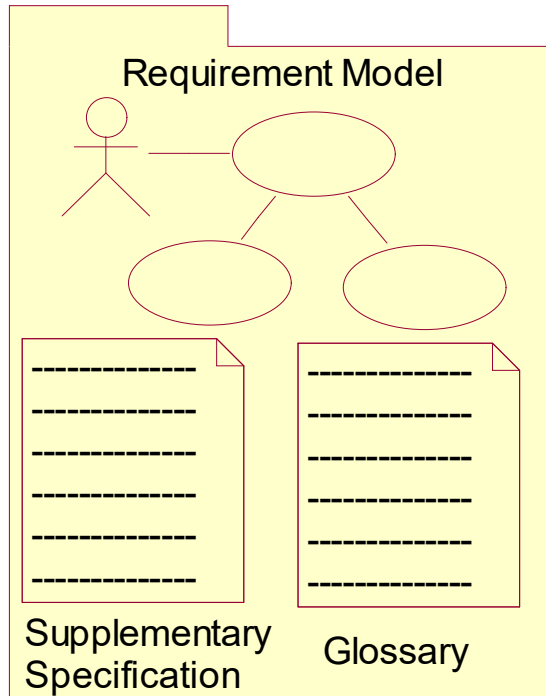


软件开发综合实训

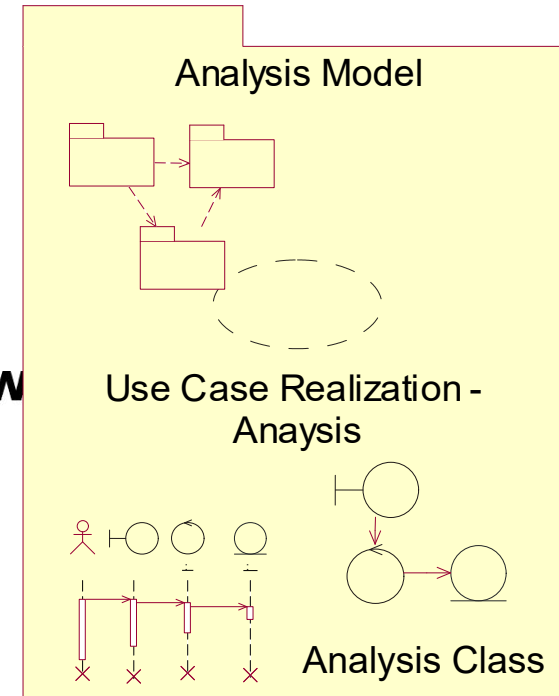
第04讲 用例分析



从需求到分析



Analysis workflow



从需求到分析

分析是建立在需求收集的基础上

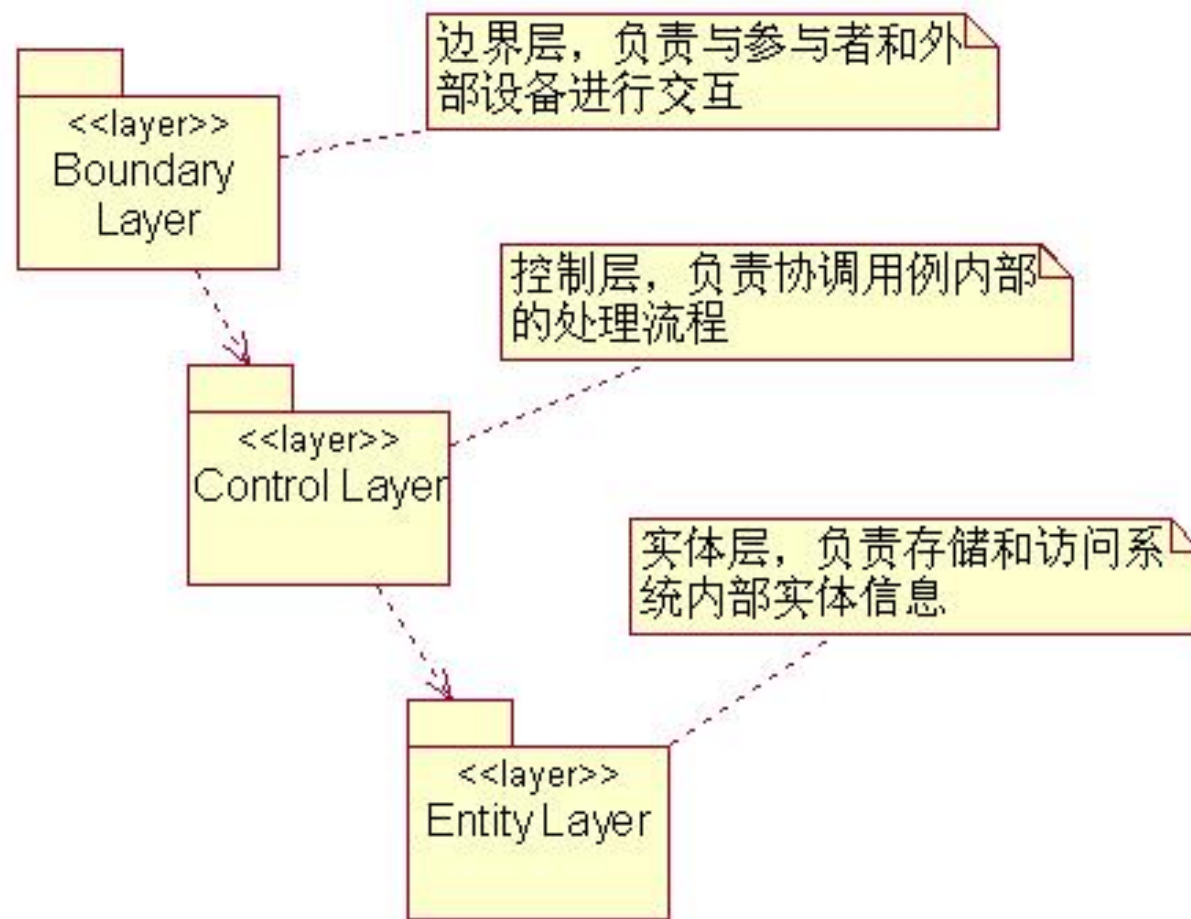
分析与需求捕获在很大程度上重叠，这两个活动常常相辅相成，为了澄清和找出任何遗漏或歪曲的需求，常常需要在需求之上作一些分析。

如何把握这种介于业务和技术之间的“度”

过度的分析会陷入设计误区，从而难以有效的达到分析的目标。

不够深入的分析则容易遗漏那些重要的信息，从而无法及早发现并处理需求中的问题。

定义基础架构

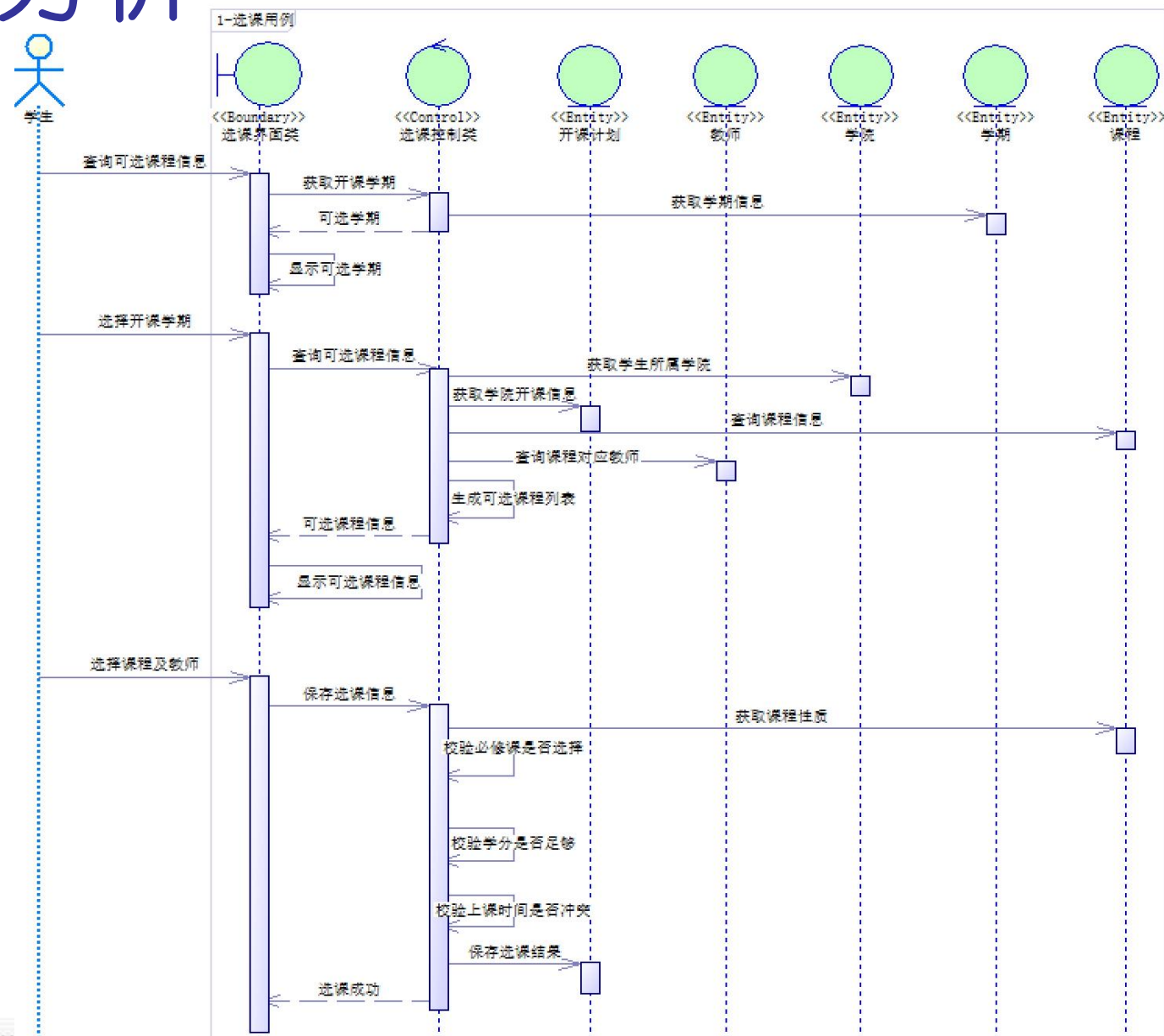


定义基础架构

分层架构B-C-E

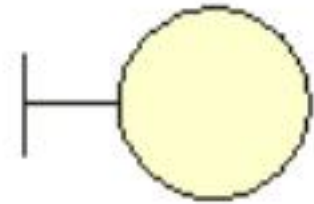
- 边界层 (Boundary) 负责系统与参与者之间的交互
- 控制层 (Control) 处理系统的控制逻辑
- 实体层 (Entity) 管理系统使用的信息

用例交互分析



用例交互分析

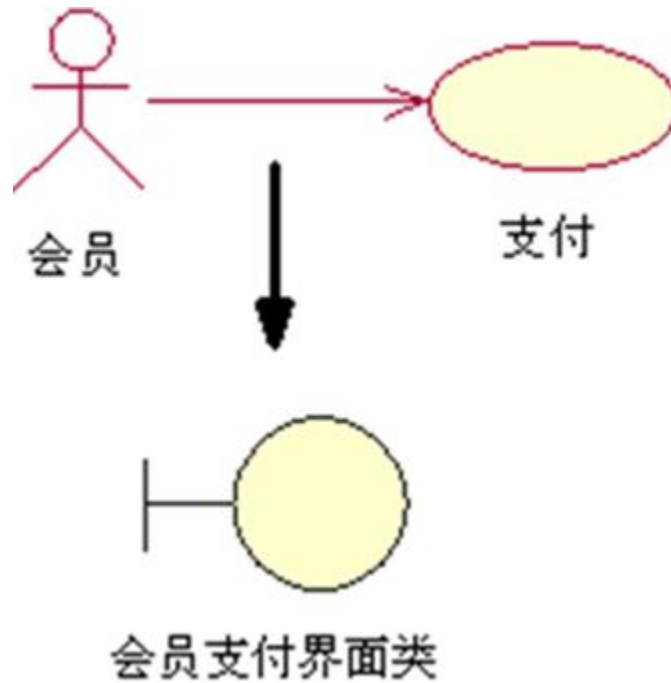
- 边界类
 - 边界类表示系统与参与者之间的边界
 - 代表系统与环境交互
 - 是接口和外部事物的中间体
 - 版型<<boundary>>
 - 两类边界类
 - 用户界面类
 - 系统和设备接口类



我就是边界类

用例交互分析

- 边界类-示例
 - 每对参与者/用例定义一个边界类



用例交互分析

- 控制类

- 控制类表示系统的控制逻辑

- 系统行为的协调器
 - 版型<<control>>

- 识别控制类

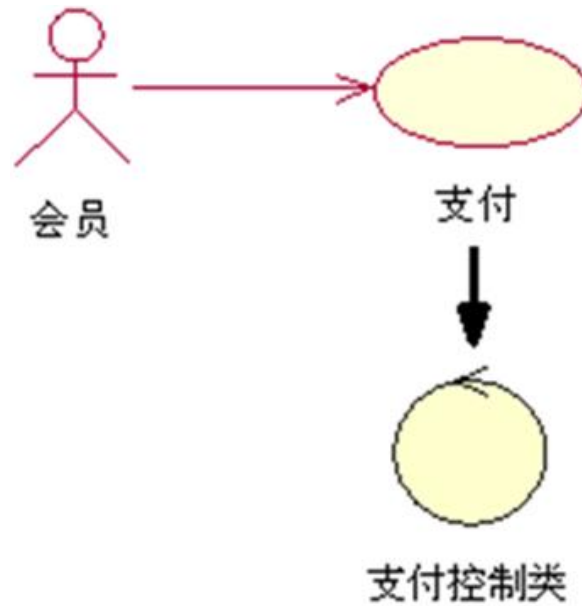
- 在系统开发早期，为一个用例定义一个控制类，负责该用例的控制逻辑
 - 针对复杂用例，可为备选路径分别定义不同控制类



我就是控制类

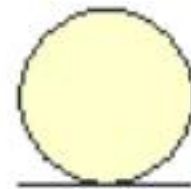
用例交互分析

- 控制类-示例
 - 每个用例定义一个控制类
 - 持续分析，一个复杂用例的控制类可以发展为多个



用例交互分析

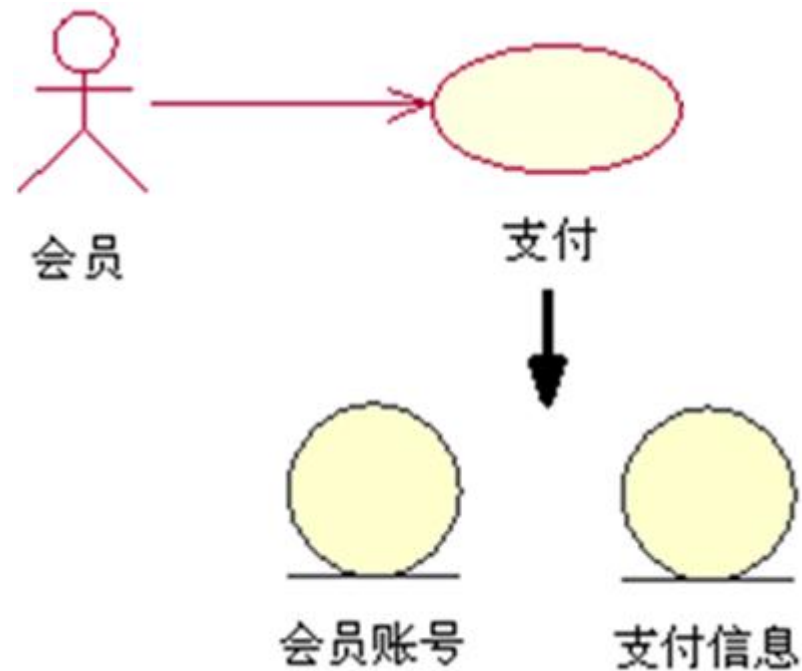
- 实体类
 - 实体代表了待开发系统的核心概念
 - 实体类提供了另一个理解系统的观点
 - 显示了系统的逻辑数据结构
 - 兼顾了数据存储、数据操作的职责
 - 版型<<entity>>
 - 可以从以下中找到实体类
 - 业务模型(业务建模)
 - 用例事件流(需求)
 - 词汇表(需求)



我就是实体类

用例交互分析

- 实体类-示例
 - “支付”用例基本路径中的候选实体类



用例交互分析

- 识别实体类的步骤
 - 分析用例事件流中的**名词**、**名词短语**找出系统所需的实体对象，这些名词可能是：
 - 对象、对象的特征和状态
 - 参与者、描述信息、系统之外的
 - 从这些名词、名词短语中进行筛选，抽取出系统对象，并抽象成类
 - 综合考虑在系统中的意义、作用和职责
 - 对于所识别的类进行命名

用例交互分析

- 识别实体类：名词筛选法
 - 将用例事件流作为输入，找出名词或名词性短语，形成了实体类的初始候选列表，进行：
 - 合并那些含义相同的名词
 - 删除那些系统不需要处理的名词
 - 删除作为参与者的名词
 - 删除与实现相关的名词
 - 删除那些作为其他实体类属性的名词
 - 对剩余的名词，综合考虑它在当前用例以及整个系统中的含义、作用以及职责，并基于此确定合适的名字，作为初始实体类存在

用例交互分析

• 识别实体类：名词筛选法

基本事件流↵

- (1) 用例起始于学生需要选课↵
- (2) 学生查询可选学期 (D-1) ↵
- (3) 系统显示可选的学期信息 (A-1) (D-2) ↵
- (4) 学生选择某个可选的学期↵
- (5) 系统显示该可选学期的所有可选课程信息列表 (A-2) (D-3) ↵
- (6) 学生确认选择的课程↵
- (7) 系统保存学生的选课信息 (B-1) , 显示选课成功 (A-3) , 用例结束↵

补充约束-数据需求↵

D-1 无特定的查询条件↵

D-2 可选学期信息是学年-学期格式, 例: 22-23 上↵

D-3 可选课程信息包括: 课程名称、课程性质 (必修/选修)、学时、学分、学院、学期、上课教师姓名、上课时间、上课教室↵

补充约束-业务规则↵

B-1 必修课程必须全部选择↵

B-2 所有选择课程的学分要满足该学期学分要求↵

B-3 所选择课程的上课时间不能冲突↵

备选事件流↵

A-* 学生在确认选课前, 随时都可以中止该用例↵

- (1) 系统提醒学生当前所有操作都会被取消↵
- (2) 学生确认后, 当前用例结束, 也可选择取消, 则继续后续操作↵

A-1 没有找到可选学期↵

- (1) 系统显示没有找到可选学期↵
- (2) 学生选择结束该用例↵

A-2 没有找到可选课程↵

- (1) 系统显示没有找到可选课程↵
- (2) 学生选择结束该用例↵

A-3 系统保存失败↵

- (1) 系统显示保存失败信息, 并提醒学生重新提交↵
- (2) 学生可以重新提交本次选择的课程, 也可以结束该用例↵

用例交互分析

- 顺序图分析交互
 - 基本顺序是ABCE (**A**ctor、**B**oundary、**C**ontrol、**E**ntity)

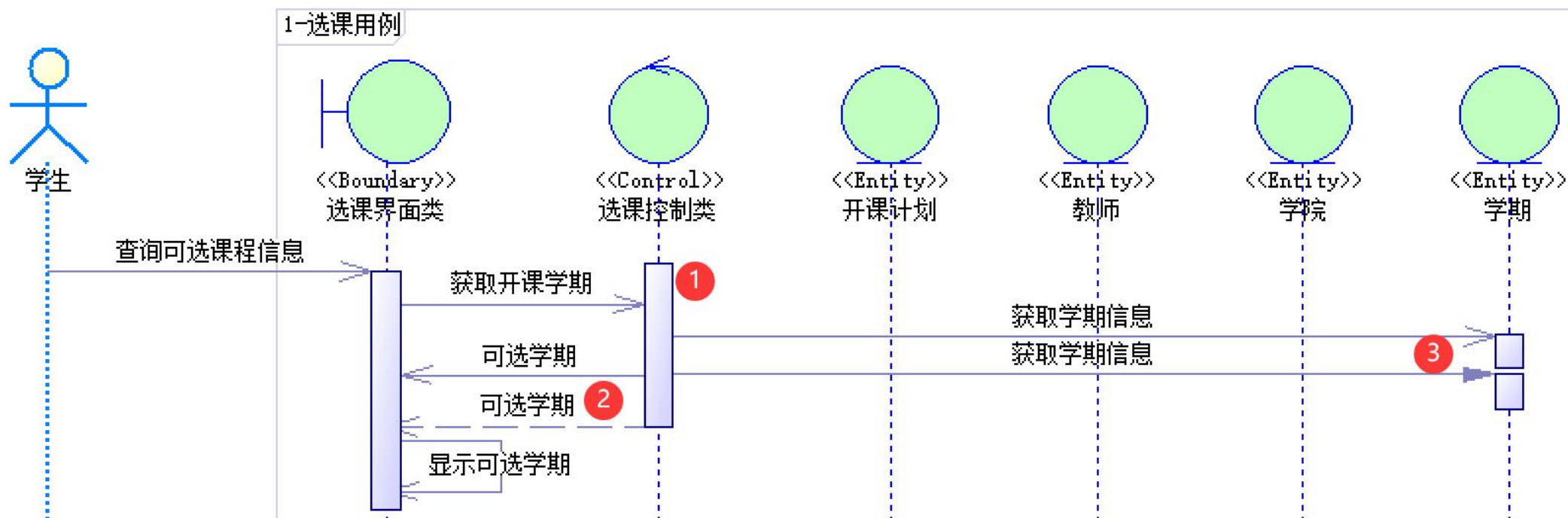


用例交互分析

- 以分析类的构造型作为分配标准
 - 边界类：承担与参与者进行通信的职责
 - 控制类：承担协调用例参与者与数据操作之间交互的职责
 - 实体类：承担对被封装的内部数据进行操作的职责

核心：“刻画”系统用例中的流程！

用例交互分析



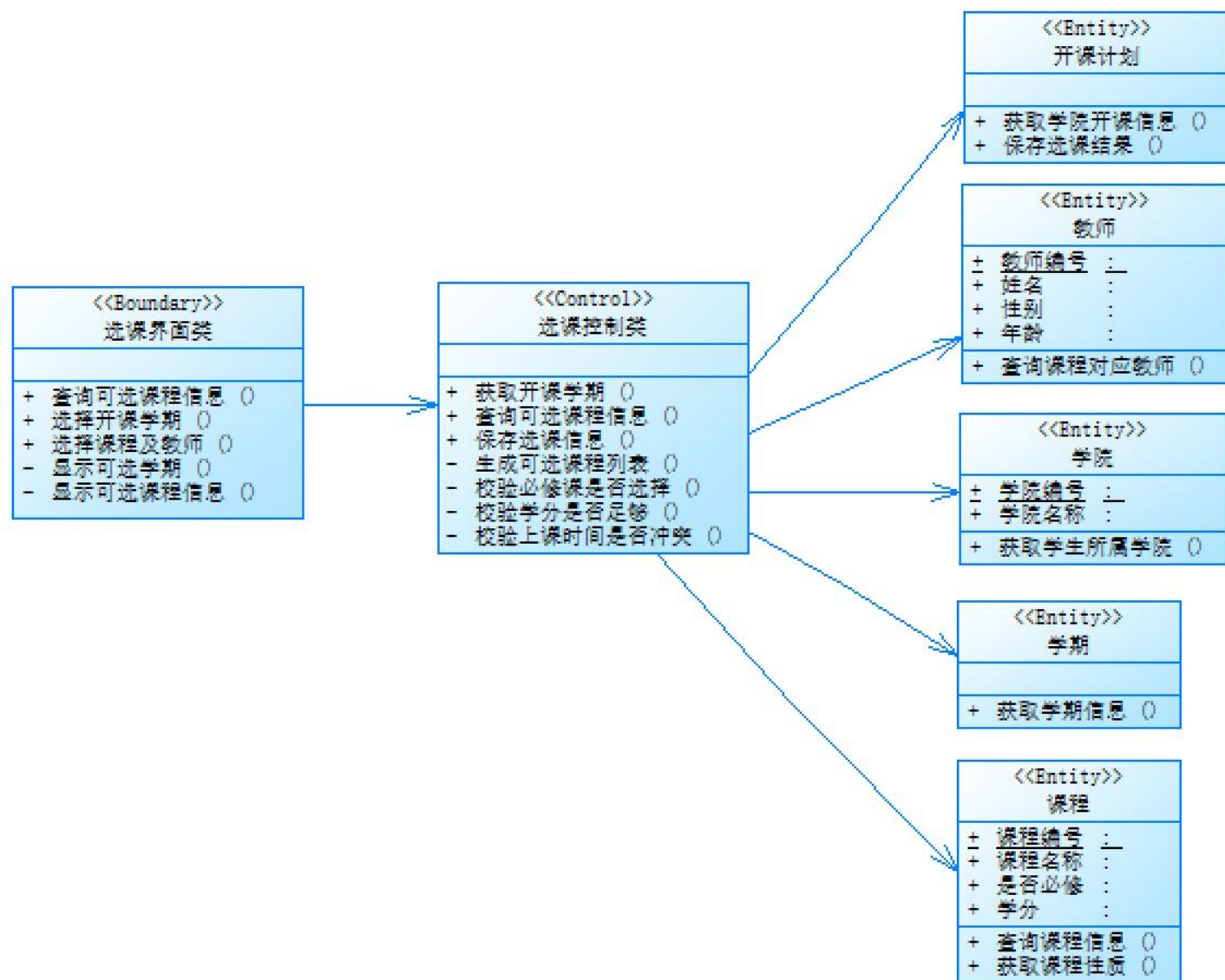
- 顺序图注意事项

- 1. 一般是调用消息时，对象才触发。
- 2. 返回消息代表着与调用消息是一组的，返回消息是虚线，而不是看方向；且返回消息只需画关键的，不用每个调用消息都画出其返回消息。
- 3. 非必要时，就使用简单消息，一般不用强调同步消息。

用例分析中的类图

- 分析类图
 - 边界类、控制类、实体类的职责，也即是类的操作，与用例交互分析中的消息调用的一一对应的。
 - 实体类的属性，是在分析过程中发现、演绎、推理得到的。
 - 边界类、控制类、实体类中的关系，一般以关联关系概要定义，在分析类图中一般不用定义多重性。
 - 同一个分析类，可能会在不同的用例交互分析中出现，其相关的用例分析全部完成，就可以代表该分析类分析完整了。

用例分析中的类图



本次作业：

- 1、以“选课系统”为例，参考“**用例交互分析**”，绘制顺序图，对选课系统“注册”和“登录”两个用例进行分析。
2. 参考“**用例分析中的类图**”，将顺序图中的边界类、控制类、实体类整合到分析类图（参考上一页ppt，可做适当修改和裁剪）。

