



CS2001

软件工程

15. 软件需求

— 软件需求分析与建模

课程回顾问题

对于一个大学选课系统而言，以下哪些需求属于质量需求（多选）？

- 1) 可根据关键字检索感兴趣的课程
- 2) 输入关键字后1秒内返回匹配的课程列表
- 3) 界面具有用户友好性
- 4) 选课系统应当在当前校园网基础上提供服务
- 5) 课程选课人数到达上限后不再允许选课
- 6) 系统宕机时间尽可能少

课程回顾问题

对于一个大学选课系统而言，以下哪些需求属于质量需求（多选）？

- 1) 可根据关键字检索感兴趣的课程
- 2) 输入关键字后1秒内返回匹配的课程列表
- 3) 界面具有用户友好性
- 4) 选课系统应当在当前校园网基础上提供服务
- 5) 课程选课人数到达上限后不再允许选课
- 6) 系统宕机时间尽可能少

需求抽取与分析

• 需求抽取：从各种需求来源那里获取需求信息

- ✓ 利益相关者：通过访谈、问卷、观察等手段
- ✓ 文档资料：查看法律、法规、市场分析报告等相关资料
- ✓ 现有系统：分析遗留系统、竞争对手系统等软件系统

• 需求分析

- ✓ 从原始需求信息中发现有用的信息并进行整理和组织
- ✓ 对于模糊的需求进行澄清或细化
- ✓ 发现潜在的冲突并进行权衡决策，涉及利益相关者冲突的需要进行协商
- ✓ 对需求优先级进行分档或排序

需求抽取、分析与反馈等交织迭代

需求交流与获取方式

- 访谈
- 研讨会
- 观察
- 调查问卷

《软件工程》 4. 3. 1

《构建之法》 8. 3

需求规格说明

- 自然语言句子：使用带编号的句子来书写，每个句子表达一条需求
- 结构化自然语言：基于标准的自然语言表格或模板书写，每个字段提供一个方面的需求信息
- 图形化表示法：辅以文本注释的图形化模型，如UML (unified modeling language, 统一建模语言)
- 使用数学符号的形式化描述：基于有限状态机或集合等数学概念进行需求描述

自然语言需求规格说明的优缺点

• 优点

- ✓ 普适：可以用于任一问题域和领域
- ✓ 灵活：允许在需求描述中进行任意的抽象和精化
- ✓ 易理解：无需训练和特殊的工具，对于大多数人都较容易理解

• 缺点

- ✓ 二义性（歧义）：存在多种合理的解释，不同的人对同一个需求可能有不同的理解
- ✓ 模糊性：容易忽略必要的细节
- ✓ 不规范：格式、陈述方式等都很随意

自然语言需求描述的二义性

- 模糊的描述带来的二义性

- ✓ 例：高年级学生选课时提醒核对毕业要求
- ✓ 例：如用户访问记录存在异常则拒绝登录

- 自然语言固有的二义性

- ✓ 多义词带来的二义性
- ✓ 语法二义性，如：导航系统必须显示最后五个目的地和起点
- ✓ 指代二义性，如：用户将身份识别卡插入读卡器后在键盘上输入个人识别码（PIN），如果无效则系统拒绝访问

使用自然语言需求描述的建议

- 使用规范化的文档模板
- 定义并严格遵循文档术语表
- 规范需求的组织结构并保持基本需求条目的原子性
- 文字表达尽量避免二义性
- 结合表格、UML模型等其他更精确的描述方式

需求描述的原子性示例：大学选课系统

R：可根据关键字检索感兴趣的课程，输入关键字后1秒内返回匹配的课程列表



R1：可根据关键字检索感兴趣的课程（功能）

R2：输入关键字后1秒内返回匹配的课程列表（质量）

刻画对于系统的基本要求、不可再分
不同类型的要求（如功能、质量）相分离
可独立进行测试验证

常用的需求分析及建模视角

- 场景分析与建模
- 类分析与建模
- 数据流分析与建模
- 行为分析与建模

用况 (Use Case) 与场景 (Scenario)

- 描述外部参与者与目标系统之间的交互过程
 - ✓ 参与者：与系统交互的外部用户、硬件或第三方系统
 - ✓ 用况：参与者与系统之间的典型交互，包含一系列交互步骤
- 例如：大学选课系统
 - ✓ 参与者：学生、教师、教务管理人员
 - ✓ 用况：选课、查询成绩、下载选课名单、排课
- 优势：从外部参与者的视角直观刻画与系统的交互过程

参与者 (Actor)

- 参与者代表了一种角色（如银行客户、学生），一个用户可以扮演多种角色
- 参与者的类型
 - ✓ 人（用户）：使用各种交互设备与系统进行物理交互
 - ✓ 外部设备：通过传感器等与系统进行交互的外部设备
 - ✓ 外部系统：通过API接口等与目标系统进行交互的其他第三方软件系统
 - ✓ 定时器：周期性地向系统发出时间事件的虚拟参与者
- 主次参与者
 - ✓ 主要参与者：发起用况并主导交互过程
 - ✓ 次要参与者：参与用况交互但非主导地位

用况与场景的关系

- 用况执行过程中会有不同情况发生
 - ✓ 系统执行过程中根据运行状况在某些步骤上作出不同的选择，如网上购物用况中系统根据用户的不同等级执行不同的优惠策略
 - ✓ 用户在某些步骤上作出不同的选择，如网上购物用况中在支付环节选择不同的支付方式
 - ✓ 异常情况的发生导致相应的处理，如网络故障等
- 场景可以视为用况的实例，代表用况的不同执行路径
 - ✓ 正常执行场景
 - ✓ 异常处理场景

用况和场景模型的UML描述

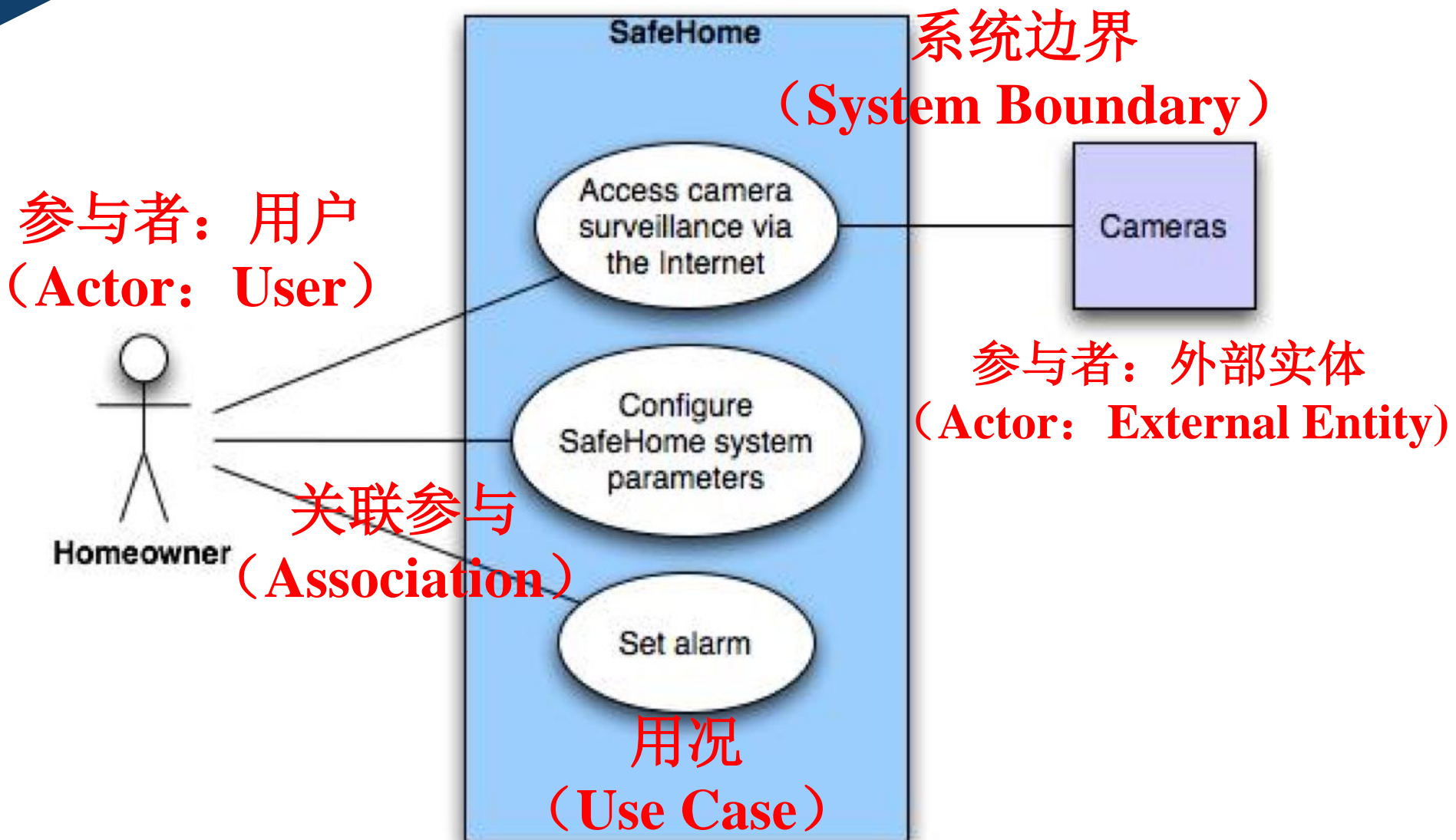
• 用况概览

- ✓ 描述参与者与所参与的相关用况之间的关系
- ✓ 使用UML用况图 (Use Case Diagram)

• 用况交互过程

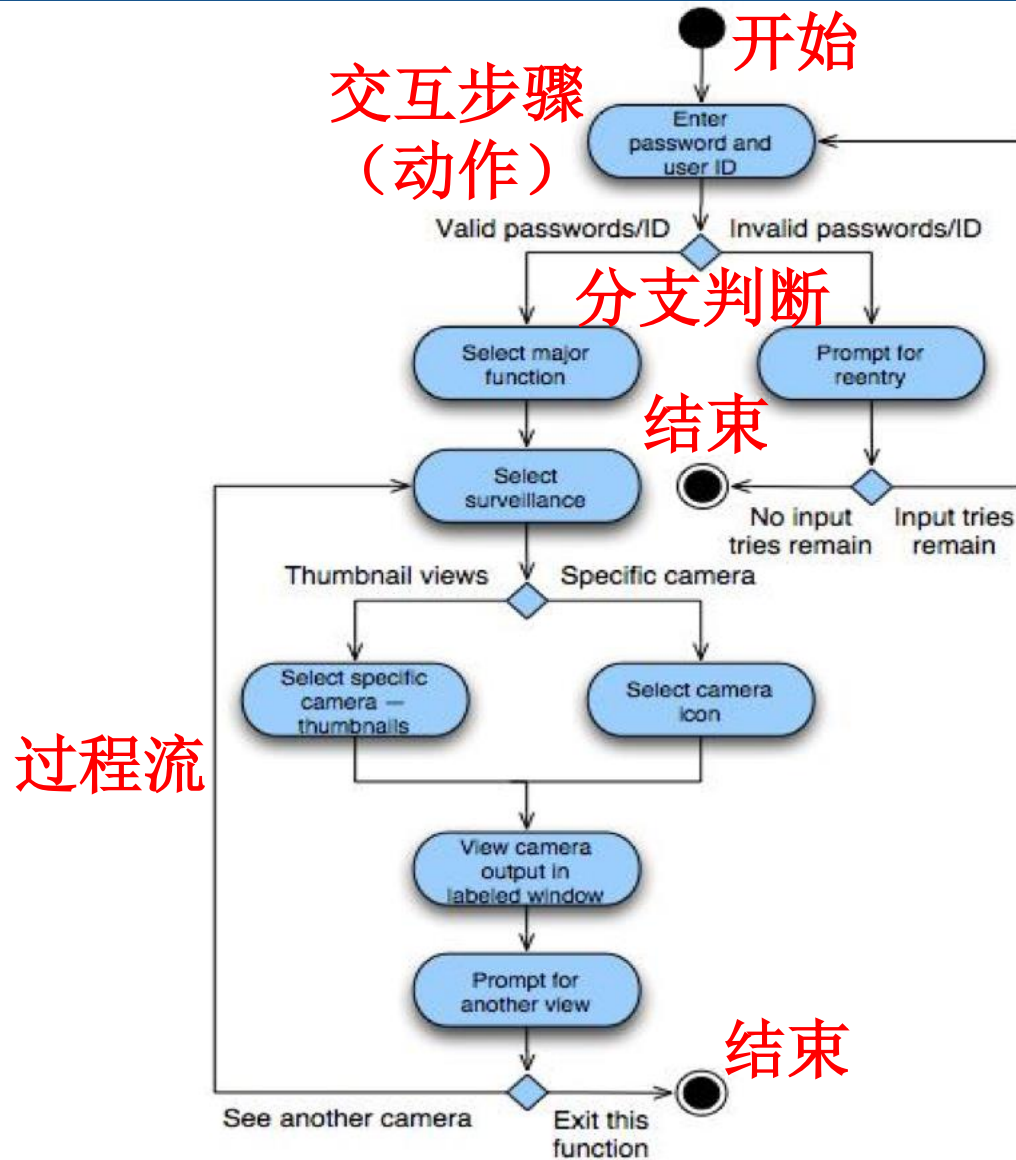
- ✓ 描述用况的具体交互步骤及其过程流
- ✓ 使用UML活动图 (Activity Diagram) / 泳道图 (Swim Lane Diagram)

UML用况图：用况概览



智能家居监控系统用况图

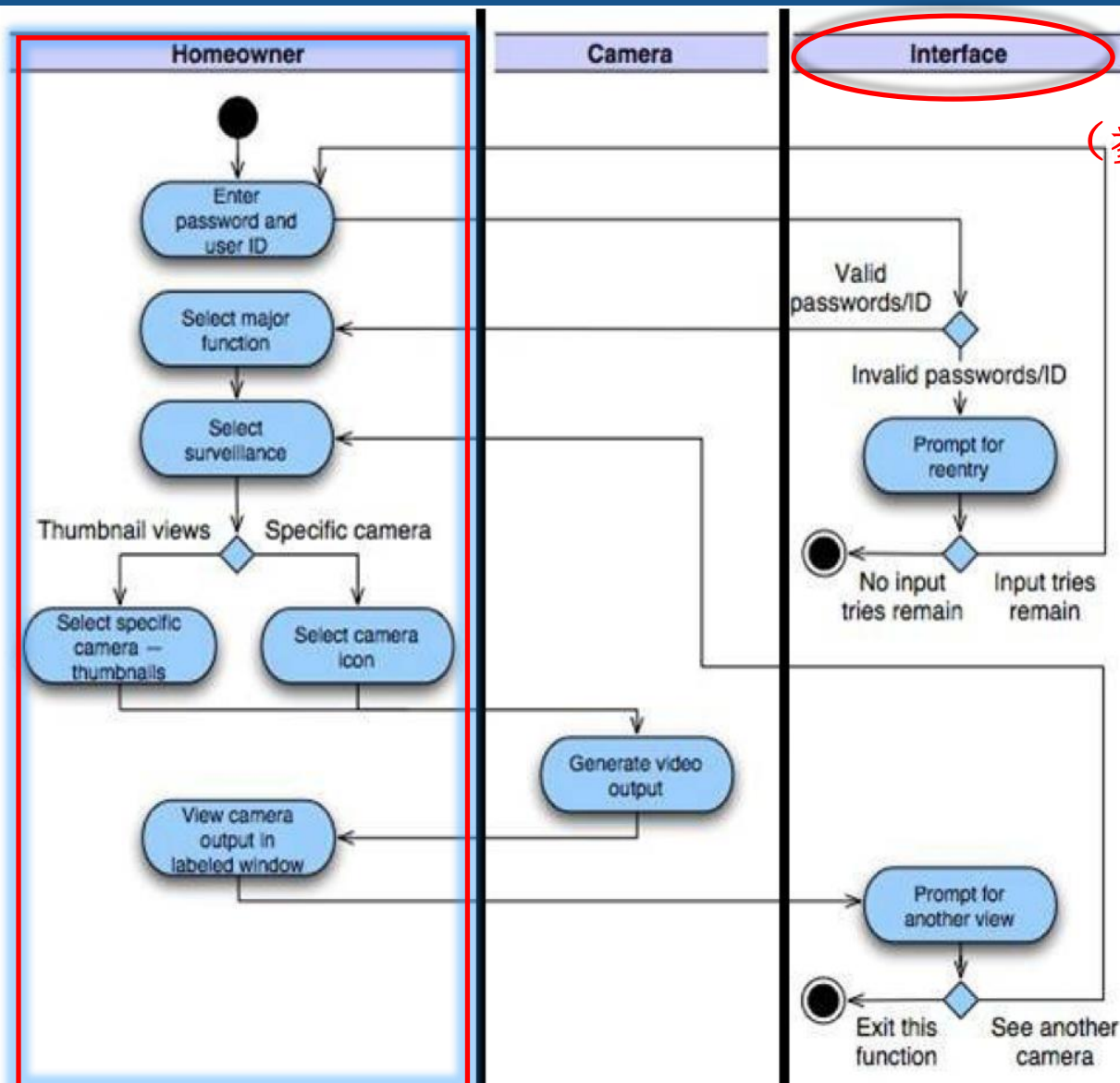
UML活动图：用况交互过程



用况：Access Camera Surveillance via the Internet

UML泳道图：突出用况交互参与方

泳道



用况：Access Camera Surveillance via the Internet

用况与场景的关系

- 用况执行过程中会有不同情况发生
 - ✓ 系统执行过程中根据运行状况在某些步骤上作出不同的选择，如网上购物用况中系统根据用户的不同等级执行不同的优惠策略
 - ✓ 用户在某些步骤上作出不同的选择，如网上购物用况中在支付环节选择不同的支付方式
 - ✓ 异常情况的发生导致相应的处理，如网络故障等
- 场景可以视为用况的实例，代表用况的不同执行路径
 - ✓ 正常执行场景
 - ✓ 异常处理场景

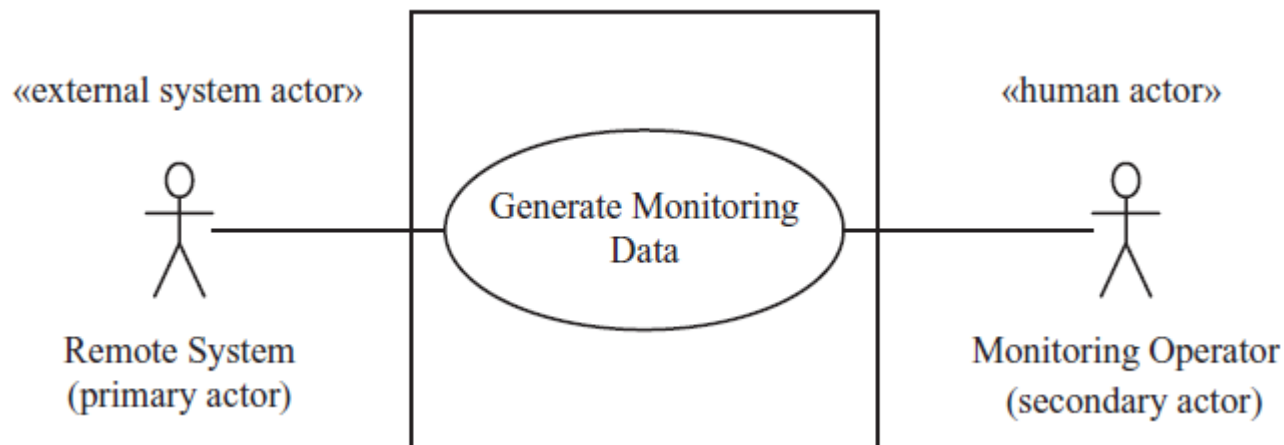
基本文本的用况描述（模板）

- 用况名称
- 主要参与者
- 目标（参与者在此用况中的目标）
- 前置条件（满足此条件用况才能发生）
- 触发条件（触发用况执行的条件）
- 主场景（一系列交互步骤）
- 其他场景（针对其他情况的交互步骤）
- 异常场景（针对异常情况的交互步骤）
- 发生频率（用况发生的频率）
- ...

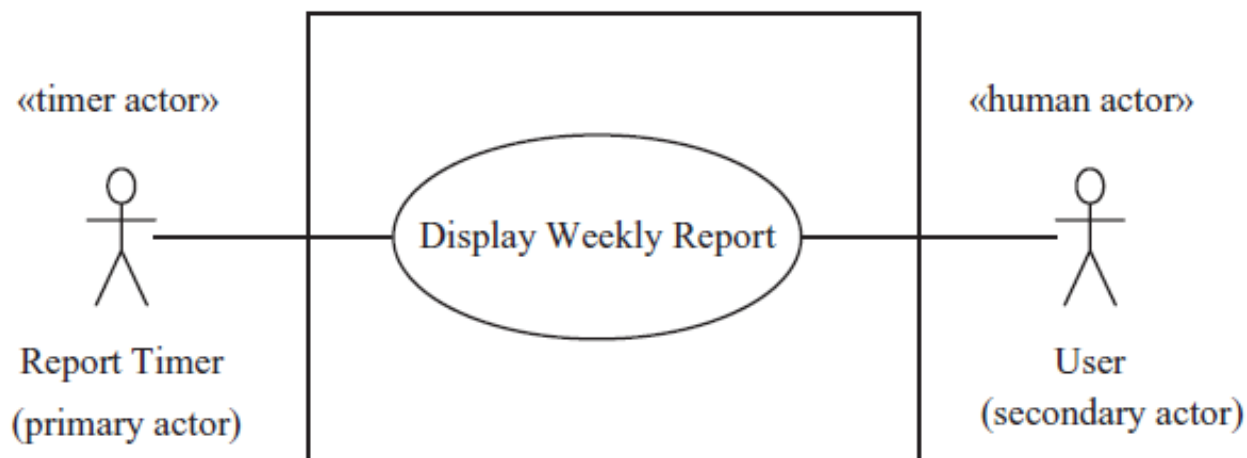
如何识别用况

- 识别典型的参与者以及他们与系统的典型交互方式
- 避免将用况变成功能分解结构
 - ✓ 例如，为表示功能分解结构，将“账户管理”作为父用况，而“账户充值”、“账户余额查询”等作为子用况
 - ✓ “账户管理”是一个功能分类并非一个交互过程
- 用况应当具备
 - ✓ 完整性：该用况有明确的开始和结束
 - ✓ 独立性：该用况可以不依赖其他用况独立完成
 - ✓ 价值性：完成该用况对于相关用户有一定的意义

用例示例



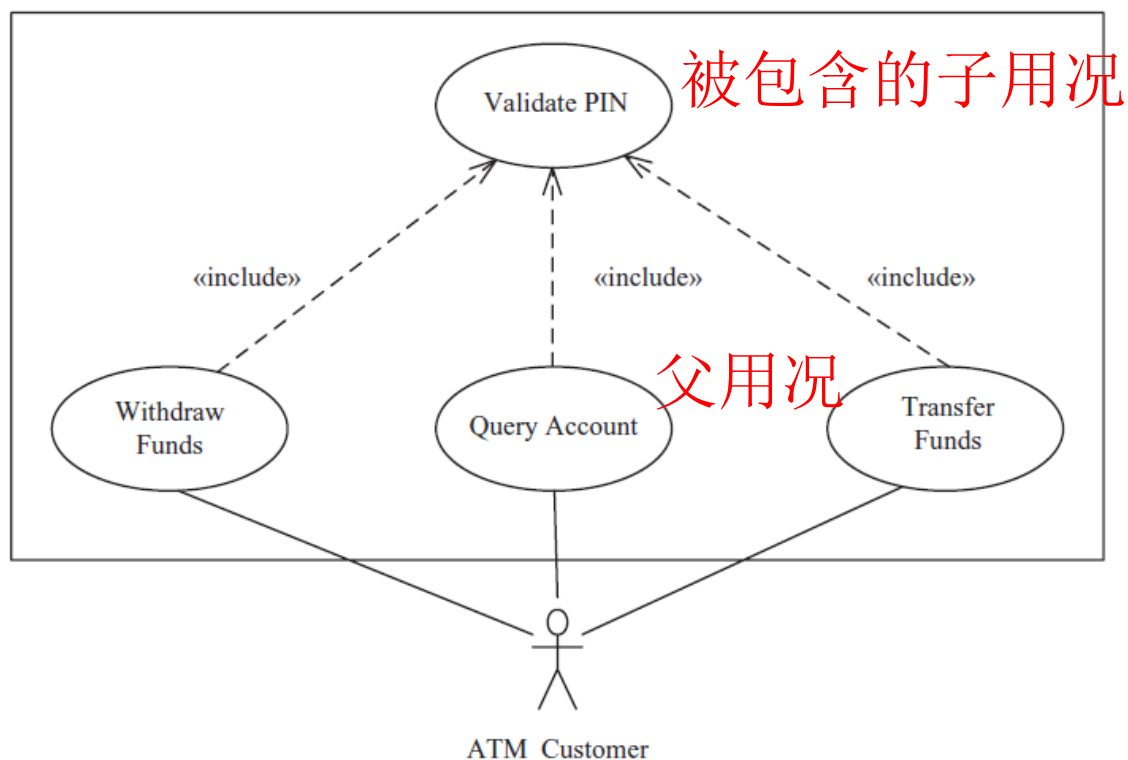
用例中的主要和次要参与者



用例中的定时器参与者

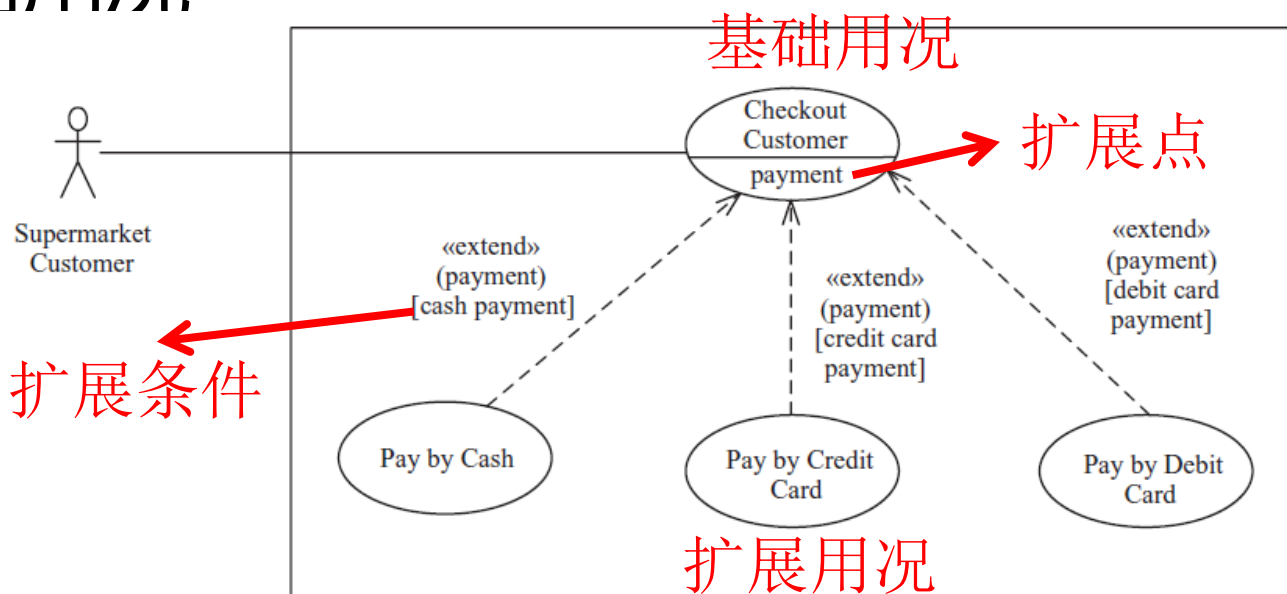
用况的包含 (include)

- 多个用况包含共同的交互序列片段
- 将共同的部分抽取出来作为一个子用况，并建立与所属用况的包含关系

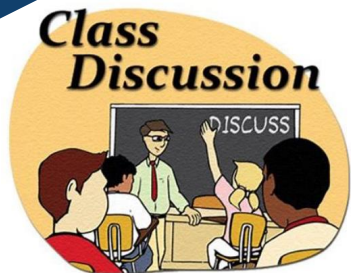


用况的扩展 (extend)

- 在一个基础用况上扩展在交互序列上略有差别的扩展用况
- 基础用况提供扩展点，在扩展点上根据不同的条件扩展出不同的新用况
- 基础用况本身可以完整执行，扩展用况依赖于基础用况



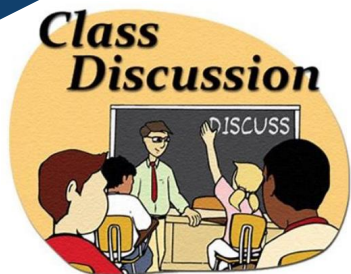
课堂讨论：合理的用况



课堂讨论：下列这些网上购物系统用况是否合适？

- 1) 输入密码
- 2) 账户管理（包括修改密码、查看登录历史记录等）
- 3) 发送促销邮件（按照一定的规则定时批量发送）
- 4) 查询订单物流状态
- 5) 选购商品下订单以及后台离线订单审核（即后台工作人员在订单提交后某个时间进行订单审核）
- 6) 选购商品下订单以及后台实时订单审核（即后台工作人员在订单预备提交时在线审核，通过后才允许提交）

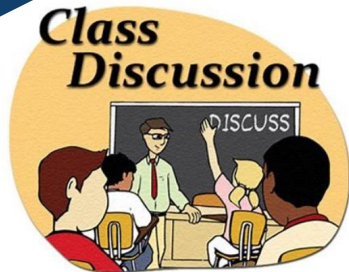
课堂讨论：合理的用况



课堂讨论：下列这些网上购物系统用况是否合适？

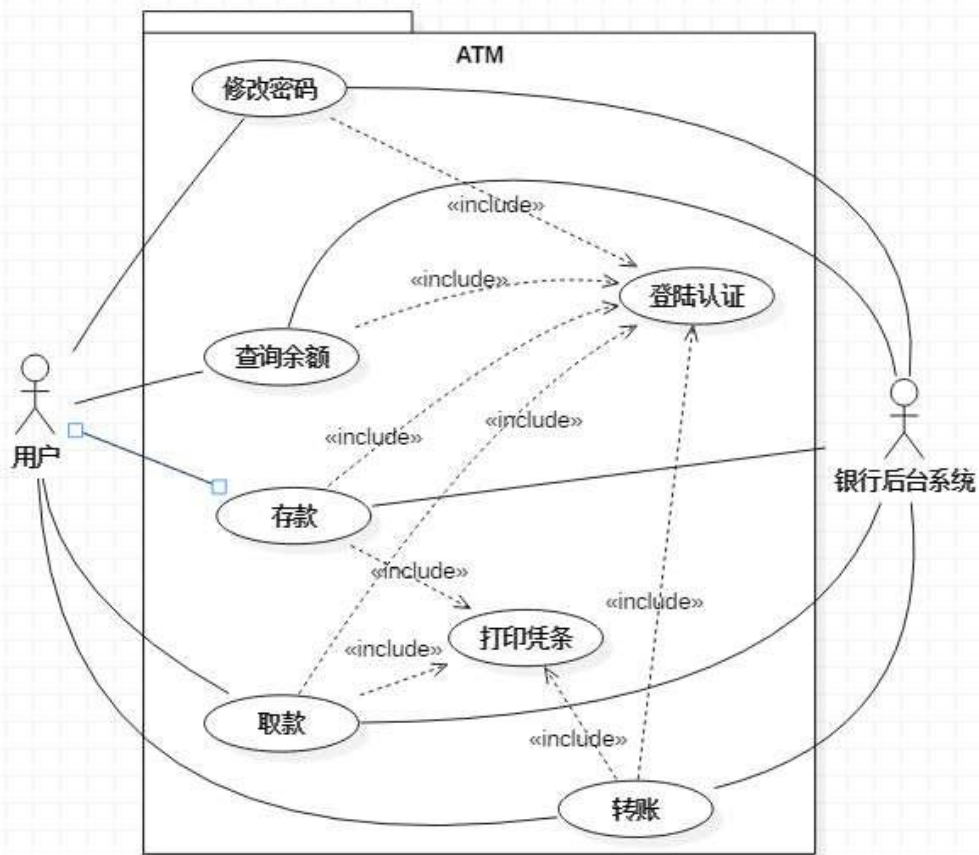
- 1) 输入密码
- 2) 账户管理（包括修改密码、查看登录历史记录等）
- 3) 发送促销邮件（按照一定的规则定时批量发送）
- 4) 查询订单物流状态
- 5) 选购商品下订单以及后台离线订单审核（即后台工作人员在订单提交后某个时间进行订单审核）
- 6) 选购商品下订单以及后台实时订单审核（即后台工作人员在订单预备提交时在线审核，通过后才允许提交）

课堂讨论：用况建模

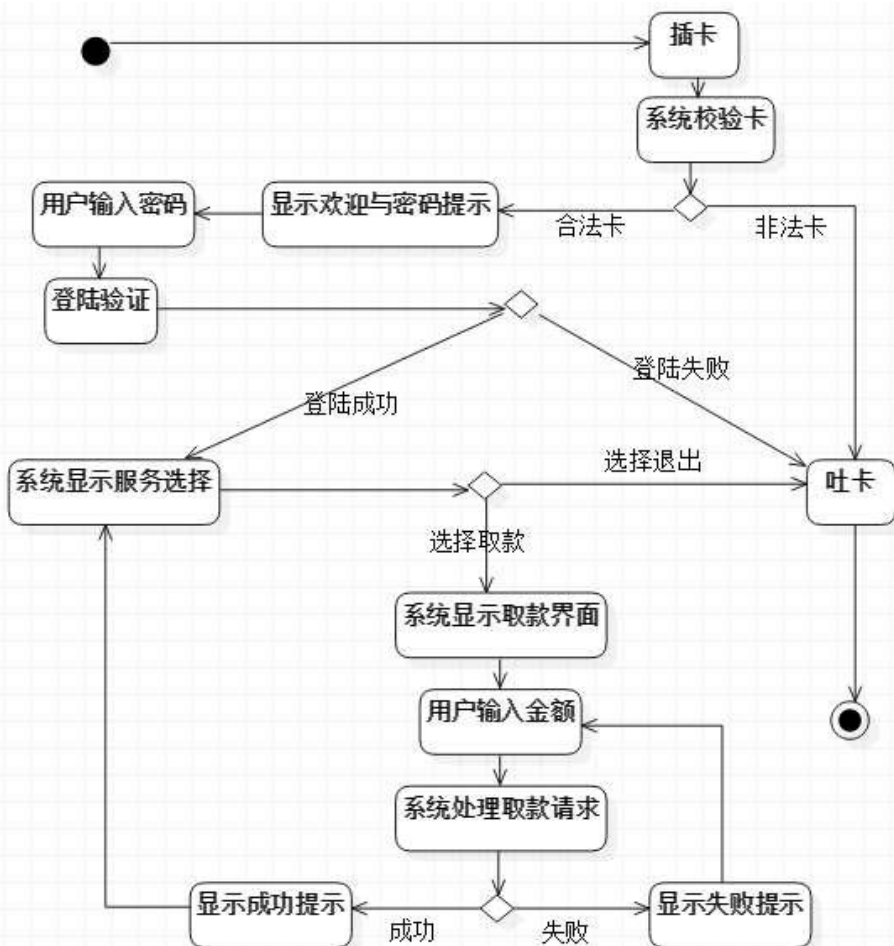


课堂讨论：请根据自身使用经验针对ATM机系统进行用况分析和建模。假设该系统后台连接已有的银行系统为用户提供服务，并且不考虑银行工作人员的维护任务。

AMT机系统用况图和活动图



用况图



“取款”用况的活动图

类分析与建模

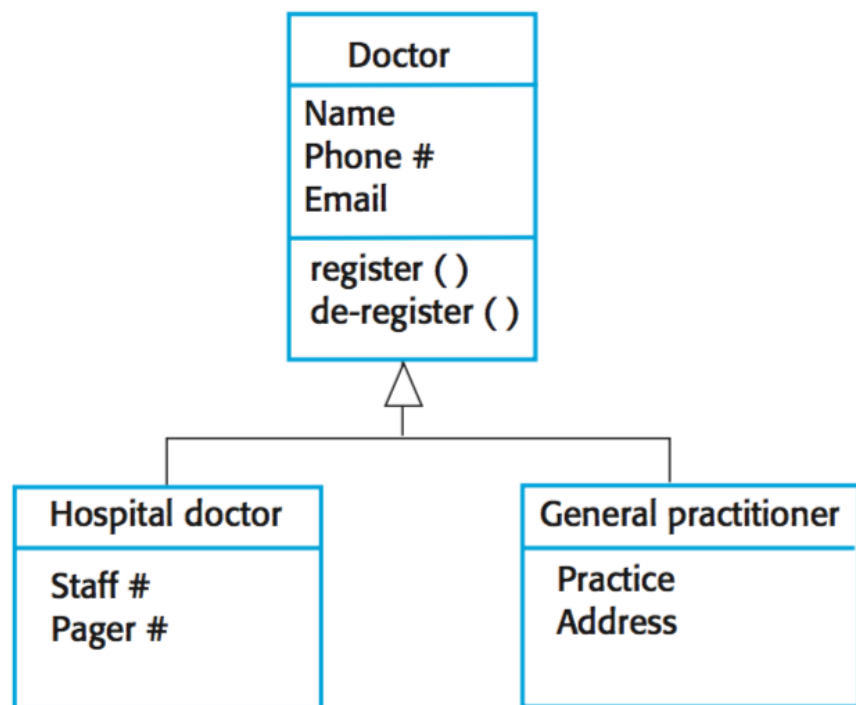
- 基于面向对象的思想，通过分析类及其关系对问题领域进行建模
- 可使用UML类图来描述
 - ✓ 分析类：属性、操作
 - ✓ 类间关系：继承（Inheritance）、聚合（Aggregation）、关联（Association）、依赖（Dependency）
- 基本过程
 - ✓ 识别分析类
 - ✓ 确定分析类之间的职责（属性、操作）分配及协作关系（类间关系）

UML类模型

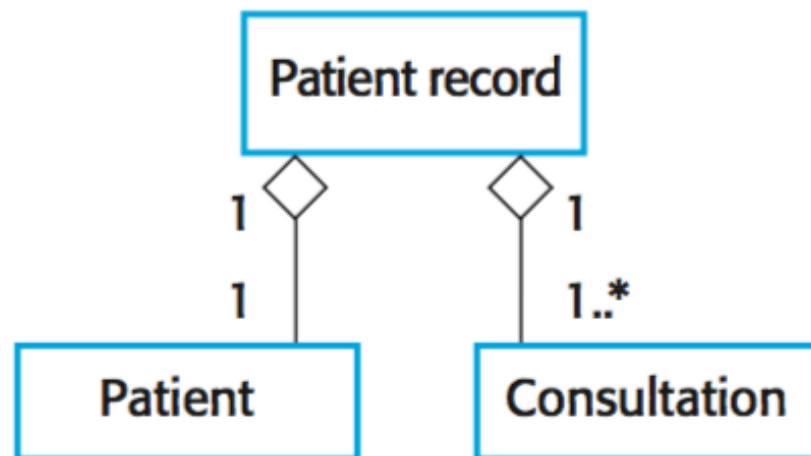
- 类：类名、属性、方法
- 类间关系
 - ✓ 关联 (Association)
 - ✓ 依赖 (Dependency)
 - ✓ 整体/部分：组合 (Composition)
聚合 (Aggregation)
 - ✓ 泛化/特殊：继承 (Inheritance)
 - ✓ 实现 (Implementation)

类间关系的强度：Generalization = Implementation > Composition > Aggregation > Association > Dependency

需求建模中常用的类关系



泛化关系



聚集关系



关联关系

识别分析类

- 从场景和功能等文本需求描述中识别名词和名词短语作为候选的分析类
- 常见的分析类类型
 - ✓ 产生或消费信息的**物理实体**，如摄像头、打印机等
 - ✓ 作为目标系统信息域一部分的**信息实体**，如订单、课程等
 - ✓ 在系统运行上下文中发生的**事件**，如来电、报警等
 - ✓ 由人扮演、与系统进行交互的**角色**，如学生、管理员等
 - ✓ 与系统相关的**组织单元**，如学院、企业等
 - ✓ 与系统相关的**地点**，如教室、房间等

分析类选取准则

- 包含信息：包含系统运行所需要的信息
- 包含操作：包含一组能够改变自身信息的操作
- 多个属性：包含多个属性而非单一属性
- 通用属性/操作：所定义的属性/操作是该类所有实例所通用的
- 重要的外部实体：产生或消费重要信息
(对于系统运行起重要作用) 的外部实体

非绝对标准，多条准则之间可能冲突

课堂讨论：分析类的选取

Class Discussion



课堂讨论：下列这些安全家居系统中的事物作为分析类是否合适？该系统通过摄像头等监控设备和传感器监控家庭中的各种情况并进行报警。系统针对各个家庭进行安装和管理，每个家庭共享一个登录密码。

- 1) 客户家庭：安装该系统的客户家庭
- 2) 房屋主人：客户家庭的主人
- 3) 系统密码：客户家庭登录系统的密码
- 4) 联系电话：客户家庭的联系电话
- 5) 摄像头：安装在家庭中的摄像头
- 6) 系统安装：系统的安装过程
- 7) 报警事件：系统的一次报警

选取准则：
包含信息
包含操作
多个属性
通用属性/操作
重要的外部实体

课堂讨论：分析类的选取

Class Discussion



课堂讨论：下列这些安全家居系统中的事物作为分析类是否合适？该系统通过摄像头等监控设备和传感器监控家庭中的各种情况并进行报警。系统针对各个家庭进行安装和管理，每个家庭共享一个登录密码。

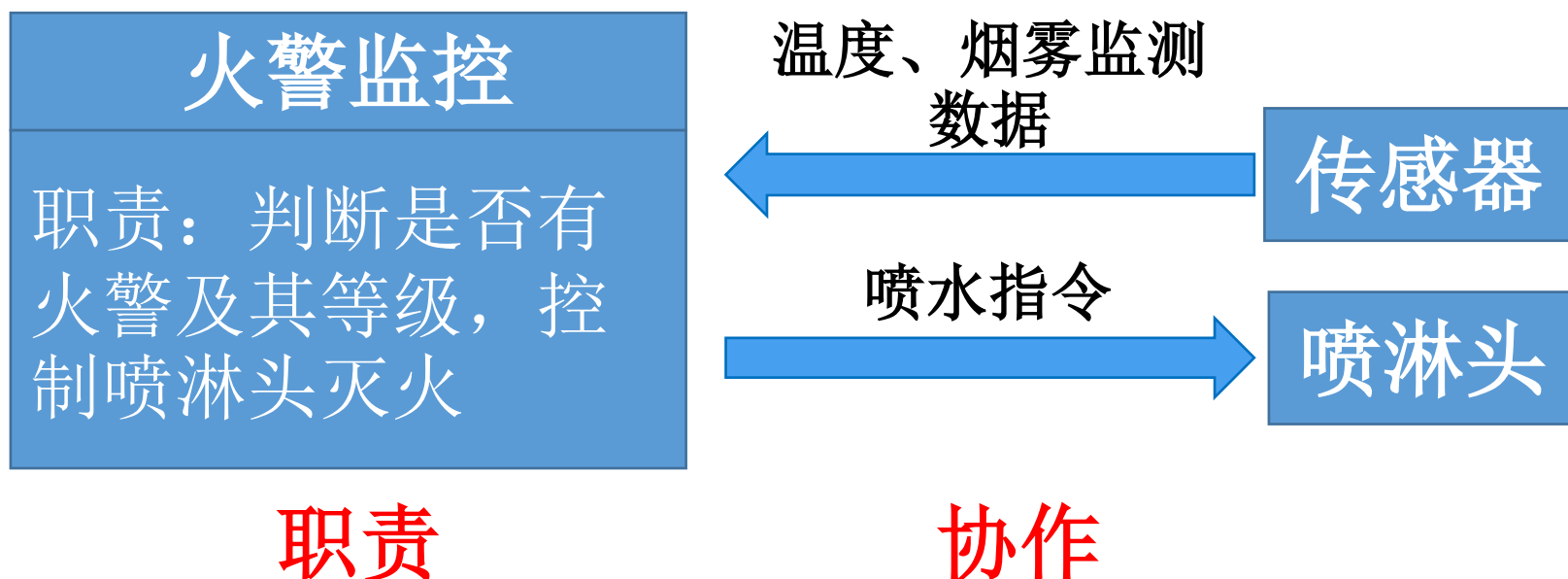
- 1) 客户家庭：安装该系统的客户家庭
- 2) 房屋主人：客户家庭的主人
- 3) 系统密码：客户家庭登录系统的密码
- 4) 联系电话：客户家庭的联系电话
- 5) 摄像头：安装在家庭中的摄像头
- 6) 系统安装：系统的安装过程
- 7) 报警事件：系统的一次报警

定义属性和操作

- 属性：反映事物根本特性，是目标系统运行所需要的
- 操作：基于类的属性及关联关系的知识实现，是目标系统运行所需要的
- 典型操作类型
 - ✓ 操纵相关数据（增、删、改、格式化等）
 - ✓ 查询对象状态和信息
 - ✓ 执行计算
 - ✓ 监控对象中控制事件的发生
- 取决于类的职责和协作关系的确定

类的职责与协作

- 类通过两种方式来完成自身的职责
 - ✓ 使用自己的操作来操纵自己的属性
 - ✓ 通过协作关系请求其他类的协助



CRC建模

- Class-Responsibility-Collaborator (CRC)
- 使用一组索引卡片来辅助识别分析类并确定其职责和协作关系

Class: FloorPlan	
Description	
Responsibility	Collaborator
Defines floor plan name/type	
Manages floor plan positioning	
Scales floor plan for display	
Incorporates walls, doors, windows	Wall
Shows position of video cameras	Camera

CRC卡片

类的职责

- 将系统的功能和信息适当分配到各个类中，避免过于集中
- 将信息与相关的行为放到同一个类中
- 尽量以一种泛化的方式陈述类的职责
- 将与某个事物相关的信息集中到一个类中
- 将职责在相关的类之间适当进行共享

类的协作

- 将一个类无法通过自身能力完成的职责委托给协作类
- 通过协作发现类之间的关系
 - ✓ **is-part-of**: 一个类由其他成员类构成
 - ✓ **has-knowledge-of**: 一个类需要获取另一个类的信息
 - ✓ **depends-on**: 其他情况, 例如一个类需要使用另一个类的操作

通过CRC模型评审完善类的职责与协作

- 针对各个分析类确定初始的职责和协作关系，分别填写CRC卡片
- 由一个负责人组织评审，将卡片发给评审者，存在协作关系的类不要发给同一个人
- 针对每个场景迭代执行以下步骤
 - ✓ 负责人逐句陈述场景描述，当提到某一个分析类时将令牌(token) 传递给持有对应卡片的人
 - ✓ 拿到令牌的人根据卡片内容描述该类中与之相关的职责，如果该职责涉及与其他类的协作则将令牌传递给协作类进行职责陈述，陈述完成后返回令牌
 - ✓ 大家讨论确定这些职责是否满足场景需求
 - 如果满足则由负责人继续场景陈述
 - 如果不满足则调整分析类以及职责和协作关系

CRC分析练习：超市POS结账系统

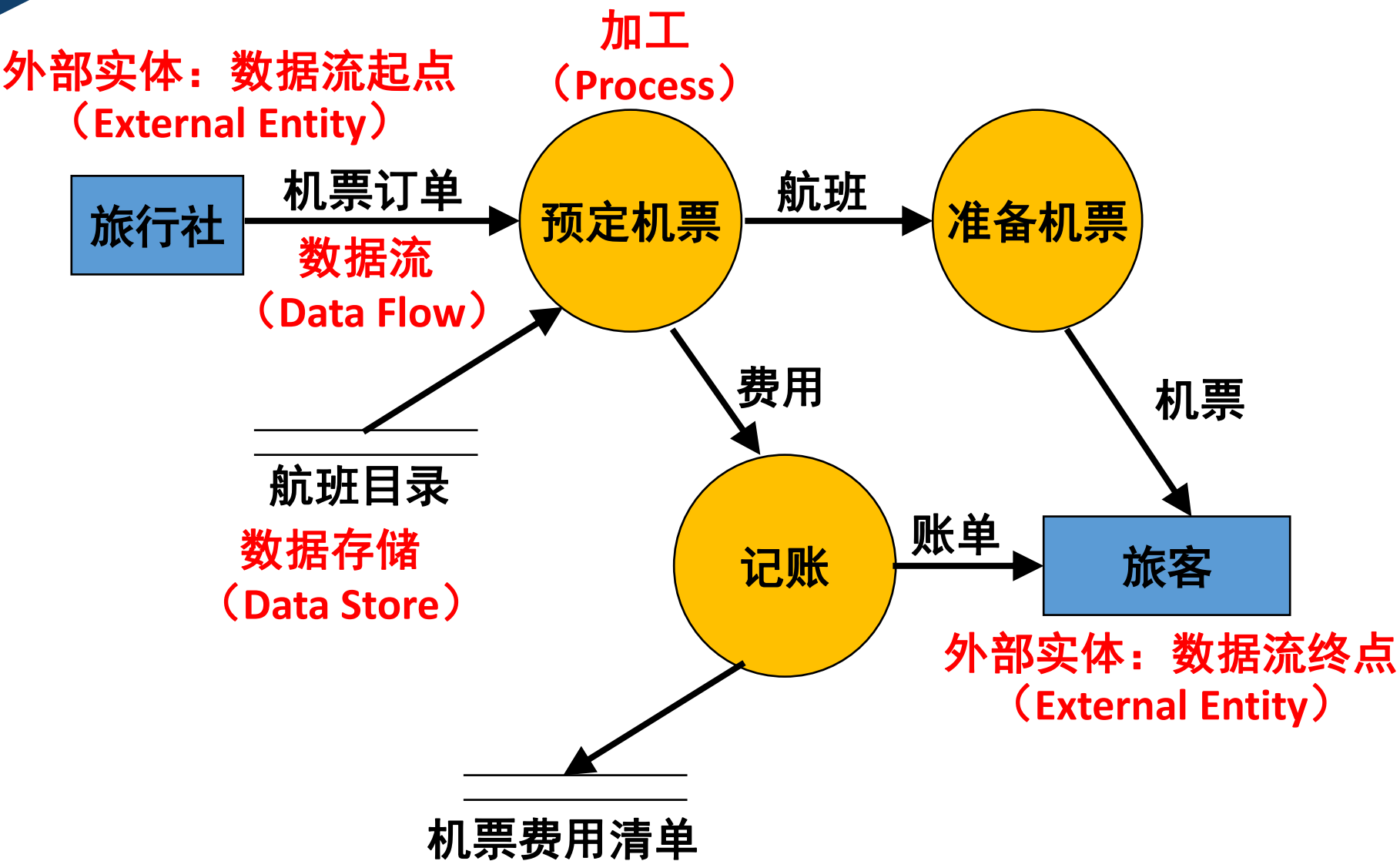
针对以下场景，按照CRC分析确定一组分析类及其职责和协作关系

- 顾客推着购物车来到收银台前，递上会员卡
- 收银员扫描会员卡获取顾客信息后问好（如“xx先生，欢迎光临”）
- 收银员逐一扫描每件商品，系统获取每件商品的名称、规格、单价等信息后累加生成购物单
- 系统生成并显示购物单总金额，读取顾客会员信息，并根据超市当前优惠活动规则以及顾客信息（会员等级、累计消费额等）计算折扣，然后显示折扣后的付款金额
- 顾客递上信用卡，收银员刷卡后确认购物单已支付
- 系统记录此次购物信息（时间、收银台编号、收银员、顾客、金额、折扣、明细等），然后更新相应商品的库存信息

数据流分析与建模

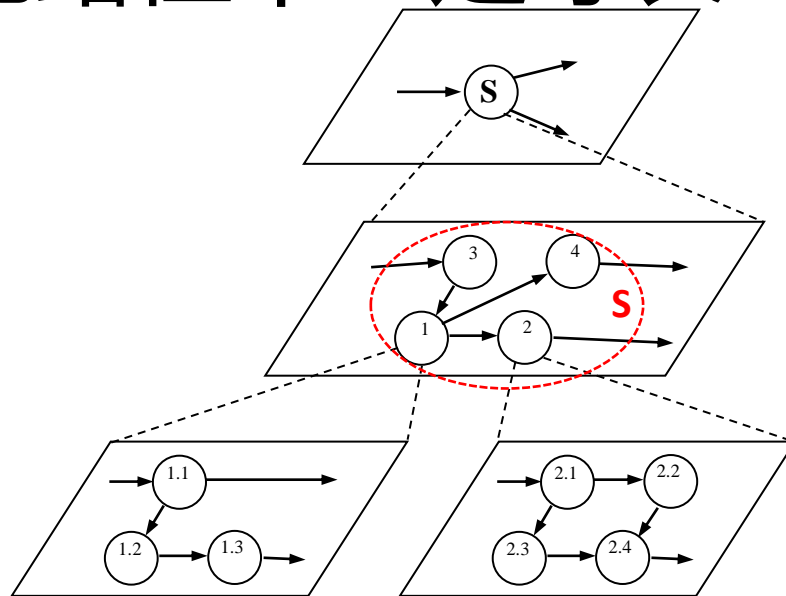
- 以输入和输出数据流的方式描述系统的功能及分解结构
 - ✓ 加工：功能单元，可逐层分解
 - ✓ 数据流：加工以及外部实体之间的数据流动
- 结构化时代的方法，使用分层数据流图及数据字典描述
- 自顶向下、逐步求精，根据软件内部数据传递和变换关系进行分析
 - ✓ 确定系统边界，画出系统上下文图
 - ✓ 自顶向下，画出各层数据流图（对加工进行分解）
 - ✓ 定义数据字典和加工说明

数据流图 (Data Flow Diagram) 示例

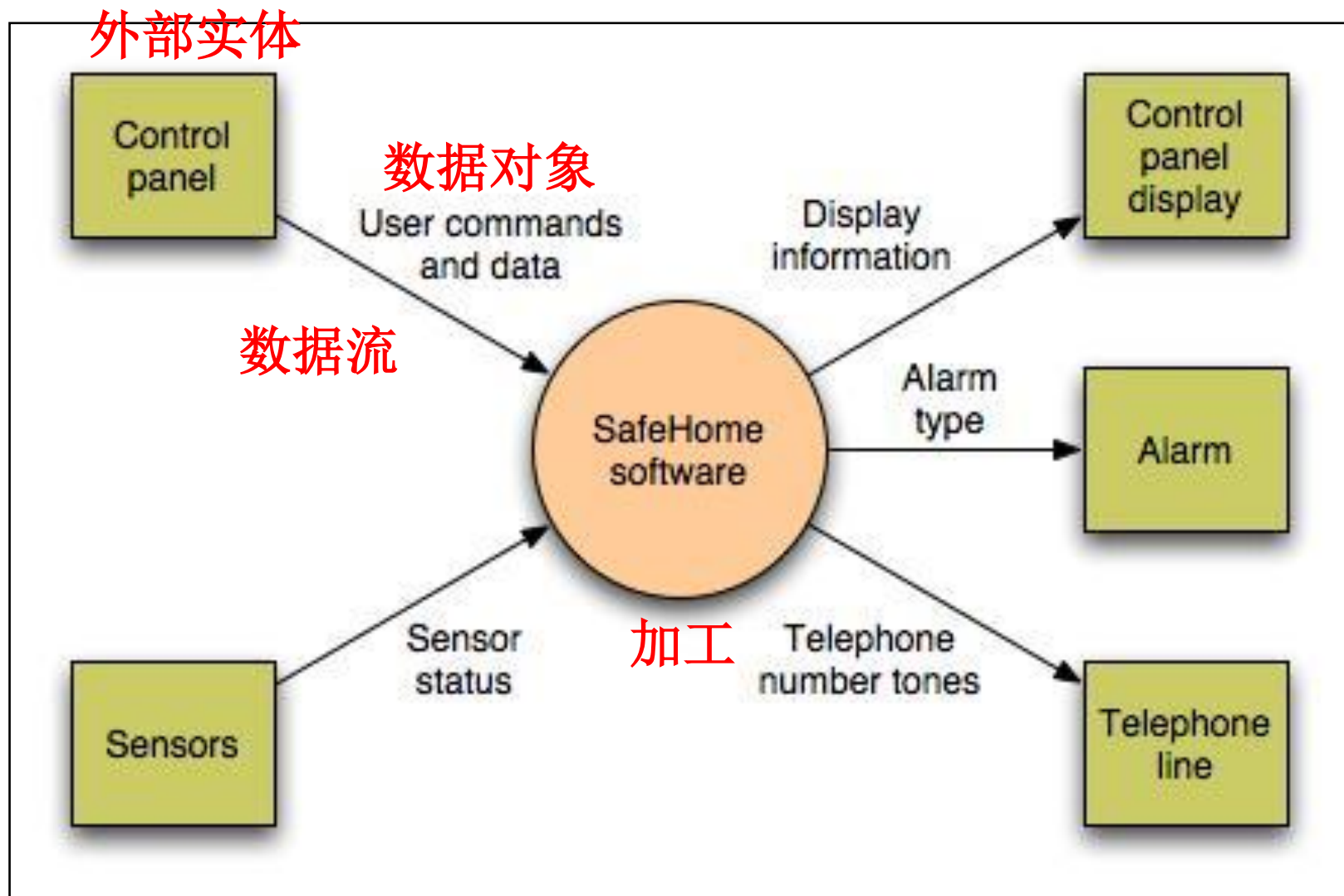


数据流图（DFD）的逐层精化

- 顶层数据流图（0层DFD或上下文图）将系统整体作为一个加工描述
- 从顶层图开始逐步精化展示更多内部处理细节，直到每个加工都足够简单
- 每个精化路径不一定等长

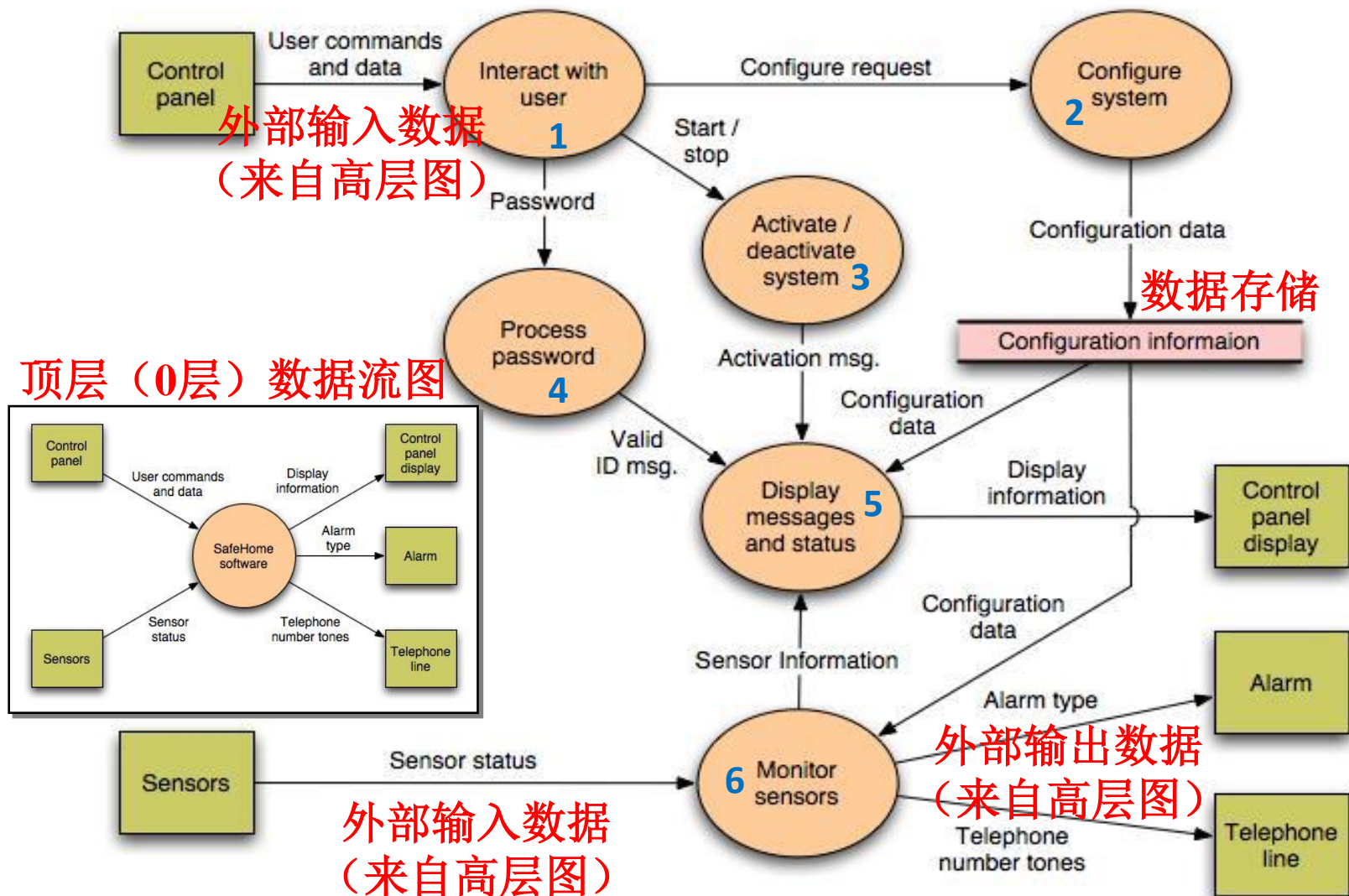


数据流图示例（安全家居系统） -1



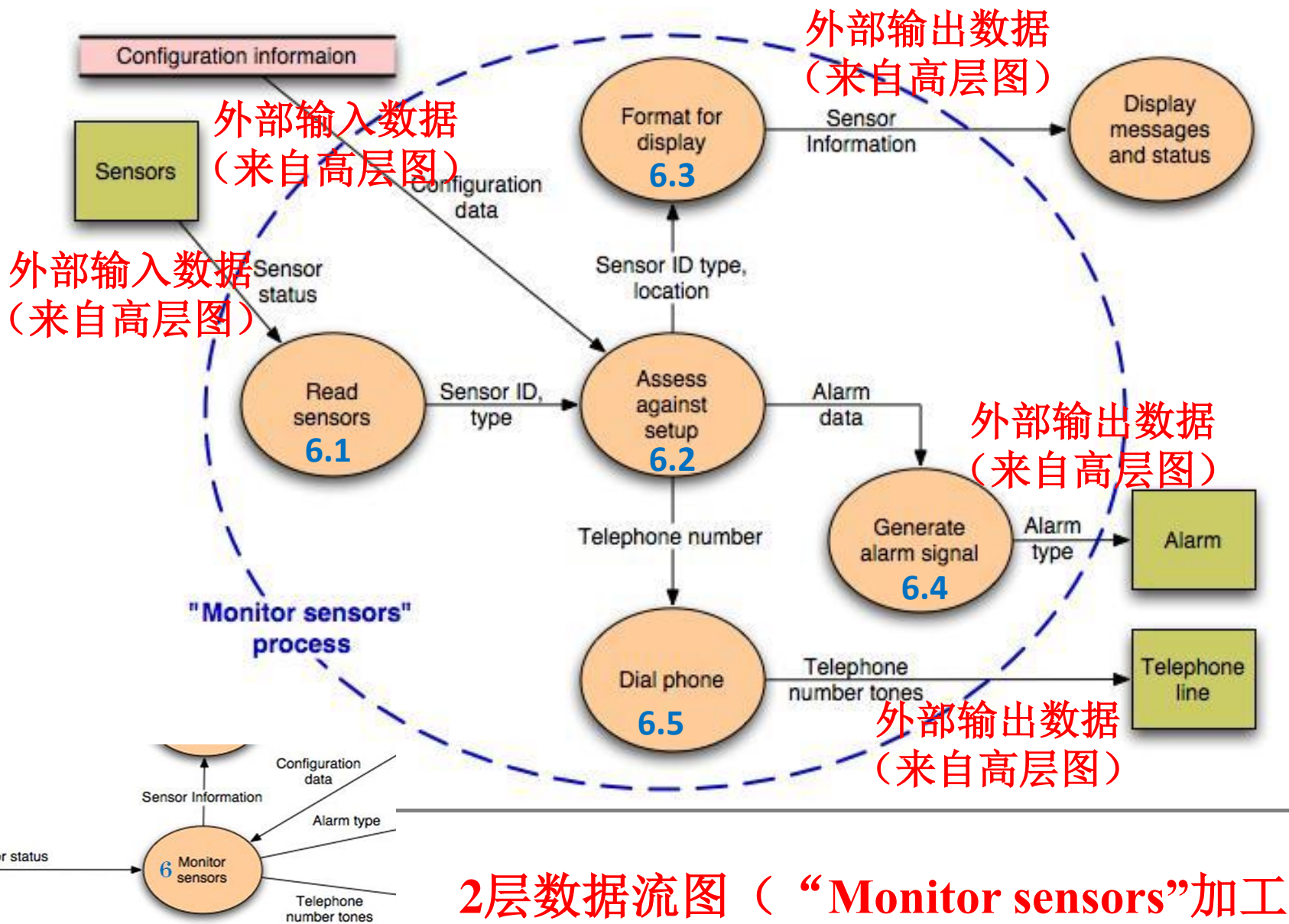
顶层（0层）数据流图（上下文图）

数据流图示例（安全家居系统） -2



1层数据流图（仅有1幅）

数据流图示例（安全家居系统）-3



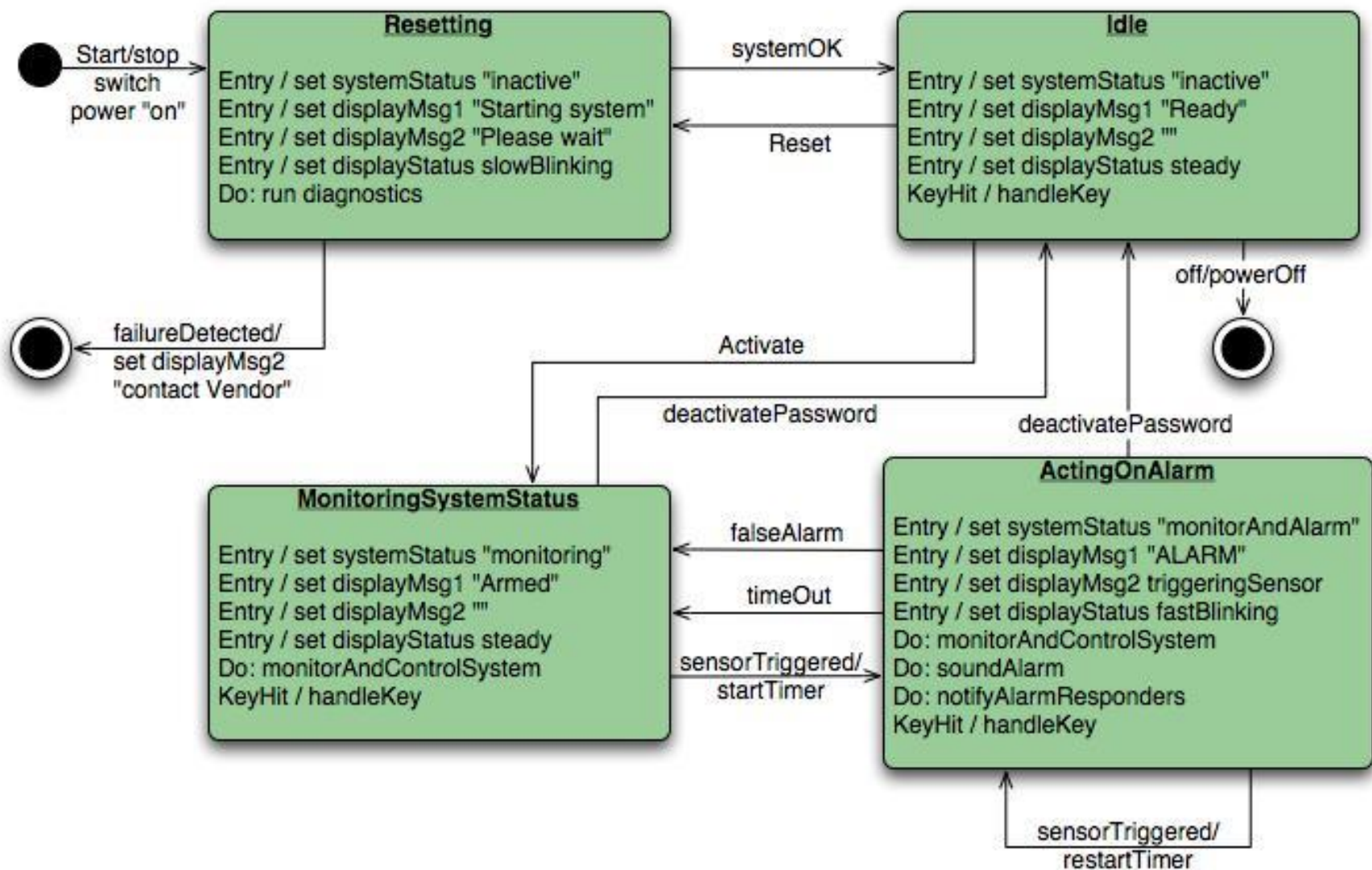
数据流图精化过程中的一致性

- 信息流持续性：图中外部输入和输出数据流应当与父图保持一致
- 数据存储本地化：数据存储只能被同一幅图中的加工访问
- 输出数据应当是对应加工所能够产生的
- 输出数据名称不能跟输入数据一样

行为建模

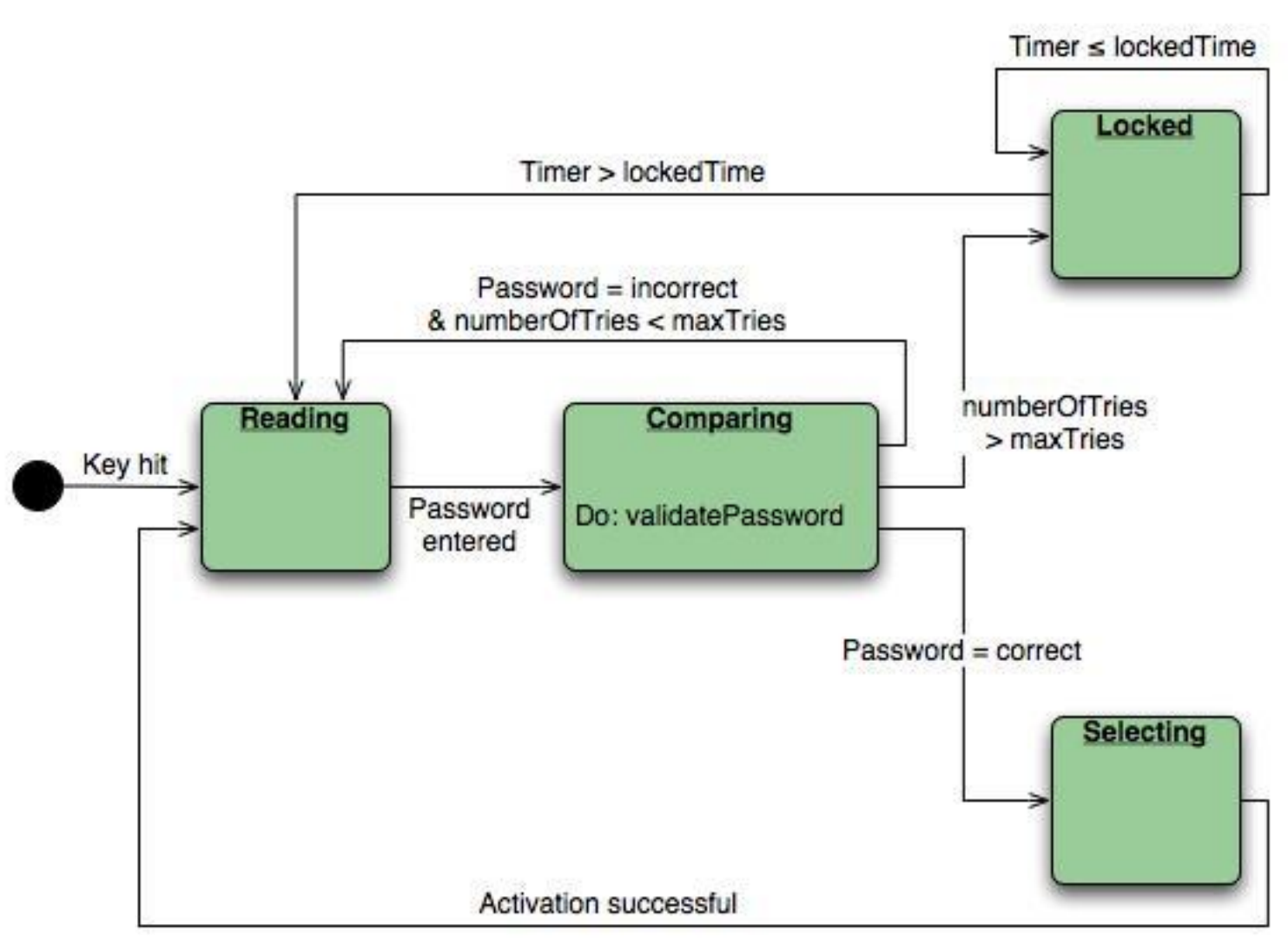
- 描述软件如何响应外部事件和激励
- 可针对软件系统整体或某个业务类（如订单）进行行为建模
- 使用UML状态机图描述
 - ✓ 状态：对象所处的某种情形
 - ✓ 事件：影响对象行为的外部或内部事件
 - ✓ 状态转换：状态之间的变迁，往往在事件驱动下发生
 - ✓ 动作：在状态维持期间或状态转换过程中执行的处理动作

系统整体行为模型 (UML状态机图)



安全家居系统整体行为模型

分析类行为模型 (UML状态机图)



安全家居系统控制面板类（对象）行为模型

阅读建议

- 《软件工程》 第4章
- 《构建之法》 第8、10章

快速阅读后整理问题
在QQ群中提出并讨论

CS2001

软件工程

End

15. 软件需求
— 软件需求分析与建模