

4.3 类、数据建模与设计

- CRC卡片分拣法-面向对象方法

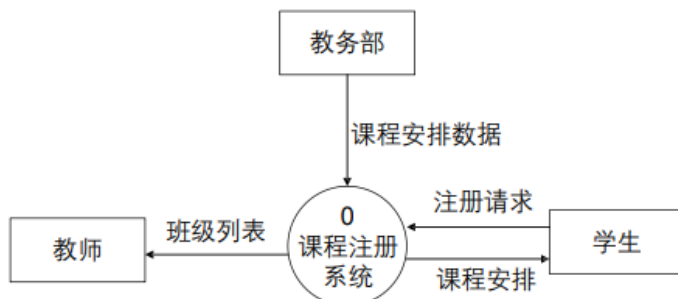
从问题域中找到合适的对象并理解它们之间的关系

- Class-Responsibility-Collaboration
 - 根据**边界类**、**控制类**和**实体类**的划分来发现系统中的类
 - 对邻域进行分析，或利用已有的**邻域分析结果**得到类
 - 参考**分析**、**设计模式**来确定类
- CRC 卡片
- 21 点游戏举例
 - 识别对象类
 - 类筛选
 - 确定初始类
 - 识别类的功能职责
 - 识别类交互协作关系
 - 类定义

- DFD-结构化方法 (Data Flow Diagram)

- 主要元素
 - 数据流
 - 加工
 - | 应有名字（动词短语）以及编号
 - 数据存储
 - 外部实体
- 层次性

- 环境关联DFD-顶层DFD

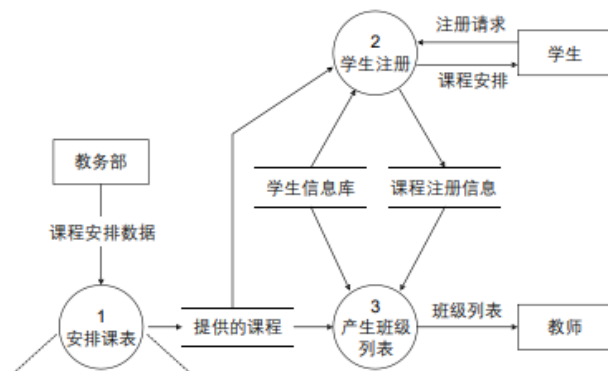


- 描述了系统与外部环境之间的数据输入/输出关系
- 通过系统和外部世界之间的联系来描述系统的范围
- 确定了通过某一接口与系统相连的外部实体，同时也确定了外部实体和系统之间的数据流
- **只包含一个加工**，用以表示被开发的系统，考虑输入、输出数据流

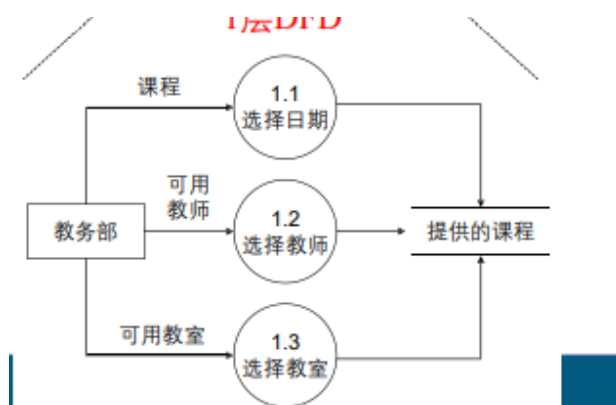
- 加工编号：0

- 系统内部DFD

- 0层DFD

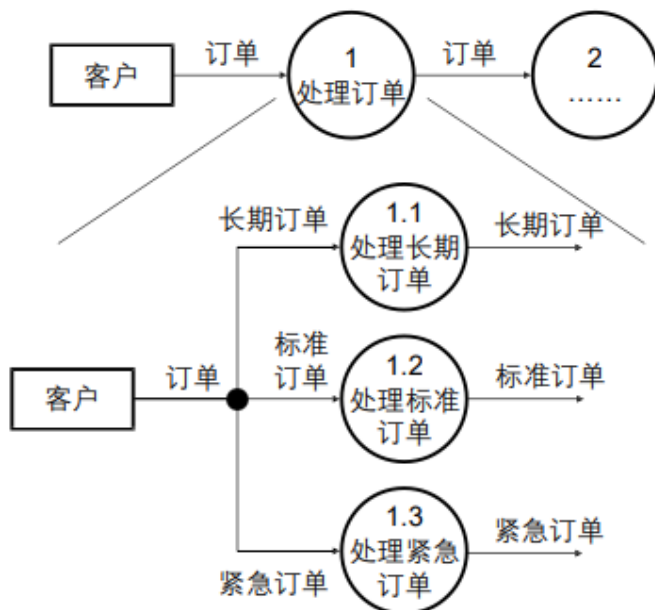


- 1层DFD



- ...

- 数据流的分解



- 绘制 DFD 的基本原则

- 数据存储放在 0 层数据流程图或更低层次子图上，不要放在顶层的关联图上
- 使用数据流程图时，不要视图让数据流程图反映处理的顺序，忽略系统运行时的时间特性
- 加工通过数据存储进行通讯，**尽量避免**从一个过程直接流到另一个过程

- 数据不能
 - 由一个**数据存储**直接流到另一个**数据存储**
 - 从一个**外部实体**直接流到一个**数据存储**
 - 从一个**数据存储**直接流到一个**外部实体**
 - 在**外部实体**之间流动
- 数据流是单向的
- 任何加工必须有输入和输出数据流
- 对现有加工进行持续的分解和组合，直到所有加工之间达到较高的聚合度
- 尽量将每一张DFD上所有元素数目控制在7-12个