

软件工程

第9章 需求建模 ——基于类的方法

徐本柱 软件学院

2018-09

主要内容

- ❖ 识别分析类
- ❖ 描述属性
- ❖ 定义操作
- ❖ 类-职责-协作者建模
- ❖ 关联和依赖
- ❖ 分析包

引言

- ❖ 基于类建模表示了：
 - 系统操作的**对象**
 - 应用于对象间能有效控制的**操作**（又称方法或服务）
 - 对象间（某种层级）的**关系**
 - 已定义类之间的**协作**
- ❖ 基于类的分析模型的元素包括
 - 类和对象、属性、操作、
 - **CRC**模型、协作图和包

9.1 识别分析类

- ❖ 检查需求模型开发的使用场景，对开发用例或处理叙述进行“语法解析”
- ❖ 开始类的识别
 - ❖ 名词或名词词组可以识别为类
 - ❖ 将这些名词输入到一个简单的表中（标出同义词）
- ❖ 如果要求某个类（名词）
 - ❖ 实现一个解决方案，这个类就是解决方案的一部分
 - ❖ 只需要说明一个解决方案，则是问题空间的一部分

分析类的表现

- ❖ **外部实体**（例如其他系统、设备、人员）：
 - ❖ 产生或使用基于计算机的系统的信息
- ❖ **事物**（例如报告、显示、字母、信号）：
 - ❖ 问题信息域的一部分
- ❖ **偶发事件/事件**（所有权转移/完成机器人的一组移动）：
 - ❖ 在系统操作环境内发生
- ❖ **角色**（例如经理、工程师、销售人员）：
 - ❖ 由和系统交互的人员扮演。
- ❖ **组织单元**（例如部门、组、团队）：
 - ❖ 和某个应用相关
- ❖ **场地**（例如制造车间或码头）：
 - ❖ 建立问题的环境和系统的整体功能
- ❖ **结构**（例如传感器、四轮交通工具、计算机）：
 - ❖ 定义了对象的类或与对象相关的类。

识别分析类

- ❖ [BUD96]建议了一种类的分类法，包括
 - ❖ 数据产生者（源）和数据使用者（汇点）、
 - ❖ 数据管理者、
 - ❖ 查看（或观察者）类
 - ❖ 以及辅助类。
- ❖ 什么不能是类或对象。
 - 通常不应该用“命令过程式的名称”为类命名
 - “面向对象的目的是封装，
 - 但仍保持独立的数据以及对数据的操作”

SafeHome实例[17]

SafeHome 安全功能允许房主在安装时配置安全系统，监控所有连接到安全系统的传感器，通过互联网、计算机或控制面板和房主交互信息。

在安装过程中，用 SafeHome 个人计算机来设计和配置系统。为每个传感器分配编号和类型，用主密码控制启动和关闭系统，而且当传感器事件发生时会拨打输入的电话号码。

识别出一个传感器事件时，软件激活装在系统上的发声警报，由房主在系统配置活动中指定的延迟时间结束后，软件拨打监控服务的电话号码并提供位置信息，报告检测到的事件性质。电话号码将每隔 20 秒重拨一次，直至电话接通。

房主通过控制面板、个人计算机或浏览器（统称为接口）来接收安全信息。接口在控制面板、计算机或浏览器窗口中显示提示信息和系统状态信息。房主采用如下形式进行交互活动……

SafeHome实例[18]

抽取这些名词，可以获得如下表所示的一些潜在类：

潜在类	一般分类	潜在类	一般分类
房主	角色或外部实体	主密码	事物
传感器	外部实体	电话号码	事物
控制面板	外部实体	传感器事件	事件
安装	事件	发声警报	外部实体
系统（别名安全系统）	事物	监控服务	组织单元或外部实体
编号，类型	不是对象，是传感器的属性		

识别分析类

- ❖ 6个选择特征，考虑每个潜在的类是否使用这些特征：
 - ❖ 1 保留信息：只有记录潜在类的信息才能保证系统正常工作，这样潜在类才能在分析过程中发挥作用
 - ❖ 2 所需服务：潜在类必须具有一种可确认的操作，这组操作能用某种方式改变类的属性值。
 - ❖ 3 多个属性：在需求分析过程中，焦点应在于“主”信息；事实上，只有一个属性的类可能在设计中有用，但是在分析活动阶段，把它作为另一个类的某个属性可能更好。
 - ❖ 4 公共属性：可以为潜在类定义一组属性，这些属性适用于类的所有实例。
 - ❖ 5 公共操作：可以为潜在类定义一组操作，这些操作适用于类的所有实例。
 - ❖ 6 必要需求：在问题空间中出现的外部实体，或者任何系统解决方案的运行所必需的信息，几乎都被定义为需求模型中的类

SafeHome实例[19]

潜在类	适用的特征编号
房主	拒绝：6 适用，但是 1、2 不符合
传感器	接受：所有都适用
控制面板	接受：所有都适用
安装	拒绝
系统（别名安全系统）	接受：所有都适用
编号，类型	拒绝：3 不符合，这是传感器的属性
主密码	拒绝：3 不符合
电话号码	拒绝：3 不符合
传感器事件	接受：所有都适用
发声警报	接受：2、3、4、5、6 适用
监控服务	拒绝：6 适用，但是 1、2 不符合

识别分析类

- ❖ 上面的表并不全面，必须添加其他类以使模型完整
- ❖ 某些被拒绝的潜在类将成为被接受的类的属性
- ❖ 问题的不同陈述可能导致作出不同的“接受或拒绝”决定

9.2 描述属性

- ❖ 属性描述了已经选择的、包含在需求模型中的类
- ❖ 实质上，属性是定义了类——明确知道什么
- ❖ 为了给分析类开发一个有意义的属性集合，
 - ❖ 软件工程师应该研究用例
 - ❖ 选择那些合情合理的“属于”类的“东西”
- ❖ 每个类都应回答如下问题：
 - ❖ 什么数据项在问题的环境内能够完整地定义这个类？

属性实例

❖ 为职业棒球手建立两种不同的类

● 数据统计系统:

- 属性是与名字、位置、平均击球次数、担任防守百分比, 从业年限、比赛次数等相关的

● 养老系统:

- 属性是平均工资、充分享受优惠权后的信用、所选的养老计划、邮件地址等相关的。

SafeHome实例[20]

- ❖ 考虑为SafeHome定义System类。
 - ❖ 房主可以配置安全功能以反映
 - ❖ 传感器信息、报警响应信息、激活或关闭信息、识别信息等
- ❖ 可以用如下方式表现这些组合数据项：

识别信息 = 系统编号+确认电话号码+系统状态

报警应答信息 = 延迟时间+电话号码

激活或者关闭信息 = 主密码+允许重复次数+临时密码

描述属性

- ❖ 等式右边的某些数据项可进一步精化到基础级
- ❖ 为System类组成一个合理的属性列表
- ❖ 传感器是整个SafeHome系统的一部分
 - ❖ 但是并没有列为图9-1中的数据项或属性
 - ❖ Sensor定义为类，多个Sensor对象将和System类关联
- ❖ 如果多个项和类相关联，避免这些项定义为属性

System类的类图

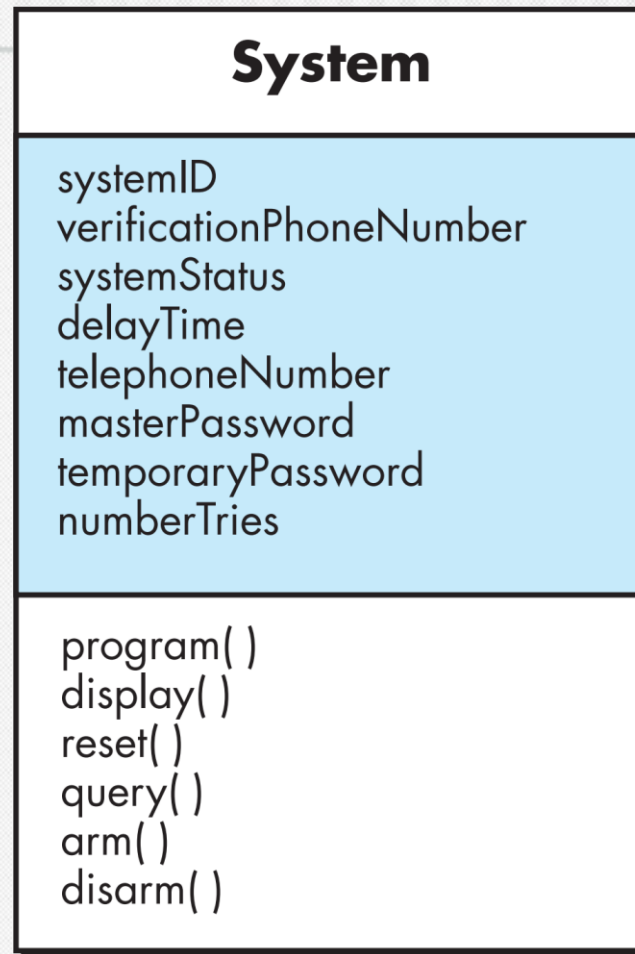


图9-1 System类的类图

9.3 定义操作

- ❖ 操作定义了某个对象的行为
- ❖ 通常可以粗略地划分为四种类型：
 - ❖ 1 以某种方式操作数据（如添加、删除等）；
 - ❖ 2 执行计算的操作；
 - ❖ 3 请求某个对象的状态的操作；
 - ❖ 4 监视某个对象发生某个控制事件的操作。
- ❖ 通过在属性和/或相关属性上的操作实现
- ❖ 操作必须“理解”类的属性和相关属性的本质

SafeHome实例[21]

在第一次迭代要导出一组分析类的操作时，可以再次研究处理说明（或用例）并合理地选择属于该类的操作。为了实现这个目标，可以再次研究语法解析并分离动词。这些动词中的一部分将是合法的操作并能够很容易地连接到某个特定类。例如，从本章前面提出的 SafeHome 处理说明中可以看到，“为传感器分配编号和类型”“主密码用于激活和解除系统”，这些短语表明：

- assign() 操作和 Sensor 类相关联。
- program() 操作应用于 System 类。
- arm() 和 disarm() 应用于 System 类。

再进一步研究，program() 操作很可能被划分为一些配置系统所需要的更具体的子操作。例如，program() 隐含着电话号码、配置系统特性（如创建传感器表、输入报警特征值）和输入密码。但是我们暂时把 program() 指定为一个单独的操作。

SafeHome实例[22]

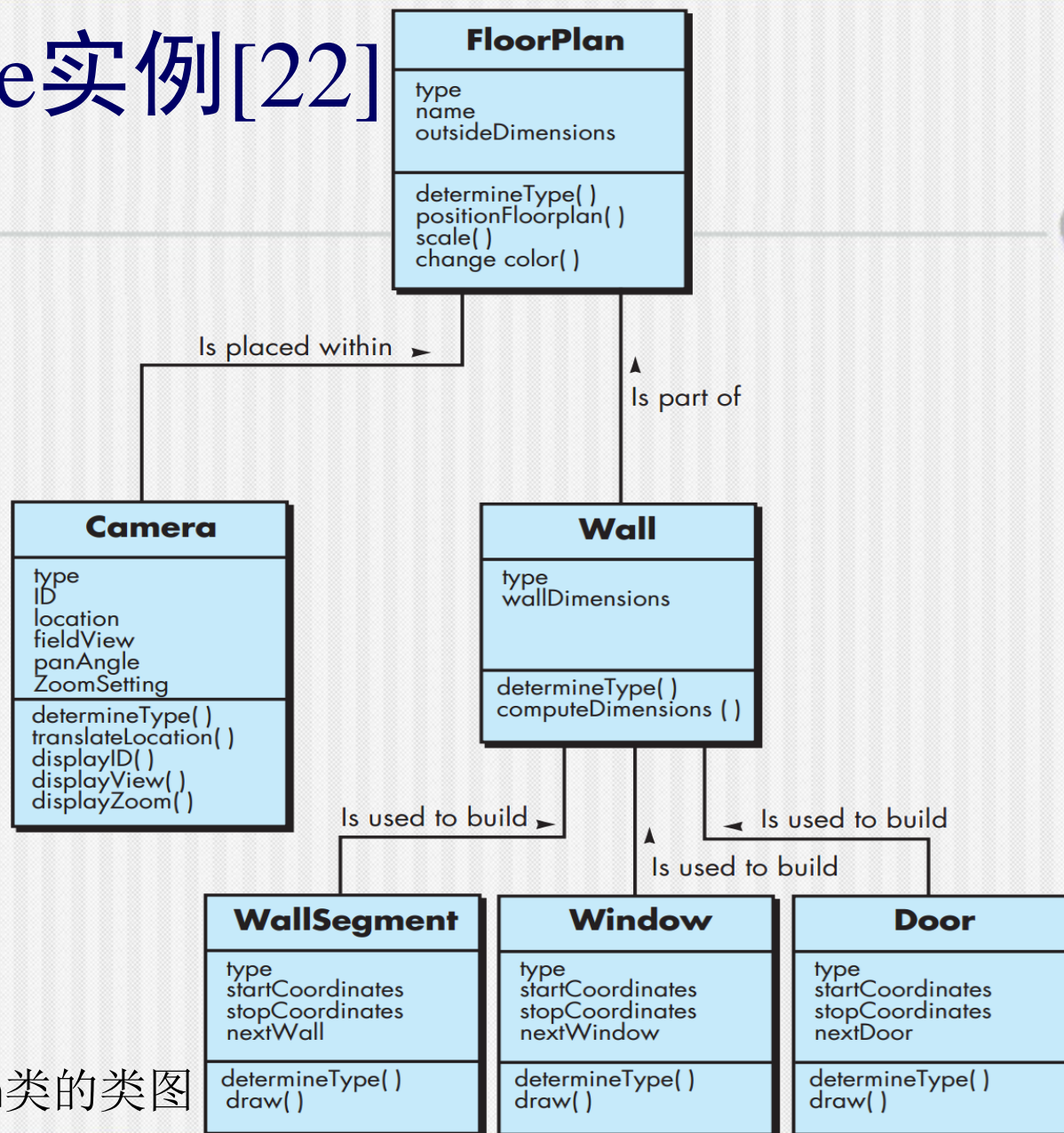


图9-2 FloorPlan类的类图

9.4 类-职责-协作者 (CRC) 建模

- ❖ CRC识别和组织与系统或产品需求相关的类。
- ❖ CRC模型实际上是表示类的标准索引卡片的集合
 - ❖ 卡片被分为三部分，顶部写类名
 - ❖ 下面左侧部分列出类的职责
 - ❖ 右侧部分列出类的协作者
- ❖ 职责是和类相关的属性和操作
 - ❖ 知道什么
 - ❖ 能做什么
- ❖ 协作者是那些提供完成某个职责所需要信息的类
 - ❖ 协作意味着信息请求或某个动作请求。

SafeHome实例[23]

Class: FloorPlan	
说明	
职责:	
定义住宅平面图名称/类型	协作者:
管理住宅平面图的布局	
缩放显示住宅平面图	
合并墙、门和窗	
显示摄像头的位置	Wall
	Camera

图9-3 CRC模型索引卡

类的分类

- ❖ **实体类**：也被称作**模型类**或**业务类**，从问题的说明中直接提取
 - ❖ 一般代表**保存**在数据库中的和**贯穿**应用程序的事物
- ❖ **边界类**：用于创建用户可见的和交互的**接口**（如交互屏幕或报表）
 - ❖ 实体类包含对用户来说很重要的信息，但是**并不显示**
 - ❖ 边界类的职责是**管理实体对象对用户的表示方式**
 - ❖ 如**CameraWindow**负责显示**SafeHome**系统的监视摄像头输出
- ❖ **控制类**：自始至终管理“工作单元”。即控制类可以管理
 - ❖ (1) 实体类的创建或更新；
 - ❖ (2) 当边界类从实体对象获取信息后的实例化；
 - ❖ (3) 对象集合间的复杂通信；
 - ❖ (4) 对象间交换的数据或用户和应用系统间交换的数据的确认。
- ❖ 通常直到**设计开始时**才开始考虑控制类。

职责

- ❖ 给类分配职责建议了以下五个指导原则：
 - ❖ 分布在所有类中以求最大程度满足解决问题的需求
 - ❖ 每个职责的说明应尽可能具有普遍性
 - ❖ 信息和与之相关的行为应在同一类中（封装、内聚）
 - ❖ 某个事物的信息应局限于一个而非分布在多个类中
 - ❖ 适合时，职责应由相关类共享。

协作

- ❖ 类有一种或两种方法实现其职责：
 - ❖ (1) 类可以使用其自身的操作控制各自的属性，从而实现特定的职责
 - ❖ (2) 和其他类协作。
- ❖ 协作定义了类之间的关系
 - ❖ 从客户职责实现角度表现从客户到服务器的请求
 - ❖ 是客户和服务端之间契约的具体实现

协作

❖ 识别类之间的关系。

- ◆ 一组类相互协作以实现某个需求，可以组织为子系统

❖ 识别协作

- ◆ 可以通过确认类本身是否能够实现自身的每个职责
- ◆ 如果不能实现每个职责，那么需要和其他类交互
- ◆ 因此就存在协作。

❖ 为识别协作者，检查类间三种不同的通用联系：

- ❖ (1) is-part-of(是.....一部分)联系；
- ❖ (2) has-knowledge-of(有.....的知识)联系；
- ❖ (3) depends-upon(依赖.....)联系。

三种不同的通用联系

- ❖ 属于某个聚合类一部分的所有类通过is-part-of联系和聚合类连接，如课本图9-4所示
- ❖ 当一个类必须从另一个类中获取信息时，就建立了has-knowledge-of关联。
- ❖ depends-upon关联意味着两个类之间具有has-knowledge-of和is-part-of不能实现的依赖关系。
- ❖ 协作类的名称记录在CRC模型索引卡右边
- ❖ 索引卡包含**职责列表**、实现这些职责的协作

聚合类

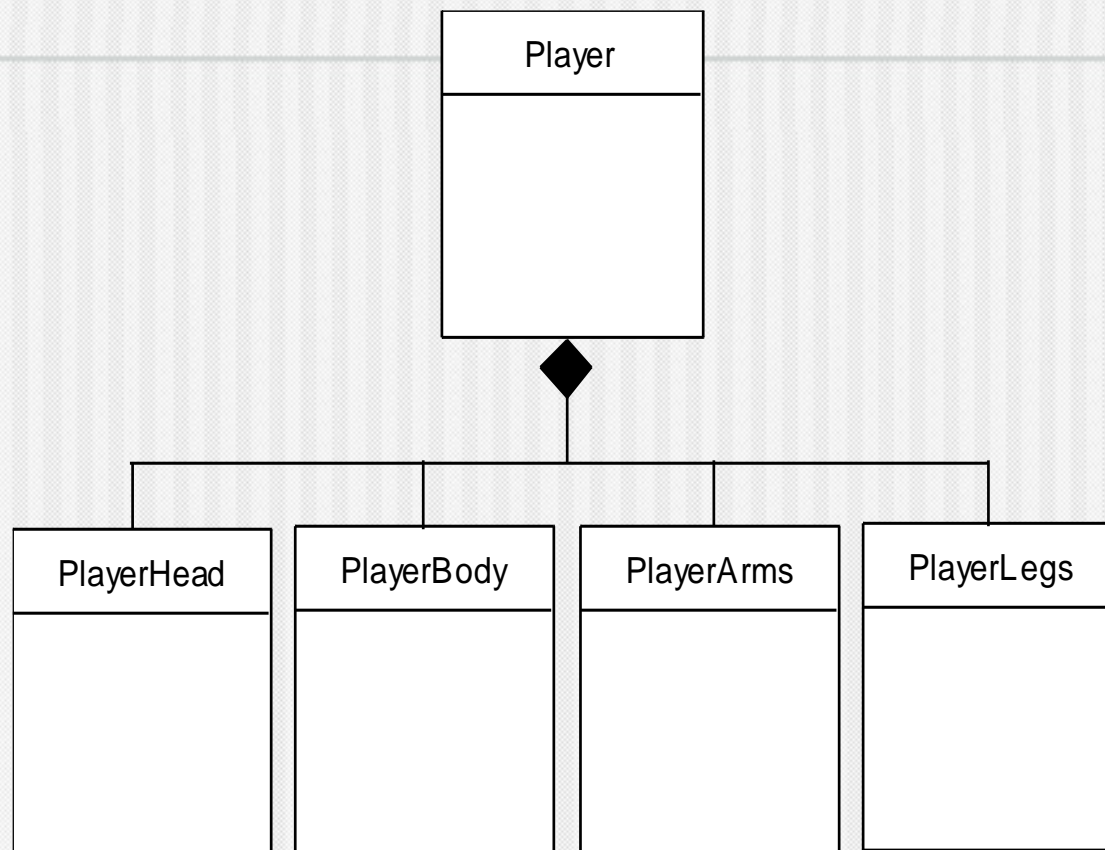


图9-4 复合聚合类

CRC模型评审

- ♥ 所有参加CRC模型评审的人员拿到一部分CRC模型索引卡
 - ❖ 拆分协作卡片（也就是说每个评审员不得有两张存在协作关系的卡片）。
- ♥ 分类管理所有的用例场景（以及相关的用例图）。
- ♥ 评审组长细致地阅读用例。
 - ❖ 当评审组长看到一个已命名的对象时，
 - ❖ 给拥有相应类索引卡的人员一个令牌。
- ♥ 当令牌传递时，该类卡的拥有者需要描述卡上记录的职责。
 - ❖ 评审组确定（一个或多个）职责是否满足用例需求。
- ♥ 如果记录在索引卡上的职责和协作不能满足用例，就需要修改卡片
 - ❖ 修改可能包括定义新类（和相关的CRC索引卡），
 - ❖ 或者在已有的卡上说明新的或修改的职责、协作。

9.5 关联和依赖

- ❖ 很多例子中，两个分析类以某种方式相互联系着
 - 在UML中，这些联系被称作**关联**(associations)
 - **关联**定义了类之间的联系，**多样性**定义了一个类和另一个类之间的联系数量关系
- ❖ 在某些情况下，关联可以更进一步地指出多样性
 - ❖ 多样性限制实例如图9-5所示，
 - ❖ 其中“一个或多个”使用1..*表示，
 - ❖ “0个或多个”使用0..*表示。
 - ❖ 在UML中，星号表示范围无上界。

多样性

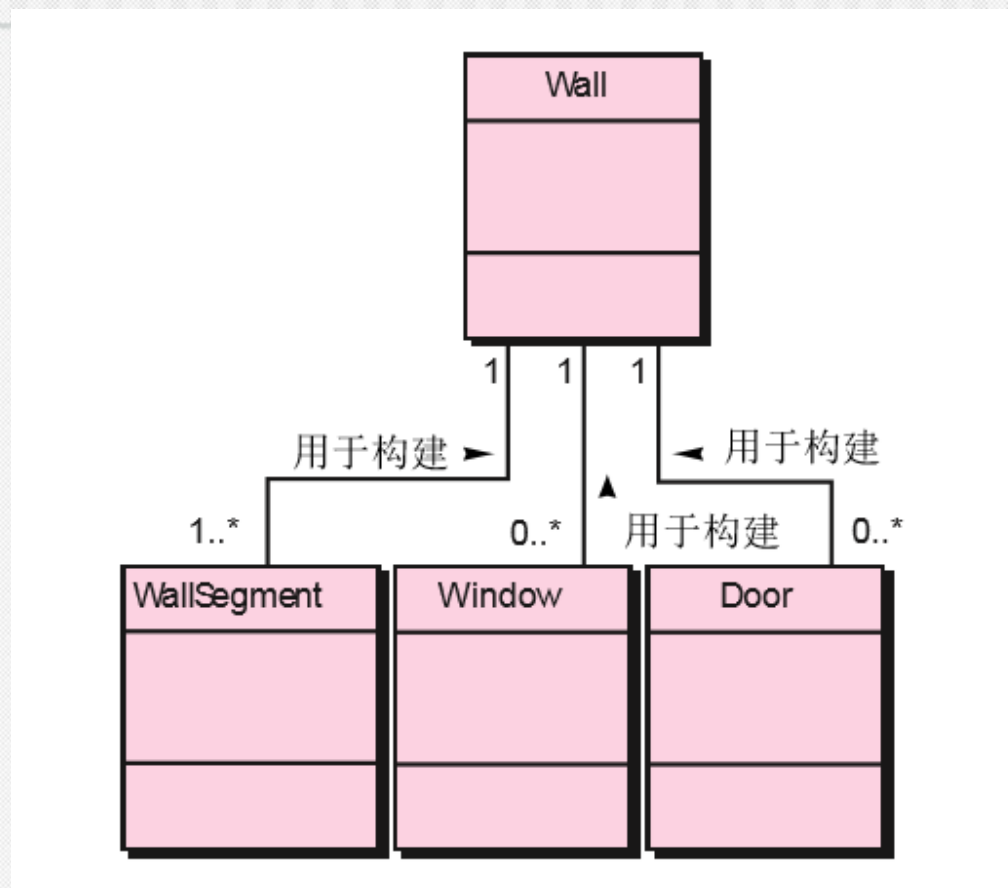


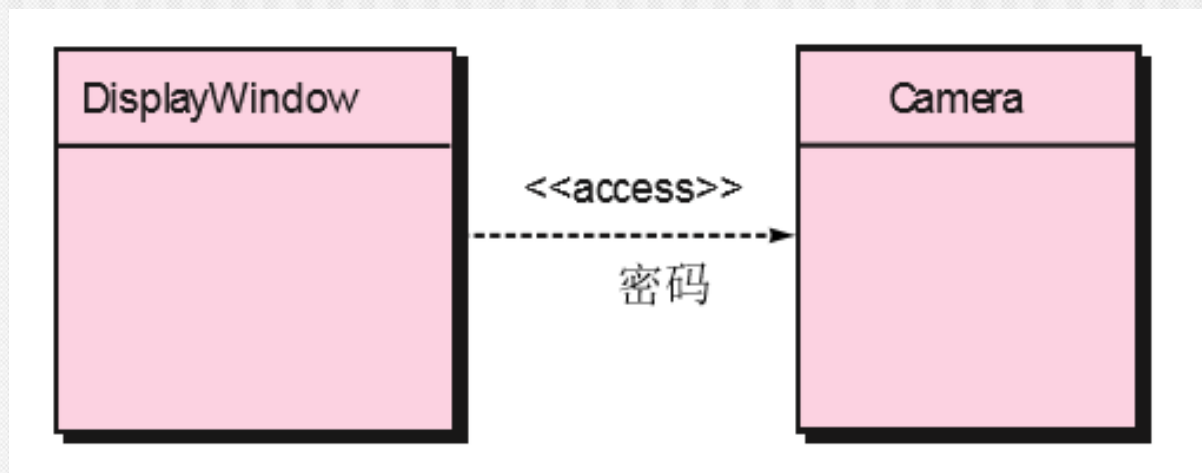
图9-5 多样性

关联和依赖

- ❖ 在很多事例中，两个分析类之间存在客户-服务器关系
 - 这种情况下，客户类以某种方式依赖于服务器类
 - 并且建立了 *依赖关系*
- ❖ 依赖由一个构造型定义
 - 在UML中，构造型是一个“可扩展机制”，
 - 允许软件工程师定义特殊的建模元素，
 - 这些建模元素的语义是由用户自定义的。
 - 在UML中，构造型由双尖括号表示。

SafeHome实例[24]

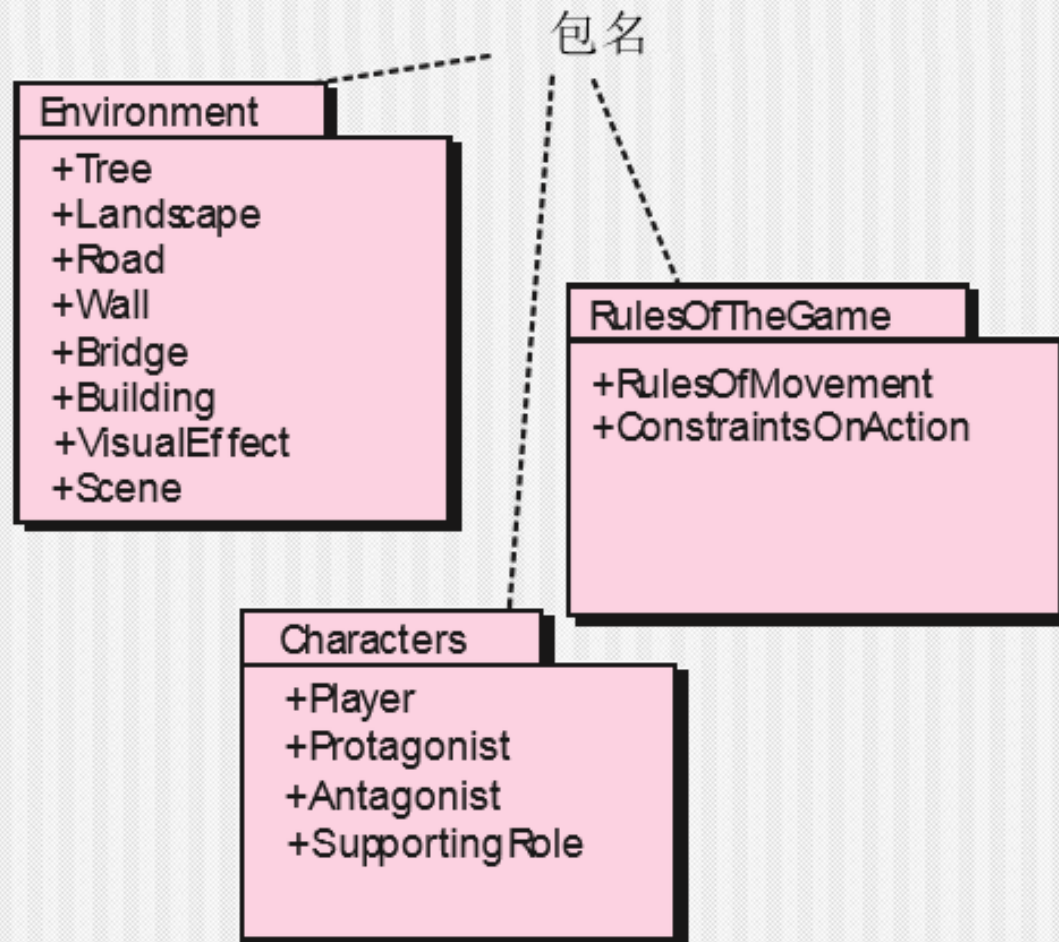
下面举例说明SafeHome监视系统中的简单依赖。一个**Camera**对象（本例中的服务器类）向一个**DisplayWindow**对象（本例中的客户类）提供视频图像。这两个对象之间的联系不是简单的联系，而是存在依赖联系。在监视用例（没有列出来）中，建模者知道必须提供特殊的密码才能查看指定摄像头的位置。为实现这一点，一种方法是让**Camera**请求密码，然后在获得**DisplayWindow**的允许后显示视频。这可以由图8-18表示，其中《access》意味着通过密码控制使用摄像头的输出。



9.6 分析包

- ❖ 分析建模的一个重要部分是分类，也就是将分析模型的各种元素（如用例，分析类）分组打包，并为其取一个有代表性的名称。
- ❖ 每个包中分析类名字前的加号表示该类是公共可见的，因此可以从其他包访问
- ❖ 但包中的任何元素之前可以添加其他符号。负号表示该元素对其他包是隐藏的，#号只能由指定包中的类访问表示该元素

分析包





谢谢!