

构建用例模型的步骤

- 第一步：找到所有的参与者和用例
 - 识别出参与者并做简单的描述
 - 识别出用例并做简单的介绍
- 第二步：编写用例
 - 列出用例
 - 给用例事件流程划分重要等级
 - 按照重要程度排序详细描述事件流程

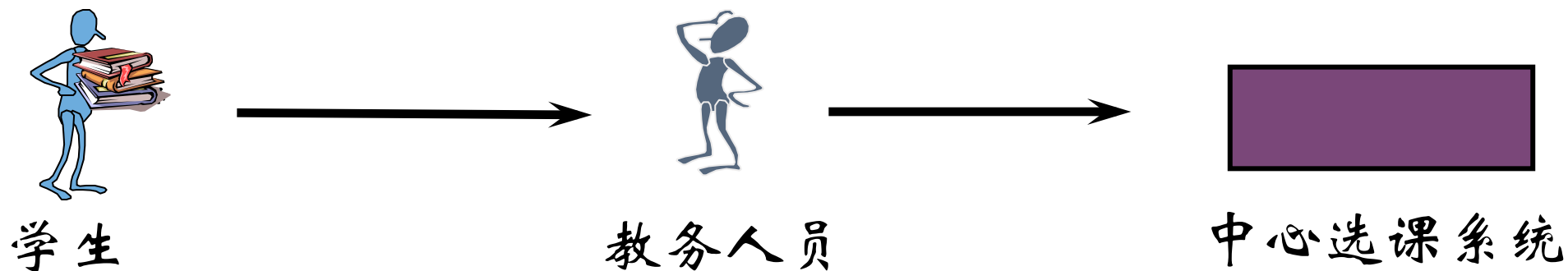


寻找参与者

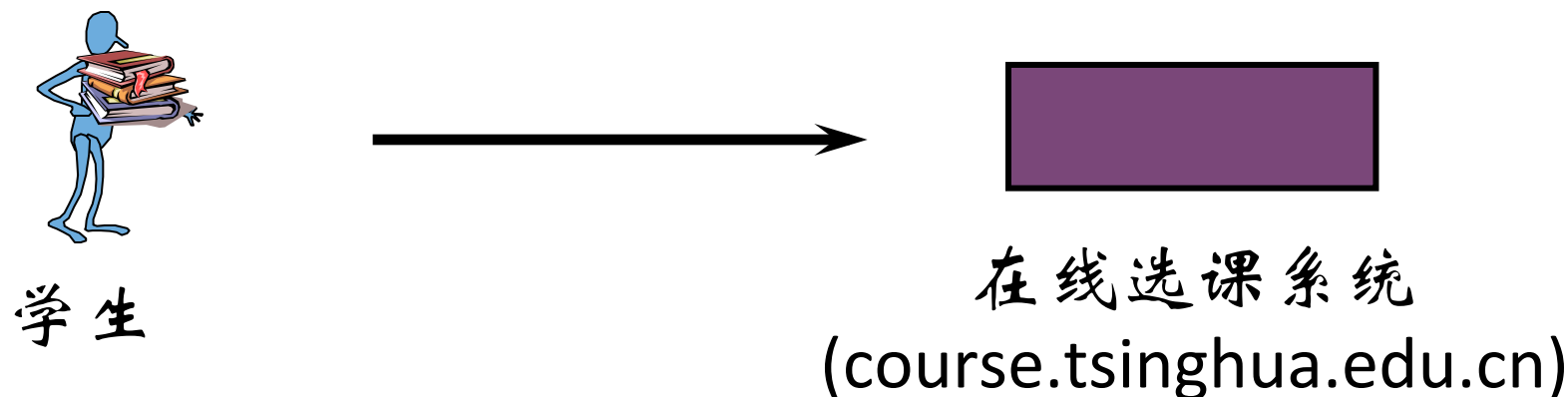
- 谁/什么使用系统?
- 谁/什么从系统中获取信息?
- 谁/什么向系统提供信息?
- 公司的哪个部门会使用系统?
- 谁/什么负责系统的维护?
- 还有哪些其他系统会使用系统?



识别参与者——是谁与系统进行交互？



学生并不直接操作选课系统；是教务人员进行操作。
或者，构建一个基于浏览器的在线应用？



参与者的描述

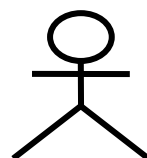
名称

学生

简要描述

注册课程的用户

和用例之间的关系



学生



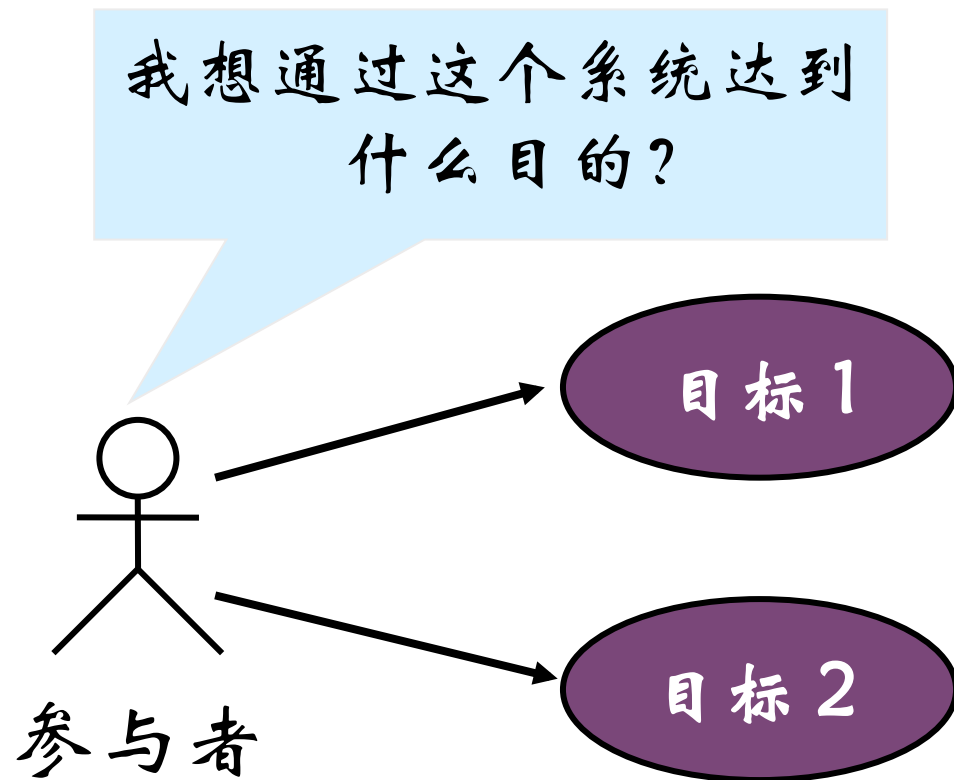
用例描述

参与者建模的检查项

- 是否找全所有的参与者？是否对系统环境中所有的角色进行了描述和建模？
- 每个参与者是否至少与一个用例发生了交互？
- 是否可以为每一个角色找到至少两个实例？
- 不同参与者与系统的交互是否一致，扮演的角色是否相似？如果有，则应该要合并这些参与者作为同一种角色



寻找用例



识别用例

- 每个参与者的目标是什么？
 - 为什么参与者要使用这个系统？
 - 参与者是否需要对系统中数据进行创建，存储，更改，删除或者读取的操作？为什么？
 - 参与者是否需要将外部事件或发生的改变告知系统？
 - 参与者是否需要知道系统内部发生的事件或改变？
- 系统是否能够应对业务中所有的正确行为与操作？

用例的描述

用例的文本描述

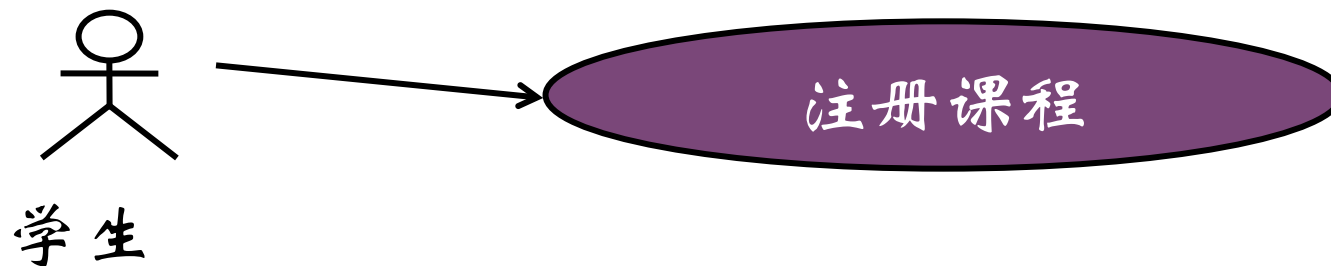
名称

注册课程

简要描述

学生选择下个学期想上的课程。生成必修课和选修课的课表信息。

与参与者的关系



用例的命名

- 表明参与者的目标或者作用
- 使用主动语态：用动词起始
- 设计一系列操作流程(to-do list)
- 几种表达：
 - Register for Courses
 - Registering for Courses
 - Acknowledge Registration
 - Course Registration

哪种表达形式可以表现出参与者的意义或价值？哪些不可以？
你会选择哪个作为你的用例名称？为什么？

用例建模过程中的检查项

- 用例建模是为了表示系统的行为。通过模型可以很容易理解系统进行的操作
- 应该识别出所有的用例，用来表达所有的需求。
- 系统的任何一个特性都可以找到对应的用例
- 用例模型并不包含多余的行为；所有的用例可以追溯到系统的功能性需求作为验证。
- 去掉所有的CRUD类的用例
 - 创建(Create), 查找(Retrieve), 更新(Update), 删除>Delete)



构建用例模型的步骤

- 第一步：找到所有的参与者和用例
 - 识别出参与者并做简单的描述
 - 识别出用例并做简单的介绍
- 第二步：编写用例
 - 找出用例
 - 给用例事件流程划分重要等级
 - 按照重要程度排序详细描述事件流程



寻找用例的方法

- 和用户交互
- 基本策略：把自己当作actor，与设想中的系统进行交互。

考虑：

- 系统交互的目的是什么？
- 需要向系统输入什么信息？
- 希望由系统进行什么处理并从它得到何种结果？

注意：确定Use Case和确定actor不能截然分开

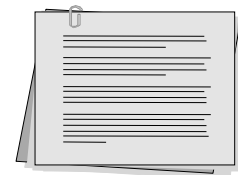
用例建模的过程：用例图→用例提纲→用例详细规约



+ 用例简单描述

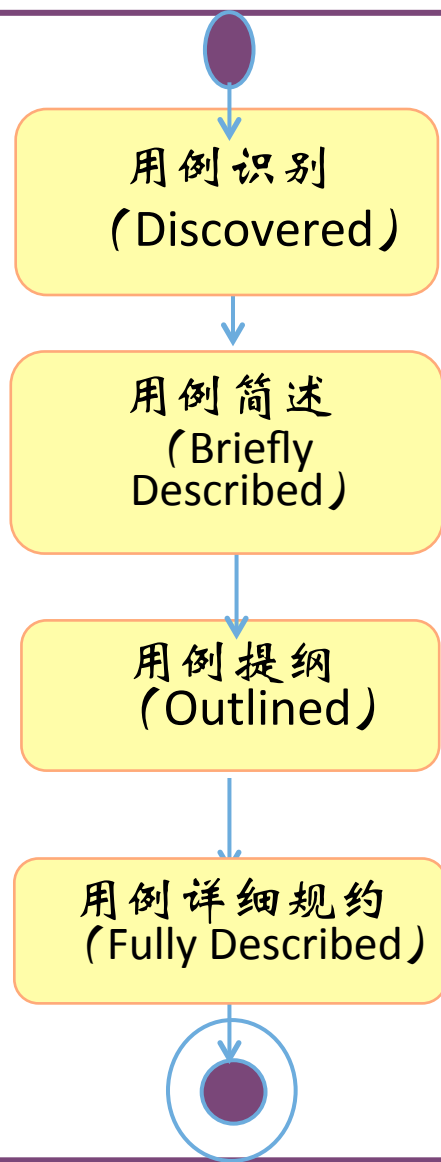


注册课程用例提纲
+ 粗略列出事件流程
• 大体步骤



注册课程用例的详细规约
+ 列出详细的事件流程
• 按步骤（详细）
+ 特殊的规约说明
+ 前置/后置条件

用例的全生命周期



结束课程注册

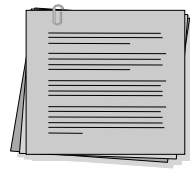
简述: 教务人员可以通过这个用例结束课程注册环节。学生人数不足的课程将被取消。收费系统会通知所有未被取消的课程的选择学生进行缴费。



结束注册环节 (概述)

- 事件流

- ...



结束注册环节 (用例规约)

- 详细的事件流

特殊的需求

- 前置/后置条件

用例简述的例子

● 用例简述:

一段简洁的摘要，主要描述用例的成功场景

● 处理购物交易:

客户带着要购买的货物到收款处，收银员使用POS机扫描记录每一种预购买的货物。系统计算总价并打印清单。客户付款，系统验证并保存销售记录。系统更新库存，客户得到收条并带着货物离开。

用例概述的例子

- 用例概述:

- 非正式、随意的格式
- 非正式段落, 覆盖各种场景

退货处理

主成功场景:

客户带着要退的货物到达收款处, 出纳员使用POS系统记录每一个要退货的货物, ...

候选场景:

若信用验证失败, 通知客户并要求使用其他付款方法

若系统检测到与外界计税系统通信失败, ...

详细用例规约的例子

用例图

用例名称: 下订单 (Place Order)

前置条件: 用户通过身份认证登录系统

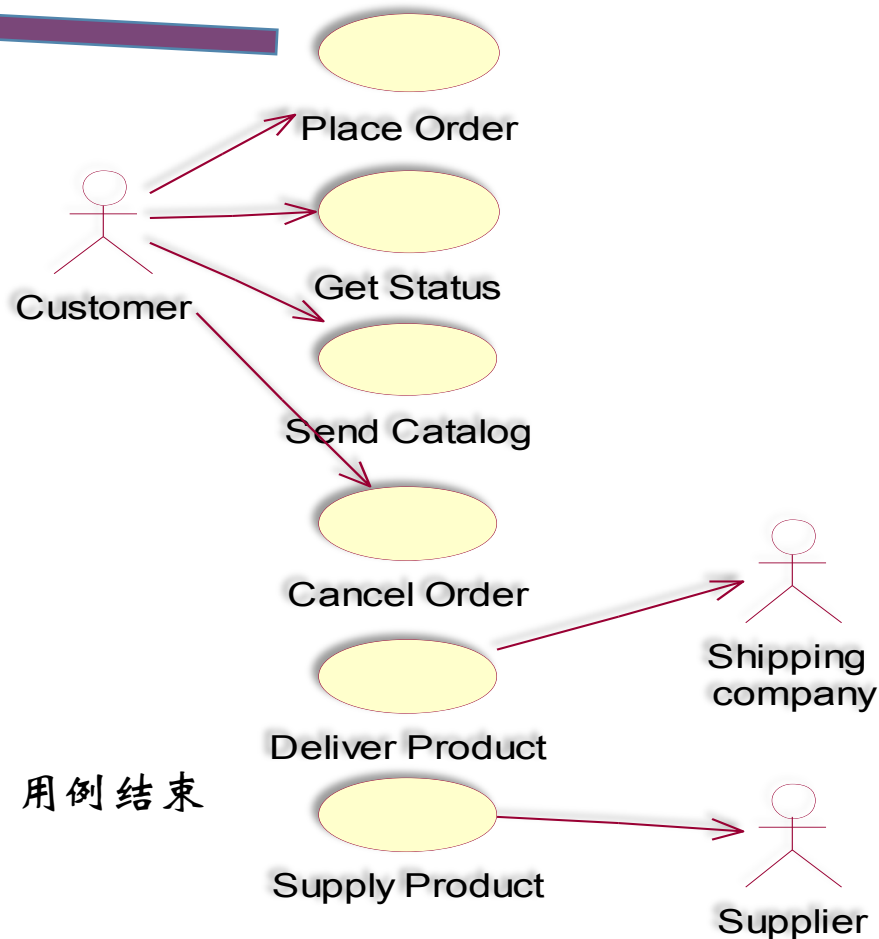
描述:

1. 当顾客选择“下订单”时, 进入该用例流程
2. 顾客输入姓名和地址信息
3. 如果顾客仅输入了邮政编码, 系统会提供州和城市信息
4. 顾客输入代购买的物品编码
5. 系统显示每个产品的描述信息和价格信息
6. 系统将持续记录顾客输入的所有商品信息和相应的总价
7. 顾客输入信用卡付账信息
8. 顾客选择提交(Submit)
9. 系统确认信息, 保存待付款订单信息, 将账单信息提交给账务系统
10. 确认付账成功后, 系统标记账单为完成状态, 向顾客显示账单ID信息, 用例结束

异常情况:

第9步中, 如果信息不正确, 系统将提示顾客对相应信息进行修改

后置条件: 系统保存订单并且标记为已确认。



用例文档模板

UC_id: 用例名

描述: 对该用例的一句或两句的描述。

参与者: 参与该用例的参与者。

包含: 该用例所包含的用例, 以及包含它的用例。

扩展: 该用例可以扩展的用例, 以及扩展它的用例。

泛化: 若该用例的子用例和父用例。

前置条件: 启动此用例所必须具备的条件。

细节: 该用例的细节。(基本流与可选流)

后置条件: 在该用例结束时确保成立的条件。

例外: 在该用例的执行的过程中可能引起的例外*。

限制: 在应用中可能出现的任何限制*。

注释: 提供可能对该用例是重要的任何附加信息。

总结：Use Case模型的建立步骤

- (1) 找出系统外部的参与者和外部系统，确定系统的边界和范围；
- (2) 确定每一个参与者所期望的系统行为；
- (3) 把这些系统行为命名为Use Case；
- (4) 使用泛化、包含、扩展等关系处理系统行为的公共或变更部分；
- (5) 编制每一个Use Case的脚本；
- (6) 绘制Use Case图；
- (7) 区分主事件流和异常情况的事件流，可以把表示异常情况的事件流作为单独的Use Case处理；
- (8) 细化Use Case图，解决Use Case间的重复与冲突问题。