

面向对象分析与设计 Object-Oriented Analysis and Design

北京理工大学软件学院
马 锐
Email: mary@bit.edu.cn

第11章 分析问题

11.1 分析过程概述

11.2 静态分析

11.3 动态分析

11.1 分析过程概述(1)

➤ 分析模型分类

- **静态分析**: 使用类图描述, 显示对象及其对象之间的关系
- **动态分析**: 使用通信图证明静态模型是可行的

➤ 分析过程

- 使用系统需求模型查找候选的类, 以描述与系统相关的对象, 并在类图上建立它们
- 确定类之间的关系(关联、聚合、复合和继承)

11.1 分析过程概述(2)

- 确定类的属性(对象的、已指定的简单特性)
- 检查系统用例, 确定已有的对象支持它们。在检查过程中微调类、属性和关系, 这个用例的实现过程将生成一些操作, 来补充属性
- 需要时更新术语表和非功能需求——用例本身不需要更新, 但可能需要某些更正

11.2 静态分析

- 11.2.1 确定类
- 11.2.2 标识类的关系
- 11.2.3 绘制类图和对象图
- 11.2.4 属性
- 11.2.5 关联类
- 11.2.6 有形对象和无形对象
- 11.2.7 好的对象

11.2.1 确定类(1)

➤ 识别对象和类

- **考虑问题域**
 - 侧重客观事物与系统中对象的映射
 - 可以启发分析员发现对象的因素包括: 人员、组织机构、物品、设备、事件(如索赔、上访、交易)、表格、日志、报告和结构等
- **考虑系统边界**
 - 把一些人员和设备看作问题域范畴以内的事物, 系统中的对象是对它们的抽象描述。侧重于以系统中的对象模拟现实中的人和设备

11.2.1 确定类(2)

- 对系统边界之外与系统进行交互的参与者，系统中需要设立相应的对象处理系统与这些实际的人和设备的交互，侧重于以系统中的对象处理现实中的人和系统与系统的交互
- 在系统中设立对象处理与外系统的接口
- 考虑系统责任
 - 对照系统责任要求的每一项功能，查看是否可以由现有对象完成这些功能。若发现某些功能在现有对象中都不能提供，则可启发我们发现问题域中某些遗漏的对象

7

11.2.1 确定类(3)

- 如果已经建立了系统用例，则使用用例产生场景，并使用场景寻找已错过的类
- 名词技术
 - 从名词到对象或类通常有一对一的映射。运用名词、代词和名词短语识别对象和类（寻找“现实世界”的对象和类）
 - 用单个的专有名词或代词（Jim、他、她、雇员号、我的工作站、我的家）以及直接引用的名词（第六个参赛者、第一百万次购买）识别对象

8

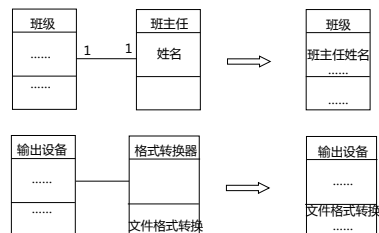
11.2.1 确定类(4)

- 用复数名词（人们、顾客们、开发商们、用户们）以及普通名词（人、顾客、雇员、工作站）来识别类
- 审查与筛选
 - 舍弃无用的对象
 - 通过属性判断：是否通过属性记录了某些有用的信息？
 - 通过操作判断：是否通过操作提供了某些有用的功能？
 - 二者都不是——无用

9

11.2.1 确定类(5)

- 对象的精简
 - 只有一个属性/一个操作的对象



10

11.2.1 确定类(6)

- 对职责过多的类进行分解
- 与实现条件有关的对象推迟到OOD考虑
- 系统责任所要求的某些功能——例如系统安装、配置、信息备份、浏览——可能无法从问题域中找到相应的对象来提供这些功能，可在设计阶段考虑专门为它们增加一些对象，既把它们推迟到设计阶段考虑
- 系统责任要求的某些功能可能与实现环境有关，也推迟到设计阶段考虑。例如：与图形用户界面（GUI）、数据管理系统、硬件和操作系统有关的对象

11

11.2.1 确定类(7)

- 对类进行调整
 - 类的属性/操作不适合该类的全部对象
 - 例：“汽车”类的“乘客限量”属性，不适用于吊车、铲车
 - 问题：分类不够详细
 - 进一步划分特殊类
 - 属性及操作相同的类
 - 经过抽象，差别很大的事物只保留相同的特征
 - 考虑能否合并为一个类，如计算机与吸尘器

12

11.2.1 确定类(8)

- 属性及操作相似的类
 - 考虑能否提升出一个一般类，或形成整体部分结构
 - 例：轿车、货车，抽象出汽车；机床与鼓风机用电机做成分
- 同一事物的重复描述
 - 例：“职员”和“工作证”
 - 取消其中一个

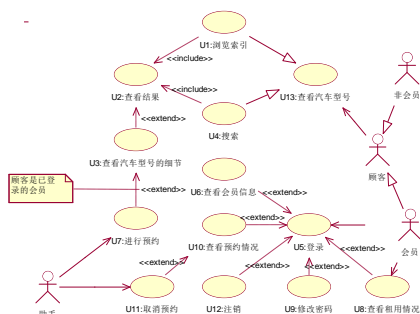
13

11.2.1 确定类(9)

- 类的命名
 - 适合该类及其特殊类的全部对象实例
 - 汽车加摩托车->机动车；马车->车辆
 - 反映个体而不是群体
 - 书-书籍；船-船舶
 - 使用名词或带有定语的名词，避免无意义的符号
 - 如线装书
 - 考虑使用适当种类的语言文字对类命名
 - 在中国：可用中、英文双重命名

14

11.2.1 确定类(10)



15

11.2.1 确定类(11)

- 顾客(Customer)
- 会员(Member)
 - 登录账号(InternetAccount)
 - 信用卡(CreditCard)
 - 地址(Address)
- 非会员(NonMember)
- 助手(Make)

16

11.2.1 确定类(12)

- 汽车型号(CarModel)
 - 汽车型号的详细信息(CarModelDetails)
- 汽车(Car)
 - 汽车的详细信息(CarDetails)
- 生产汽车的厂商(Vendor)
- 汽车目录信息(Category)
- 租车(Rental)
- 预约(Reservation)

17

11.2.2 标识类的关系(1)

- 类之间的关系
 - 继承
 - 关联
 - 聚合
 - 复合
 - 使用频率：关联 > 聚合 > 继承 > 复合

18

11.2.2 标识类的关系(2)

继承关系

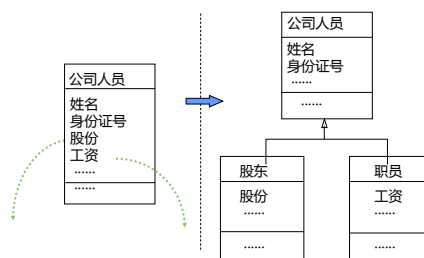
识别继承

- 学习当前领域的分类学知识
- 按常识考虑事物的分类
- 使用继承的语义
 - Is a kind of
- 考察类的属性与操作
 - 从一个类中划分出特殊类
 - 从具有共同属性/操作的类中构建继承
- 考虑领域范围内的复用

19

11.2.2 标识类的关系(3)

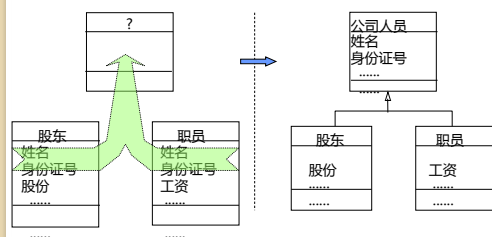
从一个类中划分出特殊类



20

11.2.2 标识类的关系(4)

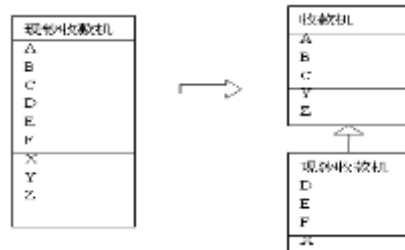
从具有共同属性与操作的类中构建继承



21

11.2.2 标识类的关系(5)

为支持复用建立继承关系



22

11.2.2 标识类的关系(6)

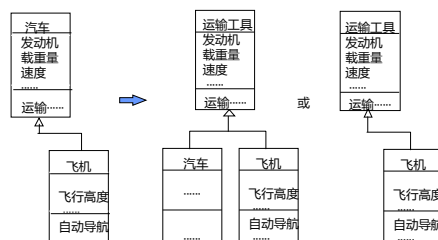
审查调整

- 问题域是否需要这样的分类? (例: 书—善本书)
- 系统责任是否需要这样的分类? (例: 职员—本市职员)
- 是否符合分类学的常识? (符合“is a kind of”关系)
- 是否构成了继承关系? (确实继承了一些属性或操作, 如航标船与一般的船)

23

11.2.2 标识类的关系(7)

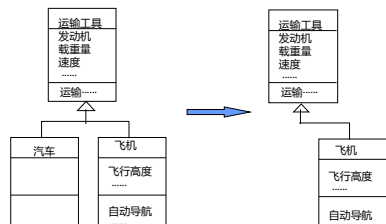
违反常识的继承结构及其修改



24

11.2.2 标识类的关系(8)

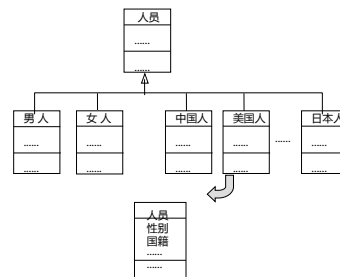
- 一般-特殊结构的简化
- 取消没有特殊性的特殊类



25

11.2.2 标识类的关系(9)

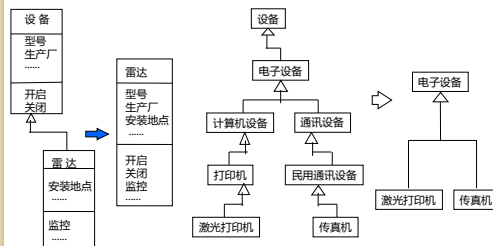
- 增加属性简化一般-特殊结构



26

11.2.2 标识类的关系(10)

- 取消用途单一的一般类，减少继承层次



27

11.2.2 标识类的关系(11)

- 一般类存在的价值
 - 有两个或两个以上的特殊类
 - 需要用它创建对象实例
 - 它的存在有助于软件复用
- 建立关联
 - 认识对象之间的静态联系
 - 考虑问题域和系统责任——哪些类的对象之间的关系需要在系统中表达
 - 认识关联的属性与操作

28

11.2.2 标识类的关系(12)

- 对于考虑中的每一种关联，进一步分析它是否应该带有某些属性和操作。就是说，是否含有一些仅凭一个简单的关联不能充分表达的信息
- 分析并表示关联的多重性
- 从连接线的每一端，看本端的一个对象可能与另一端的几个对象发生连接，把结果标注到连接线的另一端
- 分析关联的性质
 - 关联角色

29

11.2.2 标识类的关系(13)

- 聚合关系
 - 识别聚合
 - 物理上的整体事物和它的组成部分
 - 例：机器、设备和它的零部件
 - 组织机构和它的下级组织及部门
 - 例：公司与子公司、部门
 - 团体（组织）与成员
 - 例：公司与职员
 - 一种事物在空间上包容其它事物
 - 例：生产车间与机器

30

11.2.2 标识类的关系(14)

- ✦ 抽象事物的整体与部分
 - 例：学科与分支学科、法律与法律条款
- ✦ 具体事物和它的某个抽象方面
 - 例：人员与身份、履历
- ✦ 在材料上的组成关系
 - 例如，面包由面粉、糖和酵母组成，汽车是由钢、塑料和玻璃组成

31

11.2.2 标识类的关系(15)

- ✦ 审查与筛选
 - ✦ 是否属于问题域？
 - 例：公司职员与家庭
 - ✦ 是不是系统责任的需要？
 - 例：公司与工会
 - ✦ 部分对象是否有一个以上的属性？
 - 例：汽车与轮胎（规格）
 - ✦ 是否有明显的整体-部分关系？
 - 例：学生与课程，谁是整体？谁是部分？

32

11.2.2 标识类的关系(16)

- 顾客(Customer)：一般类
- 会员(Member)：特殊类
 - ✦ 登录账号(InternetAccount)：一对一关联
 - ✦ 信用卡(CreditCard)：多对一关联
 - ✦ 地址(Address)：多对一关联
- 非会员(NonMember)：特殊类
- 助手

33

11.2.2 标识类的关系(17)

- 复合
 - ✦ 汽车(Car)
 - ✦ 汽车的详细信息(CarDetails)
 - ✦ 汽车型号(CarModel)
 - ✦ 汽车型号详细信息(CarModelDetails)

34

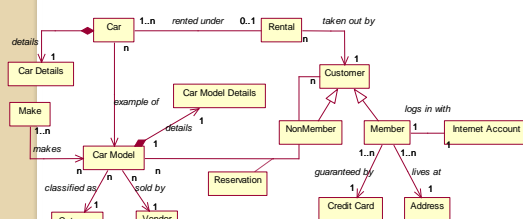
11.2.2 标识类的关系(18)

- 关联
 - ✦ 汽车—汽车型号：多对一
 - ✦ 汽车—生产厂商：多对一
 - ✦ 汽车目录信息—汽车(型号)：多对一
- ✦ 顾客—租车：一对多
- ✦ 汽车—租车：多对多
- ✦ 助手—汽车型号：多对多
- ✦ 顾客—预约—汽车型号：多对多，关联类

35

11.2.3 绘制类图 and 对象图(1)

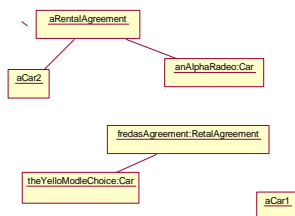
➢ iCoot的分析类图



36

11.2.3 绘制类图和对象图(2)

➤ 运行时的对象



37

11.2.4 属性(1)

➤ 识别属性

- ❖ 按常识这个对象应该有哪些属性?
- ❖ 在当前的问题域中对象应该有哪些属性?
- ❖ 根据系统责任这个对象应具有哪些属性?
- ❖ 建立这个对象是为了保存和管理哪些信息?
- ❖ 对象为了完成其功能需要增设哪些属性?
- ❖ 对象是否需要通过专设的属性区别其状态?
- ❖ 可利用需求文档中的形容词或所有短语

38

11.2.4 属性(2)

➤ 筛选

- ❖ 是否体现了以系统责任为目标的抽象; 例: 书的重量?
- ❖ 是否描述对象本身特征; 例: 课程—教师, 课程—教师电话号码?
- ❖ 一个属性对应的一个事物性质的粒度要适当
- ❖ 例如人的通信地址, 包括国家、省、城市、街道、门牌号码等内容, 但这些内容在通信地址这个概念上是不可分的。在定义“人员”对象的属性时, 应该使用一个属性“通信地址”, 而不应把有关通信地址的各项内容拆散开用多个属性来描述

39

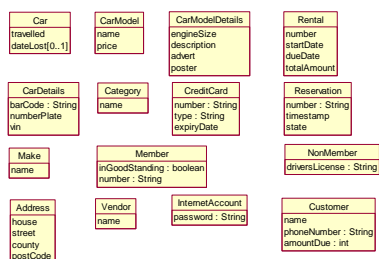
11.2.4 属性(3)

- ❖ 若一个对象与另一个对象有关系, 其属性必须捕获该对象的性质, 而不是关系或关系中的其他对象的性质;
- ❖ 如属性必须是整个实体的特征, 而不是其成分的特征
- ❖ 是否可通过继承得到
- ❖ 可以从其它属性直接导出的属性
- ❖ 与实现有关的因素, 推迟到OOD考虑 (规范化、对象标识、性能问题)

40

11.2.4 属性(4)

➤ iCoot的属性

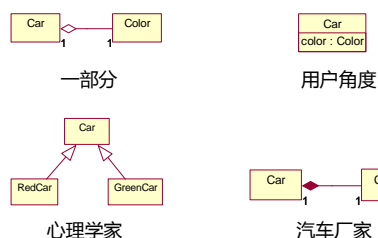


41

11.2.4 属性(5)

➤ 属性与关系

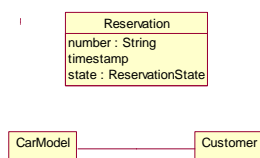
- ❖ 示例: 汽车与汽车的颜色



42

11.2.5 关联类

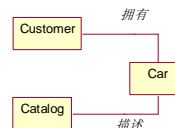
➤ iCoot的关联类



43

11.2.6 有形对象和无形对象(1)

- 考虑“汽车”、“汽车类别”与“顾客”三个类
- 假定除了销售汽车之外，经销商还给顾客提供服务



44

11.2.6 有形对象和无形对象(2)

➤ 与销售相关的信息包括：

- ❖ **modelNumber**：这类汽车的型号
- ❖ **availableColors**：这类汽车出厂前可喷涂的颜色
- ❖ **numberOfCylinders**：这类汽车引擎拥有的气缸数

➤ 与服务相关的信息包括：

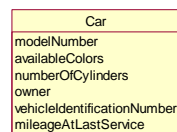
- ❖ **Owner**：汽车的注册主人
- ❖ **vehicleIdentificationNumber**：汽车的唯一标识
- ❖ **mileageAtLastService**：上次保养后汽车已行驶里程

45

11.2.6 有形对象和无形对象(3)

➤ 错误的建模

- ❖ 避免使一个类有两组完全不同的任务——这种类的内聚力很脆弱，它们的任务也不会构成一个模块



46

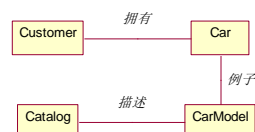
11.2.6 有形对象和无形对象(4)



47

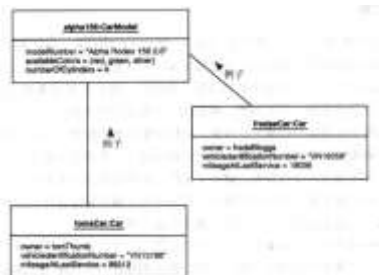
11.2.6 有形对象和无形对象(5)

➤ 正确的建模



48

11.2.6 有形对象和无形对象 (6)



49

11.2.7 好的对象

- 类取自用例中的名词
- 对象是具体的事物，是对类的实例化
- 对象拥有属性
- 对象能做什么，即对象操作
- 建议：在动态分析类之前，不要过多地关注操作

50

11.3 动态分析(1)

- 进行动态分析的原因
 - ❖ 确认类图是完整、正确的，以便尽早更正错误，包括添加、删除或修改类、关系、属性和操作
 - ❖ 相信当前的模型可以在软件中实现
 - ❖ 验证最终系统上用户界面的功能：在进行详细设计之前，最好按照用例中的线条，把对系统的访问放在各个界面上

51

11.3 动态分析(2)

- 动态分析中最重要的部分是用例的实现
- 用例实现的步骤
 - ❖ 检查系统用例，模拟对象之间发送的消息，在通信图上记录结果
 - ❖ 在接收消息的对象上引入操作
 - ❖ 根据需要添加类，以标识边界（系统接口）和控制器（复杂业务过程的占位符或者对象的创建和检索）

52

11.3 动态分析(3)

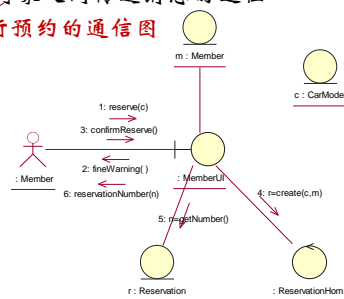
- 10.3.1 绘制用例的实现过程
- 10.3.2 边界、控制器和实体
- 10.3.3 通信图中的元素
- 10.3.4 给类添加操作
- 10.3.5 职责
- 10.3.6 状态建模

53

11.3.1 绘制用例的实现过程(1)

- 分析对象之间传递消息的过程

U7: 进行预约的通信图

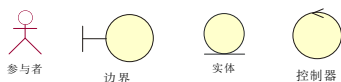


54

11.3.1 绘制用例的实现过程(2)

➤ 分析级的通信图可以显示

- 与系统边界交互的参与者(Member与MemberUI交互)
- 与系统内部的对象交互的边界(MemberUI与ReservationHome, Member, CarModel, Reservation交互)
- 系统内部的对象与外部系统的边界交互



55

11.3.2 边界、控制器和实体(1)

- **参与者**: 存在于系统外部的人或系统
- **边界**: 位于系统边缘上的对象, 在系统和参与者之间
 - 对于系统参与者, 边界提供了通信途径
 - 对于作为参与者的人, 边界表示用户界面, 以执行命令和查询, 显示反馈和结果
 - 每个边界对象通常都对应于一个用例或一组相关的用例, 通常映射到一个用户界面草案

56

11.3.2 边界、控制器和实体(2)

- **实体**: 系统内部的一个对象, 表示一个业务概念, 如顾客、汽车或型号
 - 一般实体由边界和控制器操作, 而不是自己的行为
 - 实体类出现在分析类图中, 大多数实体在设计过程结束后仍旧存在, 数据(属性)存放在数据库中

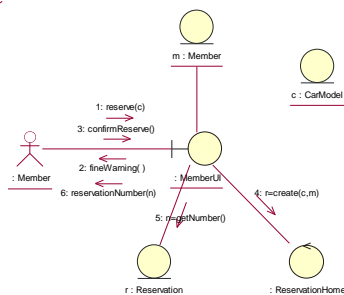
57

11.3.2 边界、控制器和实体(3)

- **控制器**: 封装了复杂或凌乱过程的系统内部对象
 - 控制器是一个服务对象, 提供:
 - 控制系统过程的全部或部分
 - 创建新实体
 - 检索已有实体
 - 控制器只是便于分析, 大多数控制器在设计过程结束后就不存在(转换成多个类中的方法)
 - Home是一个例外: Home用于创建新实体, 检索已有实体的控制器, 或包含实体信息

58

11.3.3 通信图中的元素(1)



59

11.3.3 通信图中的元素(2)

- 参与者的显示方式与用例图相同
- 对象显示为带标签的图标或带标签的方框
- 两个对象之间的线条表示链接, 与对象图相同
- 消息显示为顺序号、消息名称和参数列表
- 开放端的箭头显示消息发送的方向
- 标签拥有表示对象和参数
- 把返回值赋予一个名字可以显示为 n = getNumber()
- 条件消息可以显示为消息旁边的防护
- 迭代可以显示为顺序号旁边的*

60

11.3.4 给类添加操作(1)

➤ 识别操作

- ✦ 考虑系统责任
 - ✦ 有哪些功能要求在本对象提供?
- ✦ 考虑问题域
 - ✦ 对象在问题域中对应的事物有哪些行为?
- ✦ 分析对象状态
 - ✦ 在每种状态下对象可能发生什么行为? 对象状态的转换, 是由哪些操作引起的?

61

11.3.4 给类添加操作(2)

- ✦ 追踪操作的执行路线
 - ✦ 模拟操作的执行, 并在整个系统中跟踪
- ✦ 用动词识别操作
 - ✦ 查看每一个属性, 因为要用对象的操作来对其进行操纵
- 审查对象的每个操作
 - ✦ 是否真正有用
 - ✦ 直接提供系统责任所要求的某项功能?

62

11.3.4 给类添加操作(3)

- ✦ 响应其它操作的请求, 间接地完成某种功能的某些局部操作?
- ✦ 调整——取消无用的操作
- ✦ 是不是高内聚的
 - ✦ 一个操作应该只完成一项单一的、相对完整的功能
 - ✦ 调整——拆分或合并
- 操作的命名和定位
 - ✦ 命名: 动词或动宾结构

63

11.3.4 给类添加操作(4)

- ✦ 定位
 - ✦ 与实际事物一致
 - 例: 售货员—售货, 商品—售出
 - ✦ 在继承中的位置
 - 适合类的全部对象实例
- 描述操作
 - ✦ 把每个对象的操作都填写到相应的类符号中
 - ✦ 在类描述模板中, 写出:

64

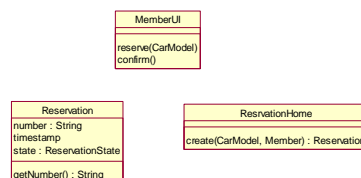
11.3.4 给类添加操作(5)

- ✦ 说明操作的职责
- ✦ 操作原型 (消息的格式)
- ✦ 消息发送 (指出在这个操作执行时, 需要请求哪些别的对象操作, 即接收消息的对象类名以及执行这个消息的操作名)
- ✦ 约束条件: 如果该操作的执行有前置条件、后置条件, 以及执行时间的要求等其它需要说明的事项, 则在这里加以说明
- ✦ 实现操作的方法 (文字, 活动图或流程图)

65

11.3.4 给类添加操作(6)

- 通信图上的每个消息都对应类上的一个操作, 从而得到用例实现的完整集合



66

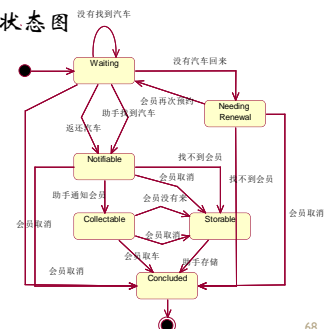
11.3.5 职责

- 职责有助于找出和正确指定操作及属性
- 只要发现系统需要的信息或行为，就应思考“哪个对象负责这个信息或行为”
- 确保每个对象不会负责多个任务（或角色）
- 把对象看作客户（提出命令，给出命令）或提供者（提供答案和服务）

67

11.3.6 状态建模

- 用于预订的状态图



68