

第三章 可视化建模基础

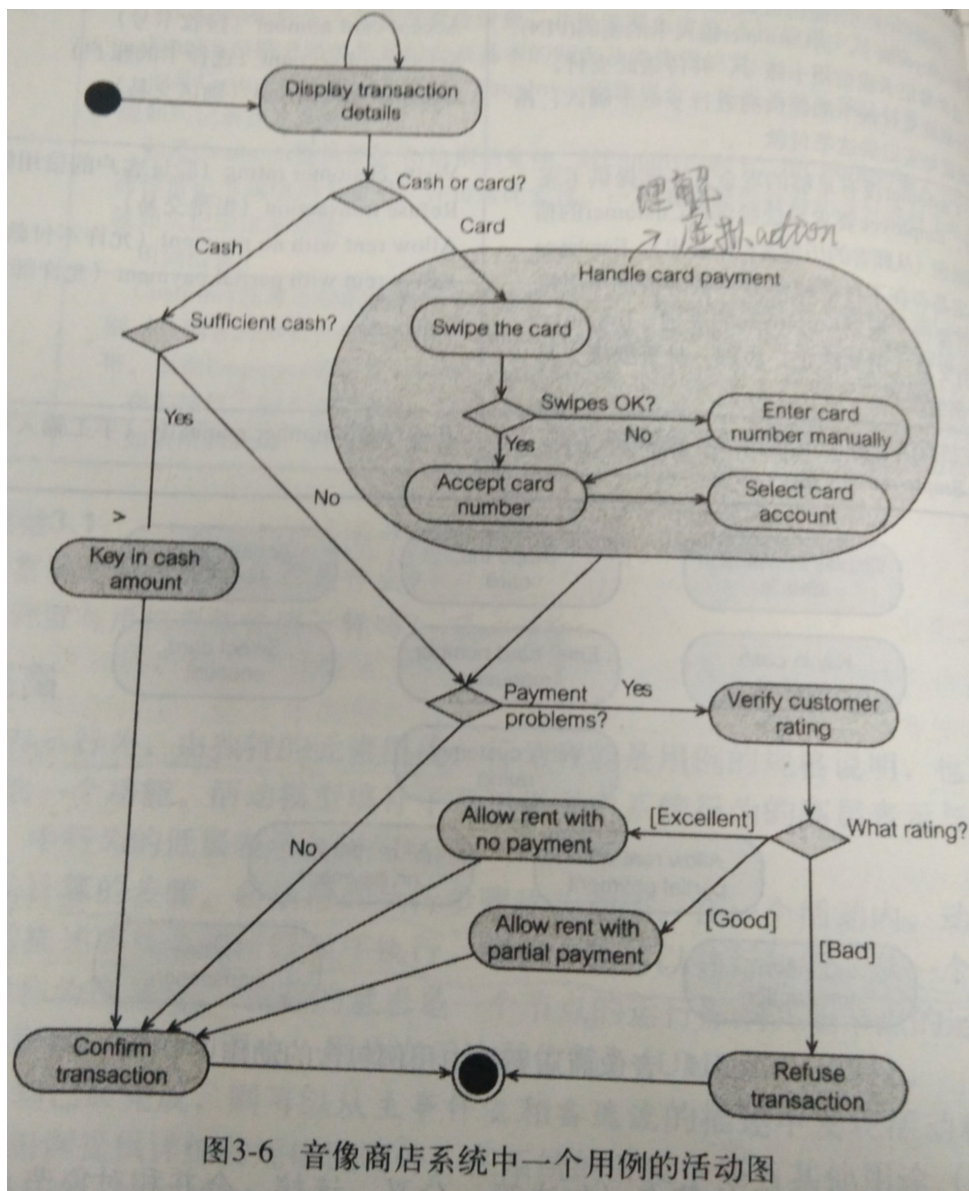
2018年9月21日 8:27

1. 用例视图Use case view:

- a. 用例Use case: 描述系统行为, 所有用例都直接或间接的对某一个或几个参与者相关联
- b. 参与者actor: 描述一类角色 (人、事务)
- c. 用例图: include , extend的正确使用
- d. 用例文档化
 - i. 简要描述
 - ii. 涉及的参与者
 - iii. 用例开始所需要的前置条件
 - iv. 事件流的详细描述
 - 1) 主事件流: 将主事件流分解为事件子流
 - 2) 备选流: 定义异常情况
 - v. 后置条件
- e. 评判要素:按照自己的假设能否说通, 前后一致

2. 活动视图activity view: 描述用例内部的活动细节,本质——支持并发的程序流程图, 图形形式描述用例中的事件流

- o 在活动图中, 用**分叉和交汇**Bifurcation and intersection**定义并发计算线程, 一个词都不能错**
- a. 流程图可以转换为程序代码, 但是活动图转换成代码还很遥远
- b. 描述用例的高层业务过程或者底层的算法过程, 即活动模型填补了用例模型中系统行为的高层表示与交互模型 (时序图和通信图) 中行为的底层表示之间的空隙。
- c. 活动图显示计算的步骤。将活动的执行步骤称为**动作**。活动图描述哪些步骤可以并发执行或者可以顺序执行。从一个动作到下一个动作控制的流程称为控制流。



3. 结构视图Structure view: 描述程序的数据结构以及它们的关系, 比如类图的**实体类图**

a. 结构视图表示系统的**静态视图**——表示数据结构、数据关系、作用在这些数据上的操作

b. 持久化类

i. 实体类 (模型类), 表示持久的数据库对象, 如订单、运输货物

c. 非持久化类

i. 部分实体类、

ii. 表现类 (边界、视图类), 如GUI屏幕表单

iii. 控制类, 控制程序逻辑及处理用户事件的类

iv. 资源类, 负责与外部数据源通信的类

v. 中介类 (调解), 为满足业务交易, 管理内存高速缓存中实体对象的类

d. 需求分析阶段必须拿下实体类建模

4. 交互视图interaction view: **时序图可以与通信图相互转换**

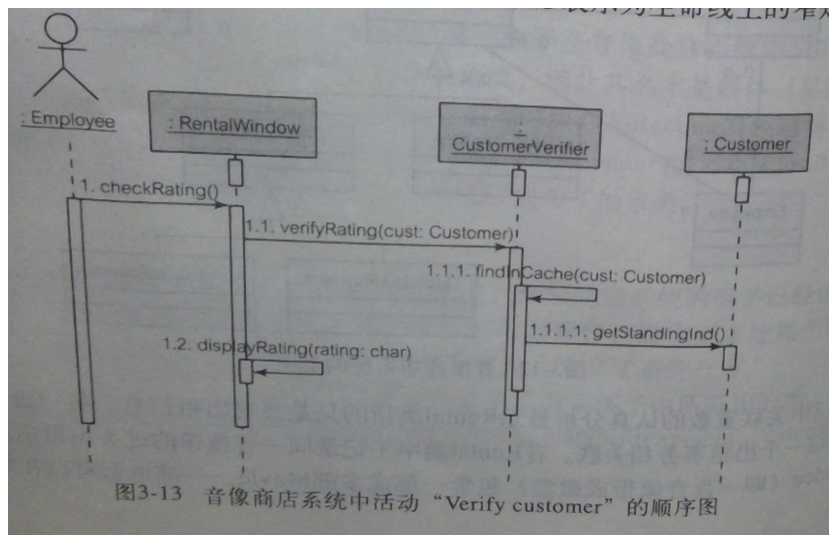
○ 捕获对象之间的交互, 为了执行一个用例或用例的一部分, 这些对象之间需要通信

○ 活动建模和交互建模都表示用例的实现。

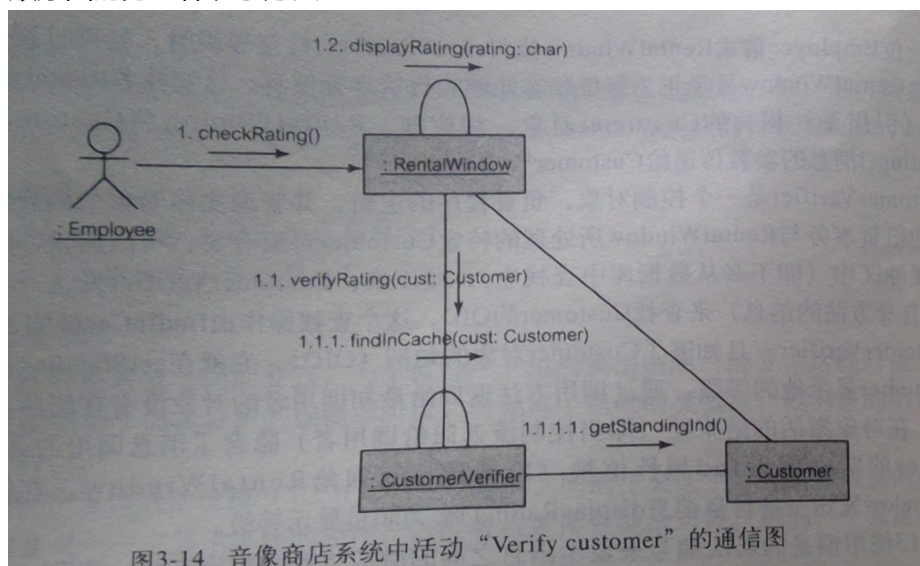
▪ 活动图更抽象, 经常捕获**整个用例**的行为

▪ 交互用例更详细, 趋向于对**用例的某些部分**的建模, 有时候一个交互图对活动图中的单个活动建模。

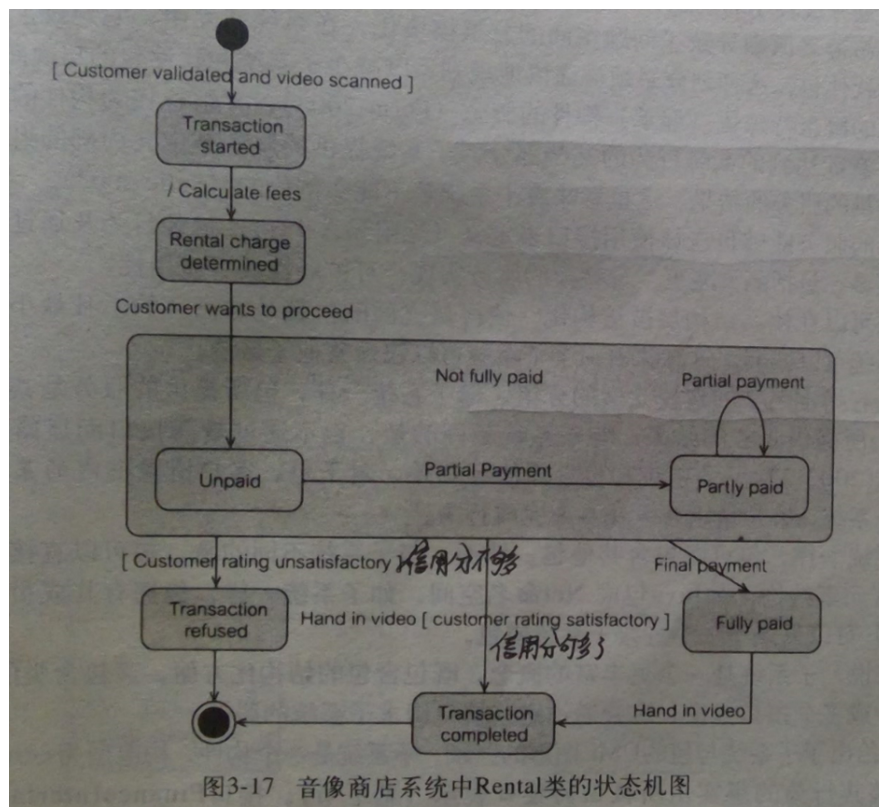
- a. 主要描述某些对象在一个用例执行期间对象间的交互协作关系
 - b. 时序图强调时间序列
 - i. 水平维上的角色可以是对象、类、接口，垂直维上从上到下显示消息的顺序
 - ii. 垂直线是对象的生命线，表示对象的生命周期
 - iii. 激活（activation）或执行规格说明，生命线上的窄矩形
 - iv. 箭头，从调用对象发给被调用对象，每条消息实际是被调用对象的一个方法
 - 形式参数可以作为方法定义的一部分
- 1) 消息与方法间一一对应，每一个消息都应该在类图上以一个方法来体现
 - 2) 时序图中的对象都应该是类图中的类，即使类图刚开始可能没有边界类或者控制类，我也要根据时序图的对象来增加类图中的新类



- c. 通信图communication/协作图collaboration：描述对象之间的关系，强调对象之间的交互
 - 顺序图的另一种表示方法



5. 状态机图State machine view：描述某一个类或一个对象不同状态的改变
 - 状态机图是状态和由事件引发的转换的偶图
 - a. 通常用在比较复杂的类，状态转换需要外部条件
 - b. 转换的完整描述由3部分组成：Event(parameters)[guard]/action,当条件守卫的值为true,才能执行改变状态的事件
 - c. 事件是影响对象的迅速发生的事件，它可能有参数



6. 实现视图implementation view：出现在设计阶段的后期，给出在不同环境下的配置方案（了解）

- 属于物理建模的范畴，但定义它们一定要适当的考虑系统的逻辑结构
- 子系统可以实现一定的功能，但是构件不一定
- 包是具有名字的建模元素的分组
- 节点是人工制品可以在上面部署运行的计算资源
- 部署图关注结构和节点依赖建模，节点定义系统的实现环境

• 另一种UML 逻辑模型分类：

- 状态模型（类图）你有什么需要什么
 - 系统的静态视图
- 行为模型（用例，活动，交互）你做什么
 - 系统的操作视图
- 状态转换模型（状态机图）你做完后的结果
 - 系统的动态视图