多个副本，因此，必须考虑同时在若干台ATM上使用同样的现金兑换卡的可能性。也就是说，系统应该能够处理并发的访问【系统要求：并发访问】
- 当用户把现金兑换卡插入ATM之后，ATM就与用户交互，以获取有关这次事务的信息，并与中央计算机交换关于事务的信息
- 首先，ATM要求用户输入密码，接下来ATM把从这张卡上读到的信息以及用户输入的密码传给中央计算机，请求中央计算机核对这些信息并处理这次事务

- 中央计算机根据卡上的分行代码确定这次事务与分行的对应关系，并且委托相应的分行计算机验证用户密码
- 如果用户输入的密码是正确的，ATM就要求用户选择事务类型(取款、查询等)。当用户选择取款时，ATM请求用户输入取款额。最后，ATM从现金出口吐出现金，并且打印出账单交给用户

## 第二节 建立对象模型

面向对象分析的首要工作，就是**建立问题域的对象模型**

对象模型描述了类与对象以及它们之间的相互关系，表示了目标系统的静态数据结构

静态数据结构对应用细节依赖较少，比较容易确定；当用户需求发生变化时，静态数据结构相对比较稳定。因此，用面向对象方法开发出绝大多数软件时，都首先建立对象模型，然后再建立其他两个子模型

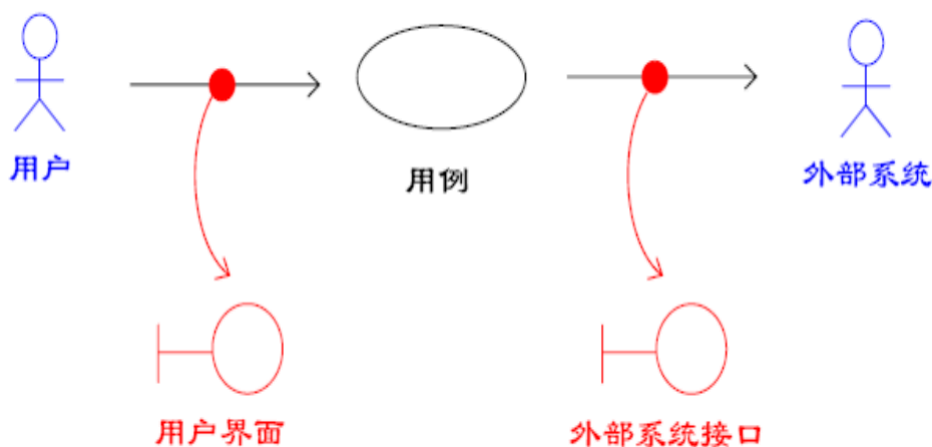
### 1、建立对象模型的步骤

1. 确定分析类
2. 确定类的关联
3. 划分主题
4. 确定属性
5. 识别继承
6. 反复修改

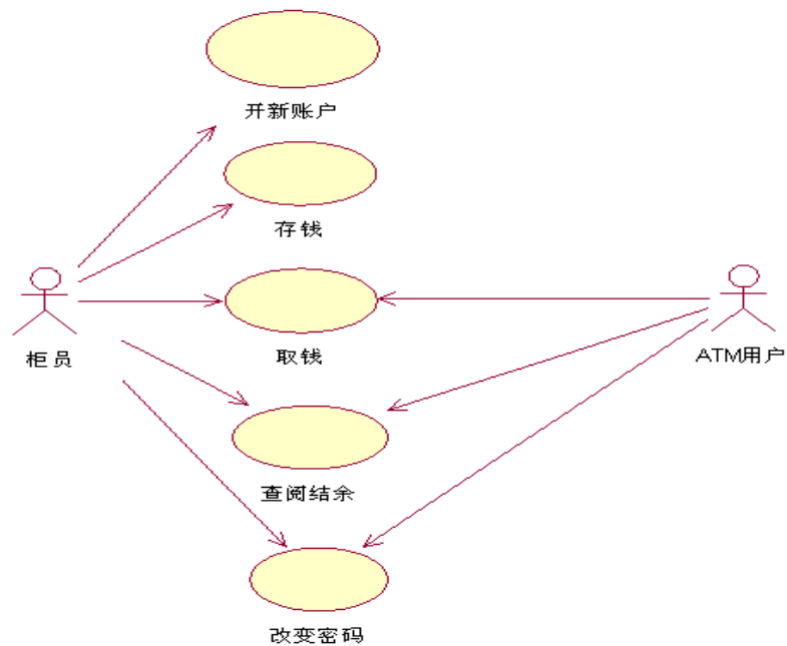
### 9.2.1 确定类与对象

#### 1、找出候选的类与对象

- **边界类**
  - 通常一参与者与一用例之间交互或通信关联对应一边界类



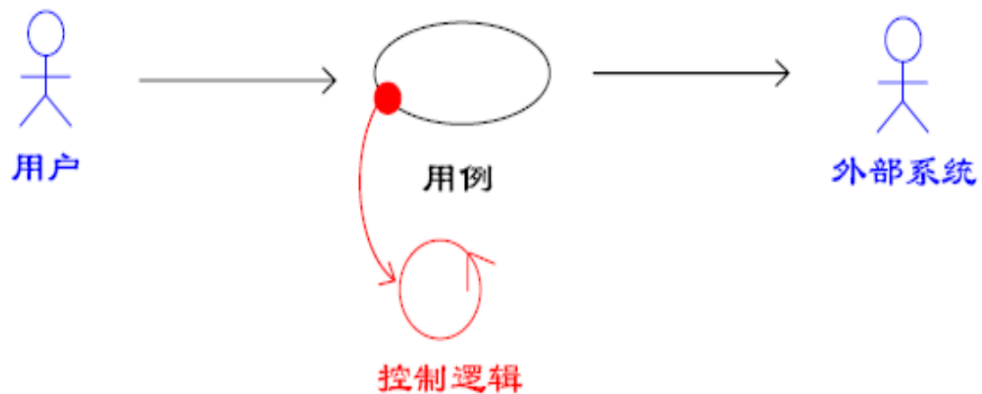
- 示例：ATM的系统用例图



边界类	说明
SetupForm	开新账户的操作界面
DepositForm	存款的操作界面
ATMWithdrawForm	ATM用户取款的操作界面
ATMBalanceForm	ATM查询余额的操作界面
ATMPasswordForm	ATM改变密码的操作界面
TellerWithdrawForm	Teller用户取款的操作界面
TellerBalanceForm	Teller查询余额的操作界面
TellerPasswordForm	Teller改变密码的操作界面

## • 控制类

- 控制类负责协调边界类和实体类，通常在现实世界没有对应的事物。一般来说，一个用例对应一个控制类

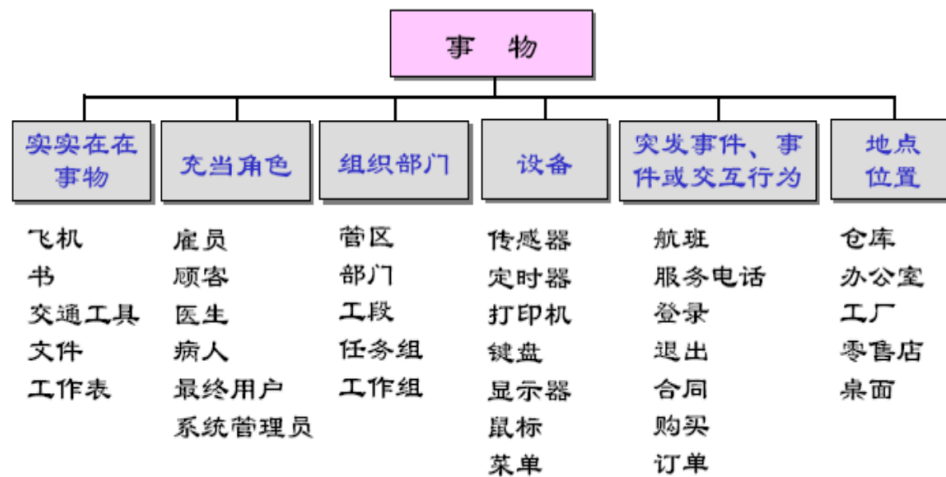


## ■ 示例

控制类	说明
SetupControl	负责执行开新账户
DepositControl	负责执行存款
WithdrawControl	负责执行取款
BalanceControl	负责执行查询余额
PasswordControl	负责执行改变密码

## • 实体类

- 实体类通常是用例中的参与对象，对应着现实世界中“事物”



- 非正式分析法：提取需求陈述中的名词

- 示例：用非正式分析法提取ATM系统中的实体类

- 银行，自动取款机(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
    - ATM系统分析员根据领域知识或常识提取出隐含的类。如通信链路、事务日志等

## 2、筛选出正确的类与对象

显然根据非正式分析方法只能确定候选的类与对象，还需要通过严格的标准去筛选出正确的，去掉不正确的

- 筛选的标准

- 冗余

- 如果两个类表达了同样的信息，应该保留此问题域中最富于描述力的名称。
    - 示例：储户与用户，现金兑换卡与磁卡及副本应去掉“用户”、“磁卡”、“副本”，保留“储户”和“现金兑换卡”

- 无关

- 如果类与当前要解决的问题无关，需要删除与本问题密切相关类放进目标系统，去掉“成本”、“市”、“街道”、“营业厅”、“储蓄所”

- 笼统——直接去掉

- 示例：银行(总行和分行)、系统、软件、信息、访问(事务)

- 属性——直接去掉

- 示例：现金、支票、取款额、账单、余额、分行代码、卡号、密码和类型

- 操作

- 需求陈述中既作名词又作动词的词，慎重考虑是作类合适，还是作类中操作合适

- 实现——直接去掉

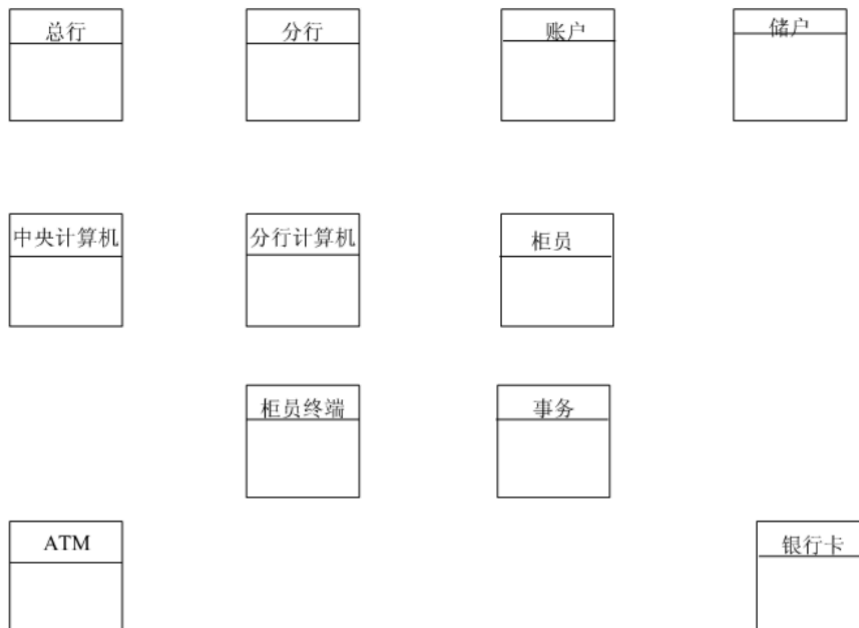
- 事务日志、通信链路

- 经过初步筛选还剩下的类

- ATM系统筛选后的类：

- 银行, 自动取款机(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, 访问确定类与对象

- ATM系统的实体分析类



## 9.2.2 确定关联

### 1、初步确定关联

- 步骤
  1. 提取需求陈述中的动词词组
  2. 发现隐含关联
  3. 与用户及领域专家讨论后补充
- 示例：ATM系统
  1. 直接提取动词短语得出的关联
    - 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
    - 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

- 初步分析得出的关联只能作为候选关联，还需要做进一步的筛选
- 筛选的标准

### 1. 已删去的类之间的关联

- 删掉某候选类，与这个类有关的关联也删去，或重新表达
  - 已删去“系统”、“网络”、“市”、“街道”、“成本”、“软件”、“事务日志”、“现金”、“营业厅”、“储蓄所”、“账单”候选类，关联也应删去：
  - ATM、中央计算机、分行计算机及柜员终端组成网络
  - ATM设在主要街道上
  - 分行分摊软件开发成本
  - 系统提供必要安全性
  - 系统维护事务日志
  - ATM吐出现金
  - ATM打印账单
  - 柜员终端设在分行营业厅及储蓄所内

### 2. 与问题无关的或应在实现阶段考虑的关联应该删去

- 如ATM系统的例子中，“系统处理并发的访问”需要删去。

### 3. 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
- 以ATM系统为例，“ATM读现金兑换卡”、“ATM与用户交互”、“中央计算机确定事务与分行对应关系”隐含“中央计算机与分行通信”

### 4. 三元或者三元以上的关联

- 三个或三个以上对象关联，可分解为二元关联或限定关联
- 在ATM系统例子中，“柜员输入针对账户的事务”分解成“柜员输入事务”和“事务修改账户”





- 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

加上属性的实体类图

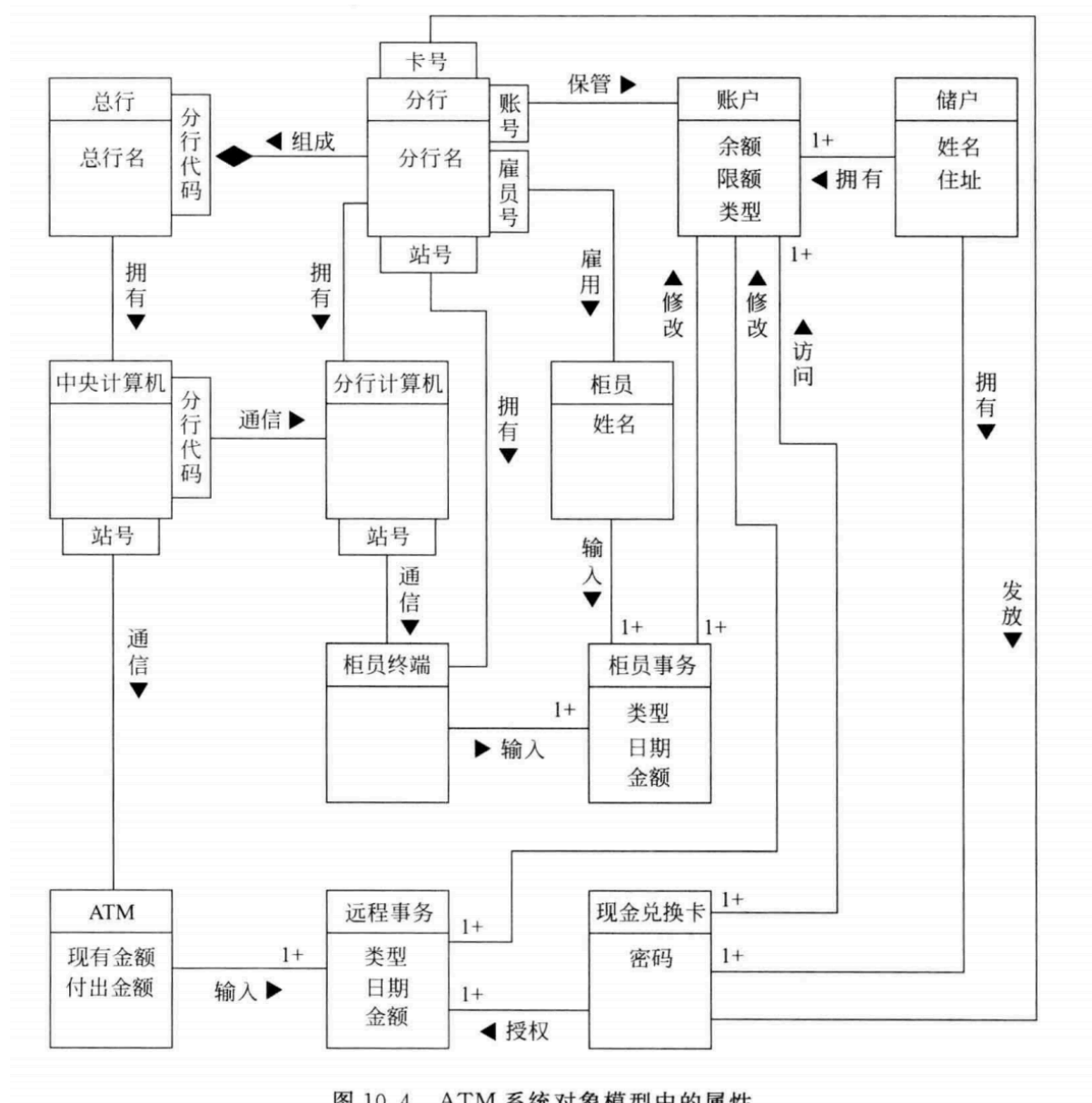
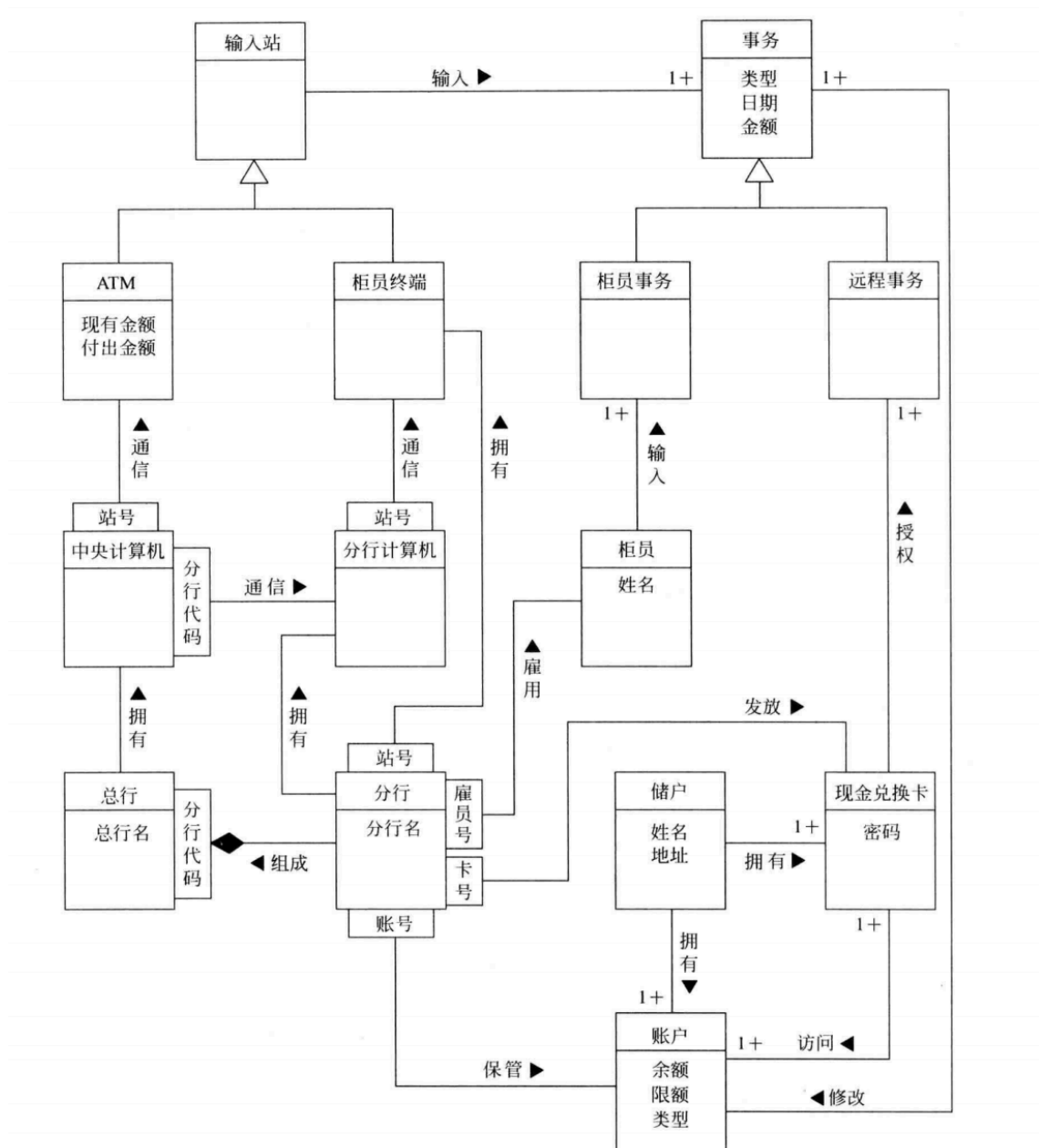


图 10.4 ATM 系统对象模型中的属性

## 9.2.5 识别继承关系

- 一般说来，可以有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
  - 自顶向下：把现有类细化成子类
- 识别出继承关系的实体类图



## 9.2.6 反复修改

下面以ATM为例，讨论可能做的修改

### 1. 分解“现金兑换卡类”

- 现金兑换卡有两独立功能：标志储户访问账号的权限；含有分行代码和卡号的数据载体。（卡权限和现金兑换卡）

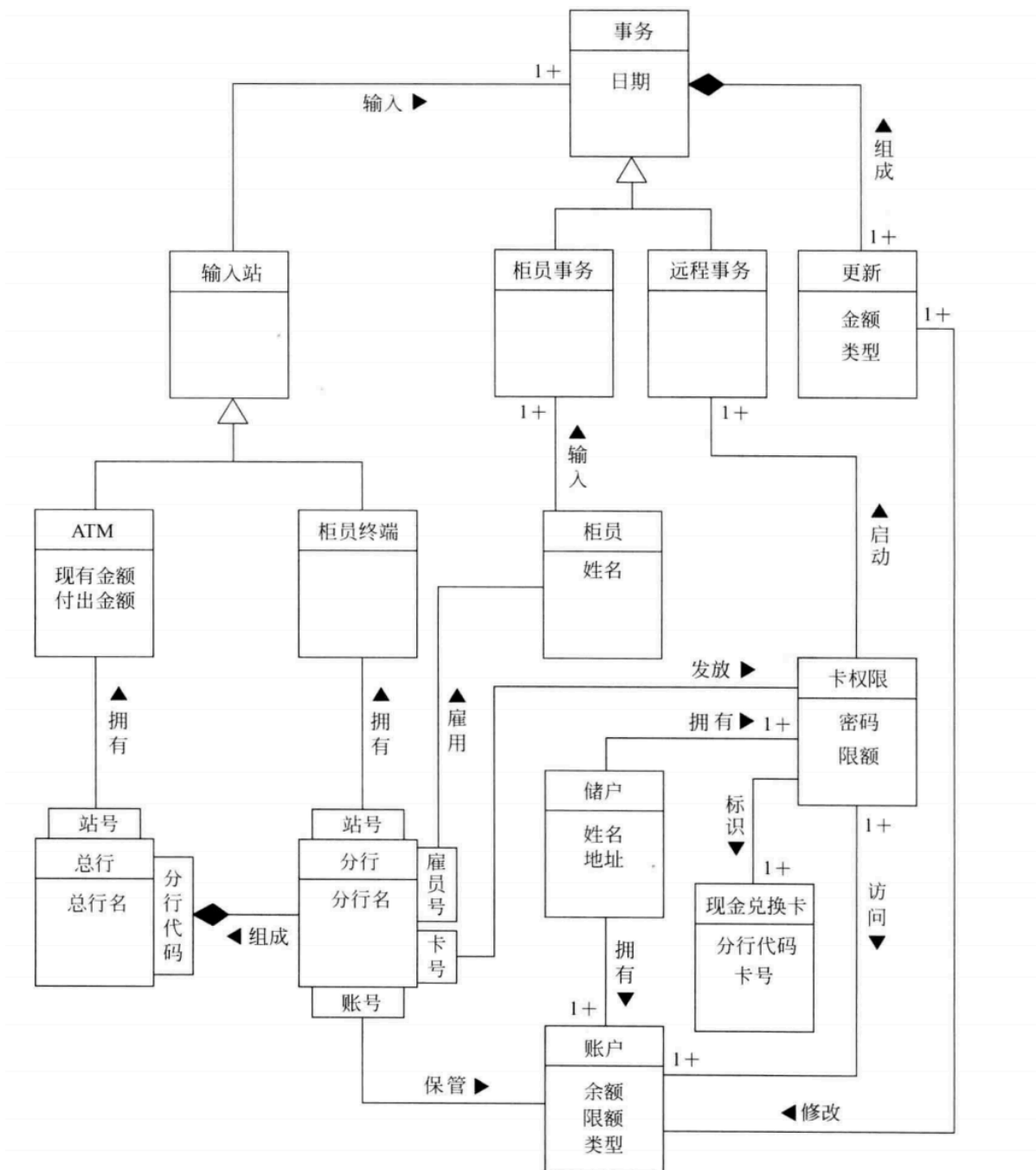
### 2. “事务”由“更新”组成

- 更新包括取款、存款、查询。有自己属性(类型、金额)，应独立存在

### 3. 合并“分行”和“分行计算机”

- 类似的应该合并“总行”与“中央计算机”

修改后的实体类图



### 第三节 建立动态模型

#### 1、应用场景

- 开发交互式系统时，动态模型十分重要

#### 2、建立动态模型的步骤

- 编写典型交互行为脚本
- 从脚本中提取事件及相关对象，用顺序图表达
- 确定对象状态及状态间转换关系，用状态图描绘
- 比较各个对象的状态图，确保事件之间的匹配性

### 9.3.1 编写脚本

#### 1、定义

- 系统在某一执行期内出现的一系列事件

#### 2、目的

- 保证不遗漏重要的交互步骤

#### 3、范围

- 并不固定，主要由编写脚本的目的决定

#### 4、顺序

- 首先编写正常情况下的脚本
- 考虑特殊情况
- 考虑出错情况

#### 5、编写脚本示例：以ATM为例

##### 1. 正常情况

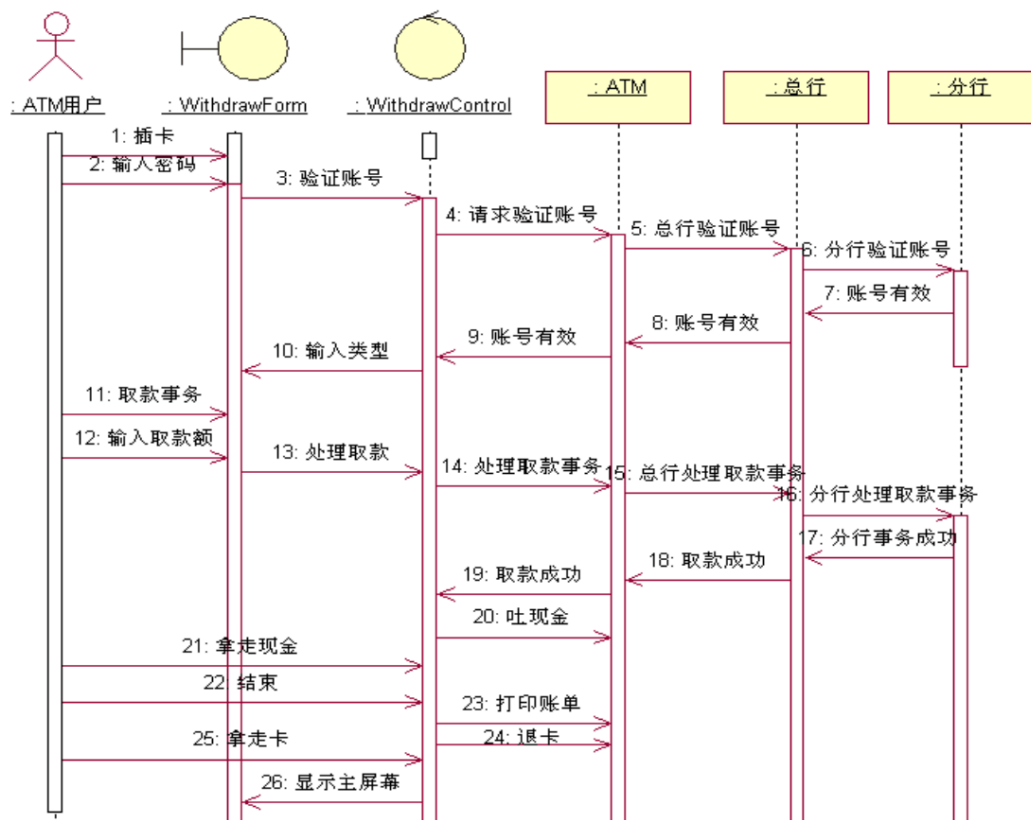
- ATM请储户插卡；储户插入一张现金兑换卡。
- ATM接受该卡并读它上面的分行代码和卡号。
- ATM要求储户输入密码；储户输入自己密码“1234”等数字。
- ATM请求总行验证卡号和密码；总行要求“39”号分行核对
- 储户密码，通知ATM这张卡有效。
- ATM要求储户选择事务类型(取款、转账、查询等)；储户选择“取款”。
- ATM要求储户输入取款额；储户输入“880”

##### 2. 异常情况

- 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；总行向有关银行咨询后拒绝这张卡。
- ATM显示“密码错”，请用户重新输入密码；储户输入1234；
- ATM请总行验证后知道输入的密码正确
- ATM请储户选择事务类型；储户选择“取款”
- ATM询问用户取款额；储户不想取款，敲“取消”键
- ATM退出现金兑换卡，请储户拿走；储户拿走
- ATM请储户插卡

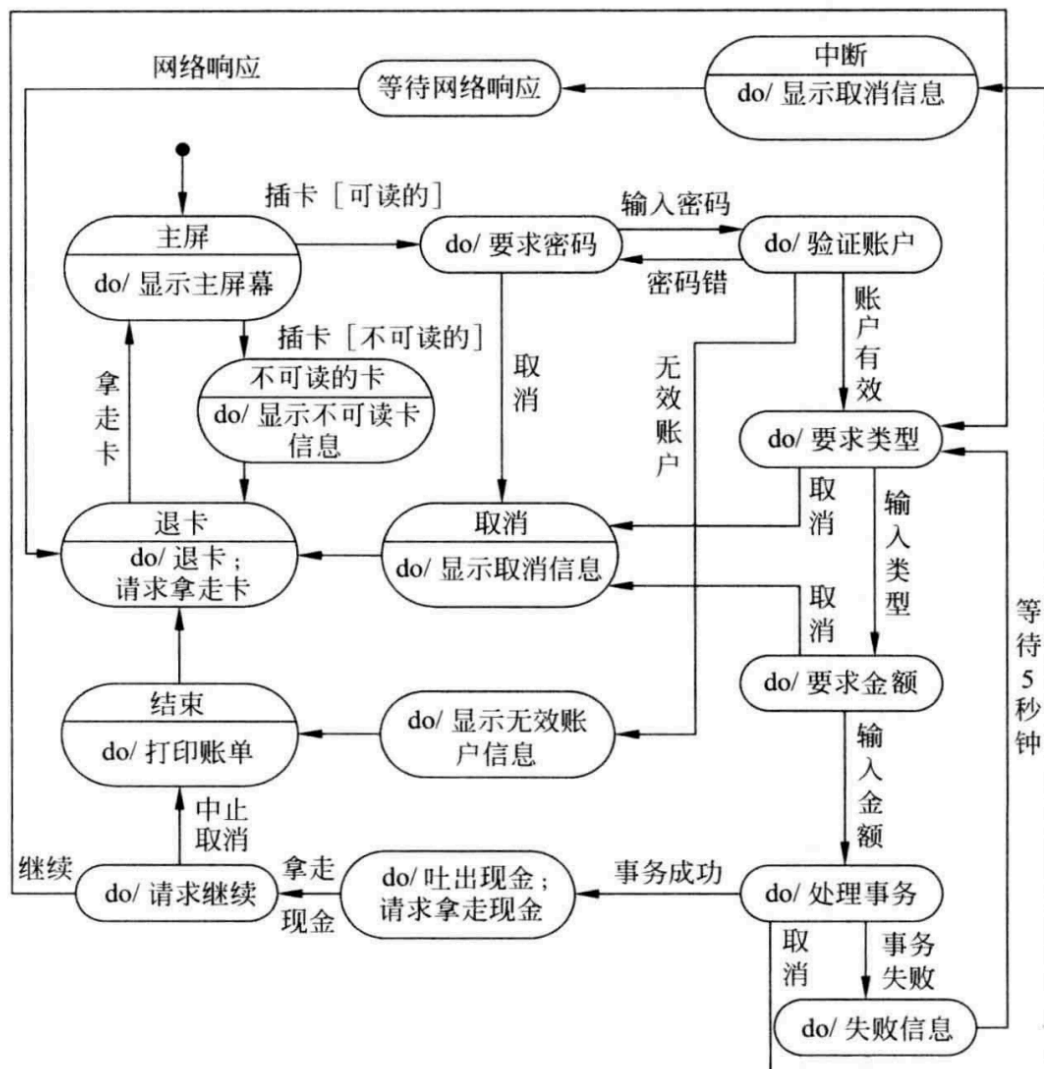
### 9.3.2 画顺序图

- 从脚本提取所有外部事件，确定每类事件发送和接收对象
- 针对系统中的典型功能，画出顺序图
- ATM用户取款顺序图示例：

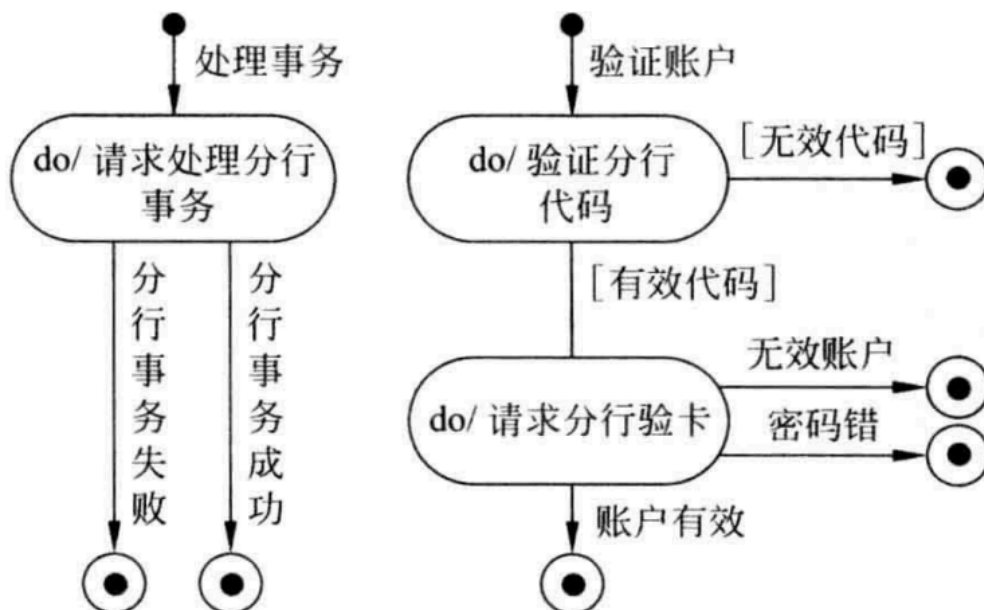


### 9.3.3 画状态图

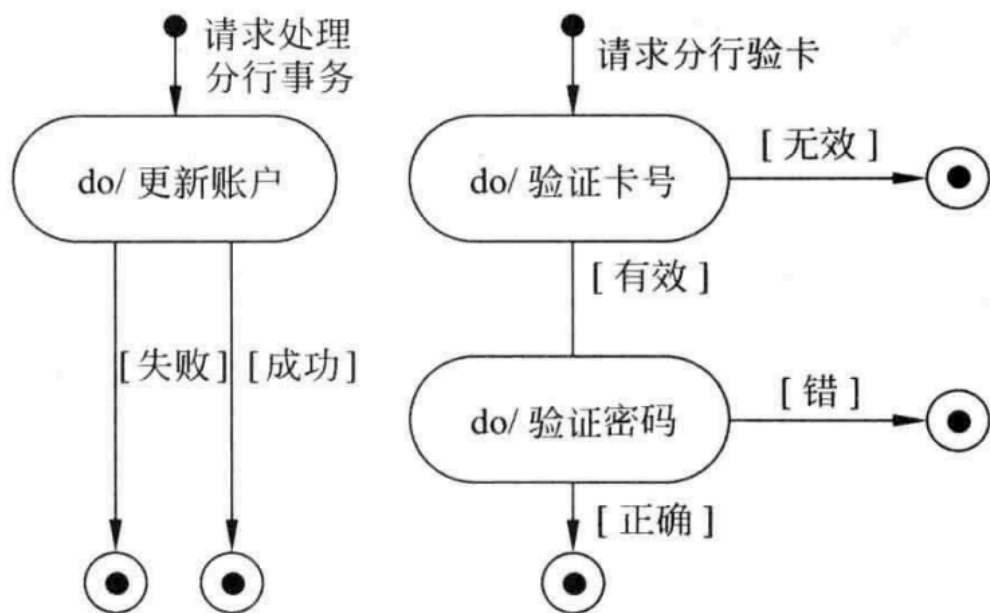
- 用一张状态图描绘类的行为，集中考虑具有交互行为类
- 根据一张顺序图画出生状态图之后，再把其他脚本的顺序图并进来
- ATM类的状态图



- 总行类的状态图



- 分行类的状态图



## 第四节 建立功能模型

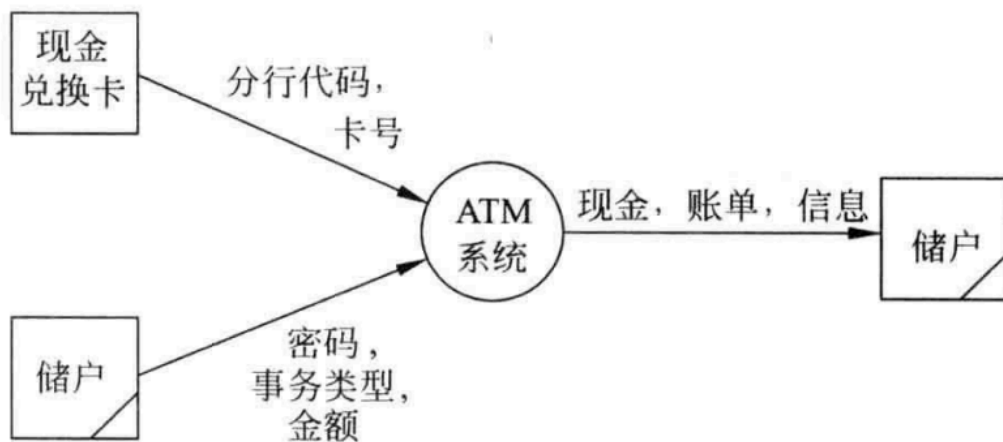
功能模型表明了数据之间的依赖关系以及有关的数据处理功能，它由一组数据流图组成

### 1、建立功能模型的步骤

1. 画出基本系统模型图
2. 画出功能级数据流图
3. 描述处理框功能

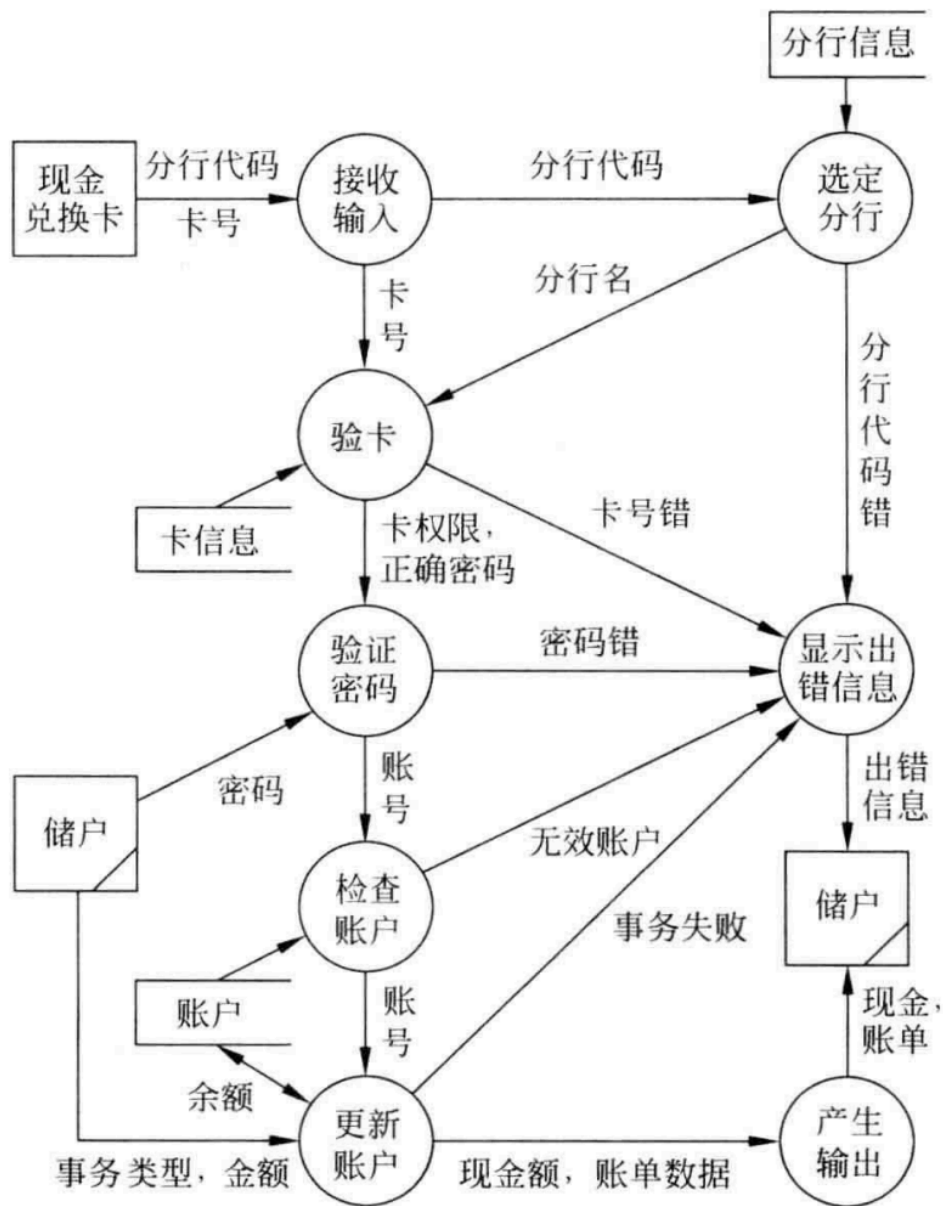
### 2、画出基本系统模型图

- 基本系统模型由若干个数据源点和终点，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。基本模型指明了目标系统的边界
- 示例：ATM系统的基本系统模型



### 3、画出功能级数据流图

- 把基本模型中的单一处理框分解成若干个处理框，就可以得到功能级数据流图
- 示例：ATM系统的功能级数据流图



### 3、描述处理框功能

- 把数据流图细化分解到一定程度，就可以描述各个处理框功能了
- 描述可以是说明性的，也可以是过程性的
- 示例：ATM中更新账户这个处理功能的描述
  - 更新账户(账号、事务类型、金额) -> 现金额、账单数据、信息
  - 如果取款金额超过该账户当前金额，拒绝该事务且不付出现金
  - 如果取款额不超过账户当前余额，从余额中减去取款额作为新的余额，付出储户要取的现金
  - 如果事务是存款，把存款额追加到余额中得到新余额，不付出现金
  - 如果事务是查询，不付出现金
  - 在上述任何一种情况下，账单内容都是：ATM号，日期，时间，账号，事务类型，事务金额，新余额



## 第五节 定义服务

- 常规行为
  - 定义类中每个属性都是可以访问的
- 从事件导出操作
- 与数据流图中处理框对应的操作
- 利用继承减少冗余操作
  - 应该尽量利用继承机制减少所需定义的服务数目