

《系统分析与设计》

System Analysis and Design

任课教师： 范 国 祥

电 话： 0451-86418876-811(O)
13199561265(微信同号)

邮 箱： fgx@hit.edu.cn

哈工大计算学部/
国家示范性软件学院
软件工程教研室

2022.03



本章主要内容

1. 结构化方法vs面向对象方法
2. 结构化的系统分析方法
3. 数据流图（DFD）
4. 数据字典（DD）
5. 数据分析（ERD、IDEF1X）



本章主要内容

1. 结构化方法vs面向对象方法
2. 结构化的系统分析方法
3. 数据流图（DFD） 结构化系统分析核心 数据和处理之间的关系
4. 数据字典（DD）
5. 数据分析（ERD、IDEF1X）

软件工程方法：结构化VS面向对象

结构化方法

- 复杂世界 → 复杂处理过程（事情的发生发展）
- 设计一系列功能（或算法）以解决若干问题
- 寻找适当的方法存储数据

面向对象方法

- 任何系统都是由能够完成一组相关任务的对象构成
- 如果对象依赖于一个不属于它负责的任务，那么就需要访问负责此任务的另一个对象（调用其他对象的方法）
- 一个对象不能直接操作另一个对象内部的数据，它也不能使其它对象直接访问自己的数据
- 所有的交流都必须通过方法调用

软件工程方法：结构化VS面向对象

举例：五子棋游戏

- 面向过程（事件）的设计思路是首先分析问题的步骤：

1. 开始游戏，初始化画面（棋盘）
2. 黑子走，绘制画面（棋盘）
3. 判断输赢，如分出输赢，跳至步骤6
4. 白子走，绘制画面（棋盘）
5. 判断输赢，如未分出输赢，返回步骤2
6. 输出最后结果

- 面向对象的设计思路是分析与问题有关的对象：

1. 玩家：黑白双方，这两方的行为是一模一样的
2. 棋盘：负责绘制画面
3. 规则：负责判定诸如犯规、输赢等

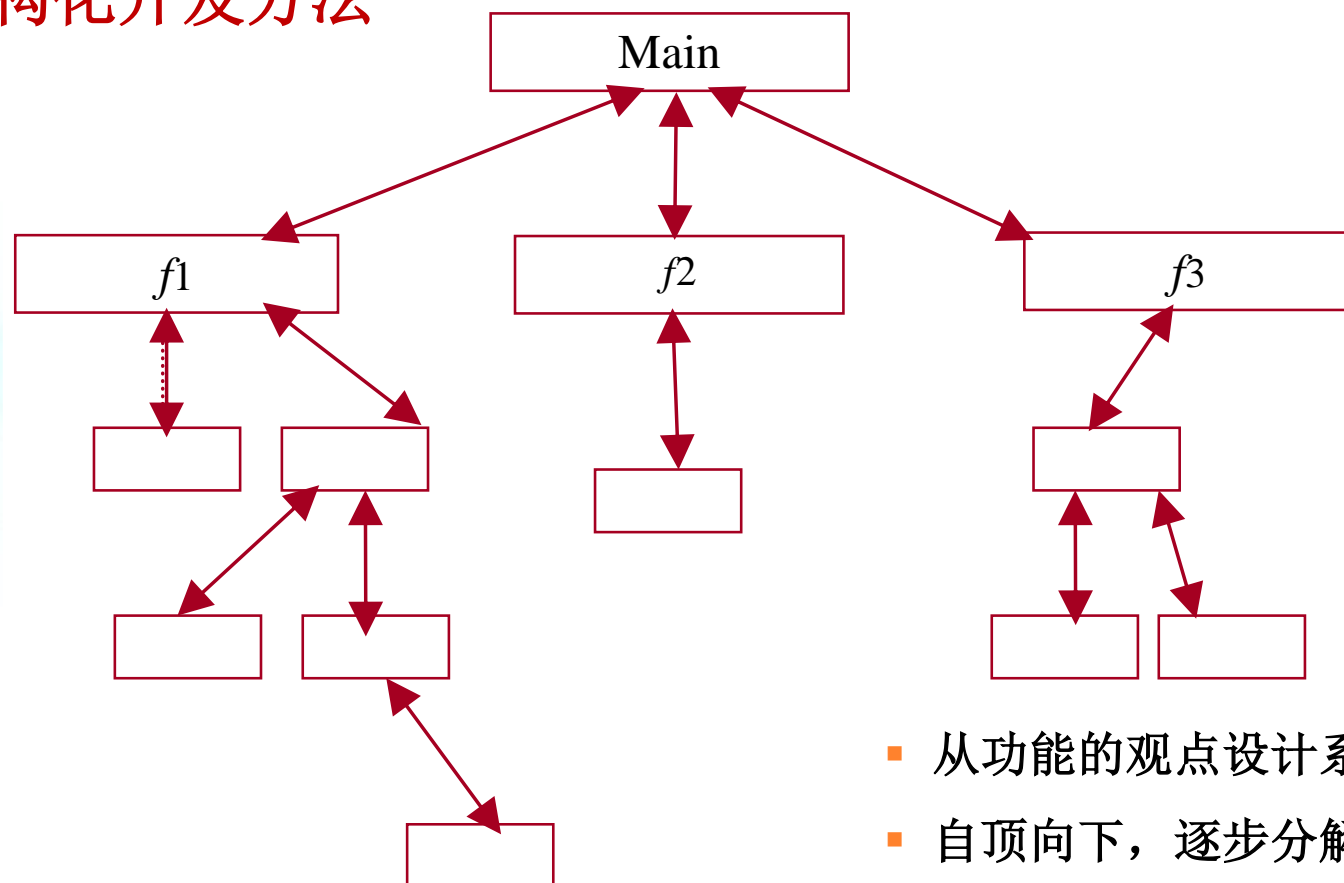
软件工程方法：结构化VS面向对象

结构化开发方法

- **起源时间：**20世纪60年代产生软件危机，70年代出现“结构化开发方法”
- **思想方法：**自顶向下、问题分解、分而治之、由分到合；
软件/程序 = 数据结构 + 算法，软件结构化、模块化、层次化
- **表达模型：**分析模型（系统流程图、数据流图、ERD、数据字典）
设计模型（系统结构图、程序流程图、ERD）
- **分析与设计线索：**
 - 面向过程（处理） – 过程驱动
 - 面向数据 – 数据驱动
- **优点：**思维自然，符合人们思考问题的方式；总体可控性强；适合偏重数学计算方面的项目
- **缺点：**不容易描述客观世界的需求；耦合性相对高

软件工程方法：结构化VS面向对象

结构化开发方法



- 从功能的观点设计系统
- 自顶向下，逐步分解和细化
- 将大系统分解为若干模块，主程序调用这些模块实现完整的系统功能

软件工程方法：结构化VS面向对象

面向对象开发方法

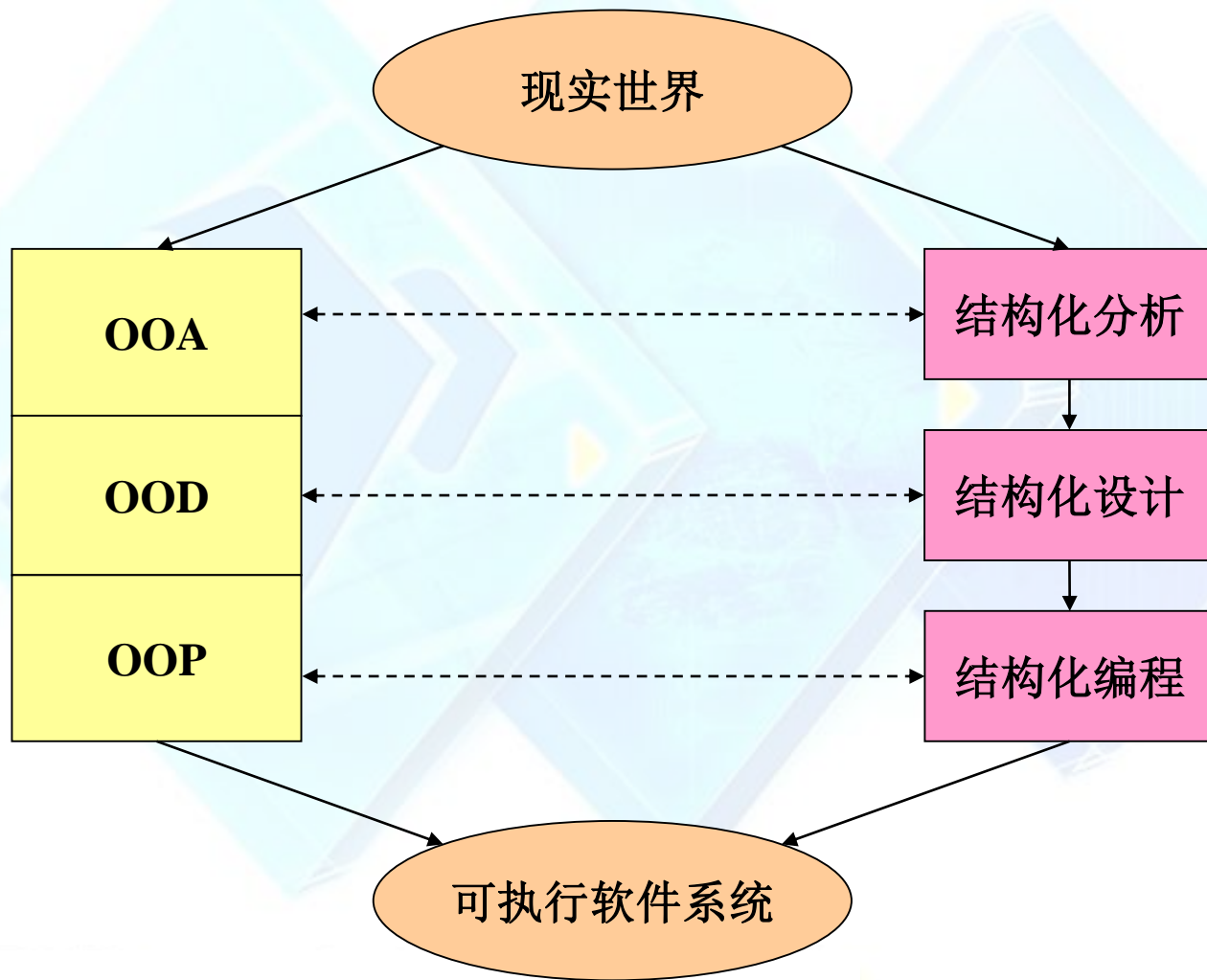
- 起源时间：20世纪80年代（70年代开始OOP）
- 思想方法：从客观世界的具体事物出发构建系统；
自底向上，先考虑“对象”，再考虑“关系”
- 表达模型：类层次结构图（对象的属性、行为、继承、消息连接等）
- 优点：符合自然世界的状态，描述自然，思维简单
适合复杂的事务处理、大量信息处理类的项目
耦合性容易降低，容易复用
- 缺点：OO模型表达系统需求分析和设计不够充分

软件工程方法：结构化VS面向对象

基于UML的面向对象开发方法

- 起源时间：20世纪90年代末
- 思想方法：开发一整套系统分析与设计模型及描述方法，从形式上和概念上统一描述软件系统
- 表达模型：用例图、活动图、状态图、序列图（协作图）、类图、组件图、包图、部署图等
- 优点：统一标准；容易交流；建模工具较多；适合大型信息系统开发
- 缺点：模型不容易掌握

软件工程方法：结构化VS面向对象





本章主要内容

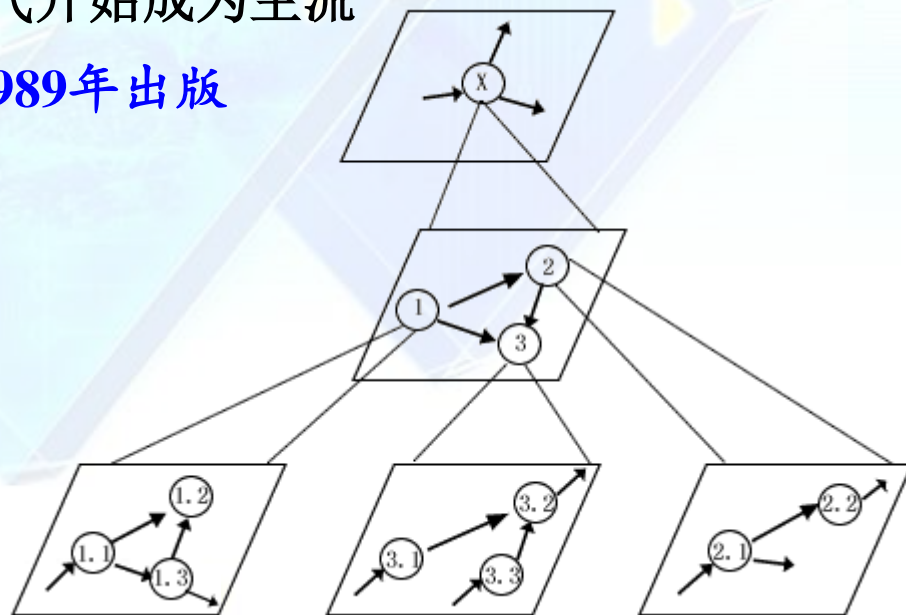
1. 结构化方法vs面向对象方法
2. 结构化的系统分析方法
3. 数据流图（DFD）
4. 数据字典（DD）
5. 数据分析（ERD、IDEF1X）

结构化的系统分析方法

- 结构化分析方法(SA): 将待解决的问题看作一个系统, 从而用系统科学的思想方法(抽象、分解、模块化)来分析和解决问题
 - 起源于结构化程序设计语言(事先设计好每一个具体的功能模块, 然后将这些设计好的模块组装成一个软件系统)
 - 以动词性的“功能”为核心展开分解
- 最早产生于1970年代中期, 1980年代开始成为主流
 - Yourdon (美国软件工程师) 于1989年出版《Modern Structured Analysis》



- 核心思想:
 - 自顶向下的分解(top-down)

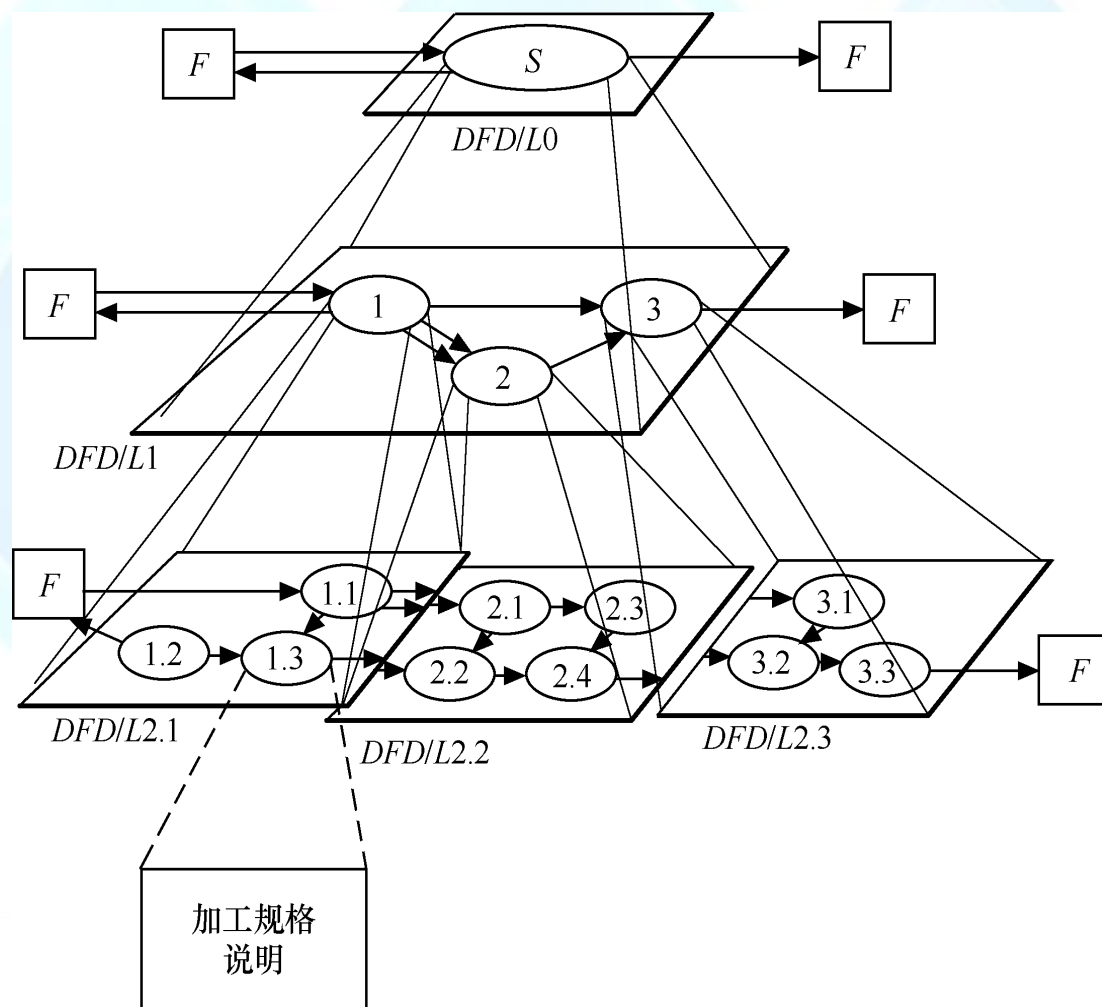


结构化的系统分析方法

- 结构化分析：帮助开发人员定义系统需要做什么（处理需求），系统需要存储和使用哪些数据（数据需求），系统需要什么样的输入和输出以及如何把这些功能结合在一起完成任务
 - 数据流图（**DFD**图）
 - 实体-关系图（**ERD**，**IDEF1X**图）

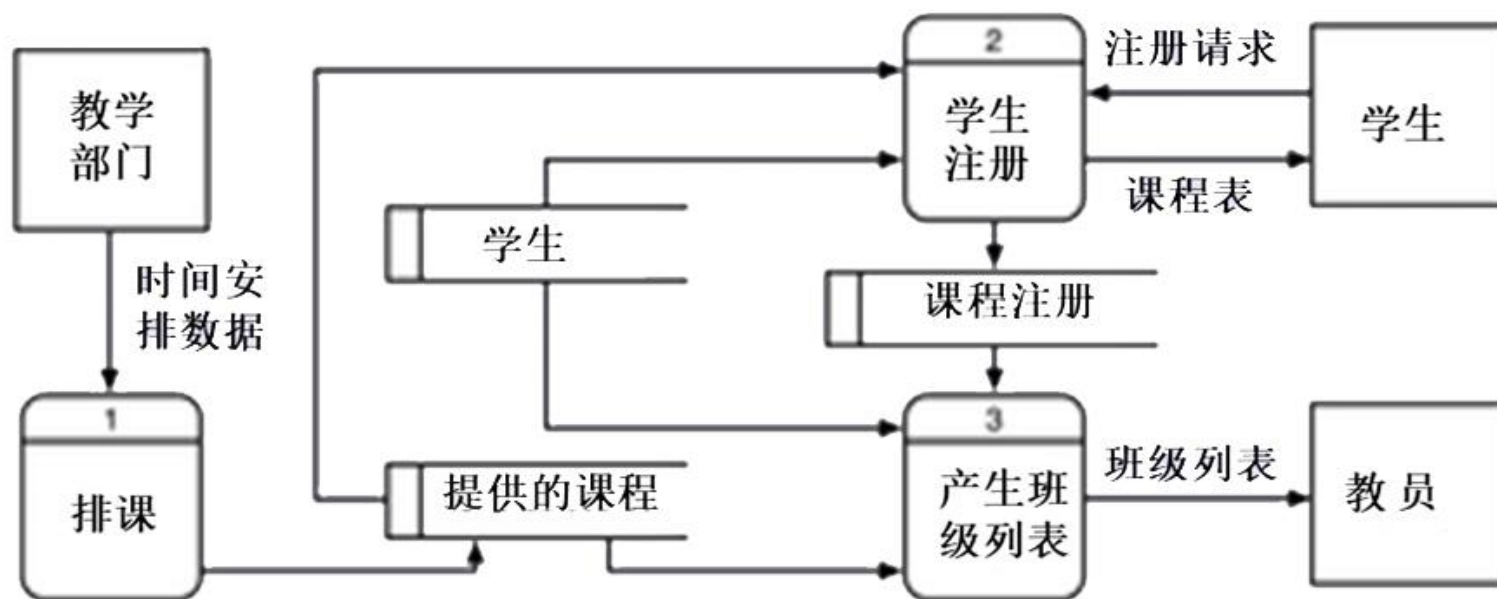
结构化的系统分析方法

基于数据流的需求分析建模 -- DFD



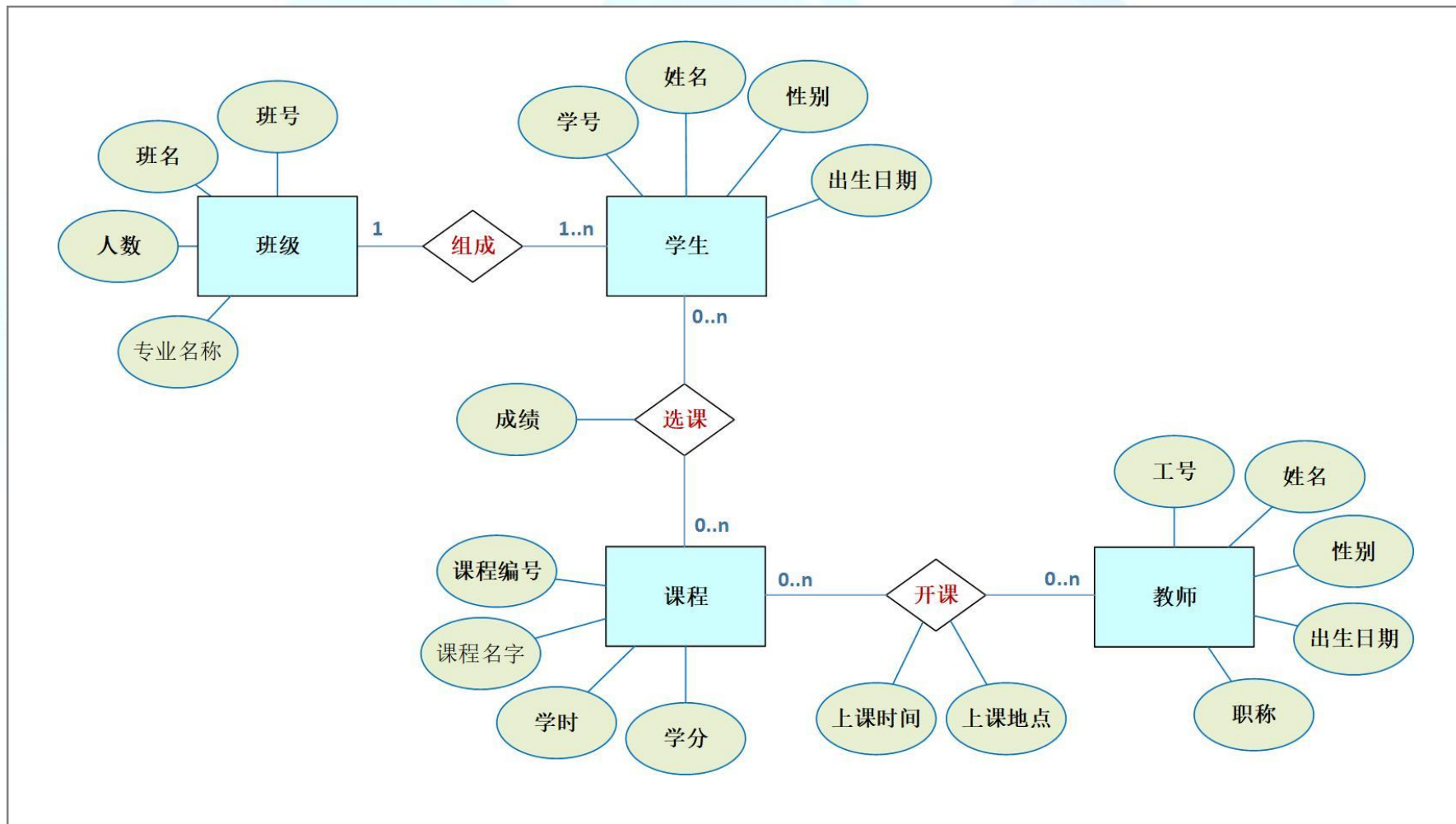
结构化的系统分析方法

基于数据流的需求分析建模 -- DFD



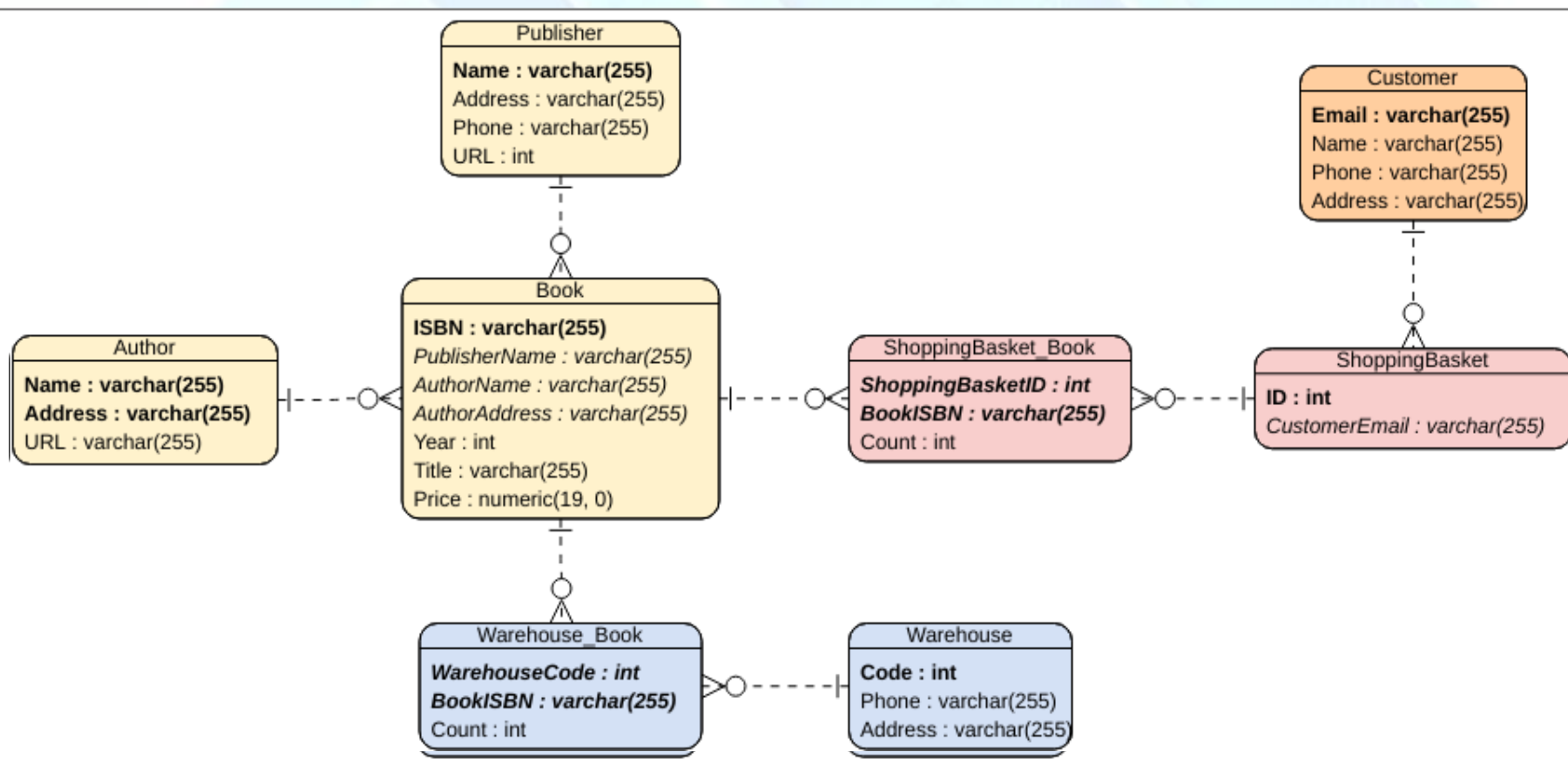
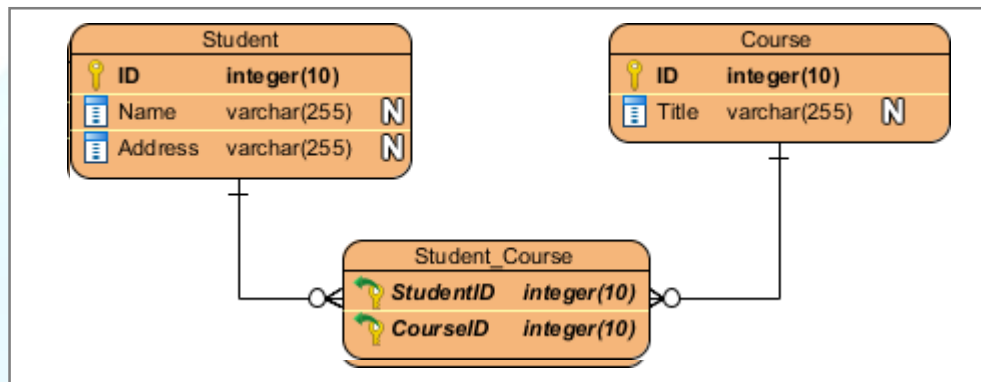
结构化的系统分析方法

数据分析建模 -- ERD



结构化的系统分析方法

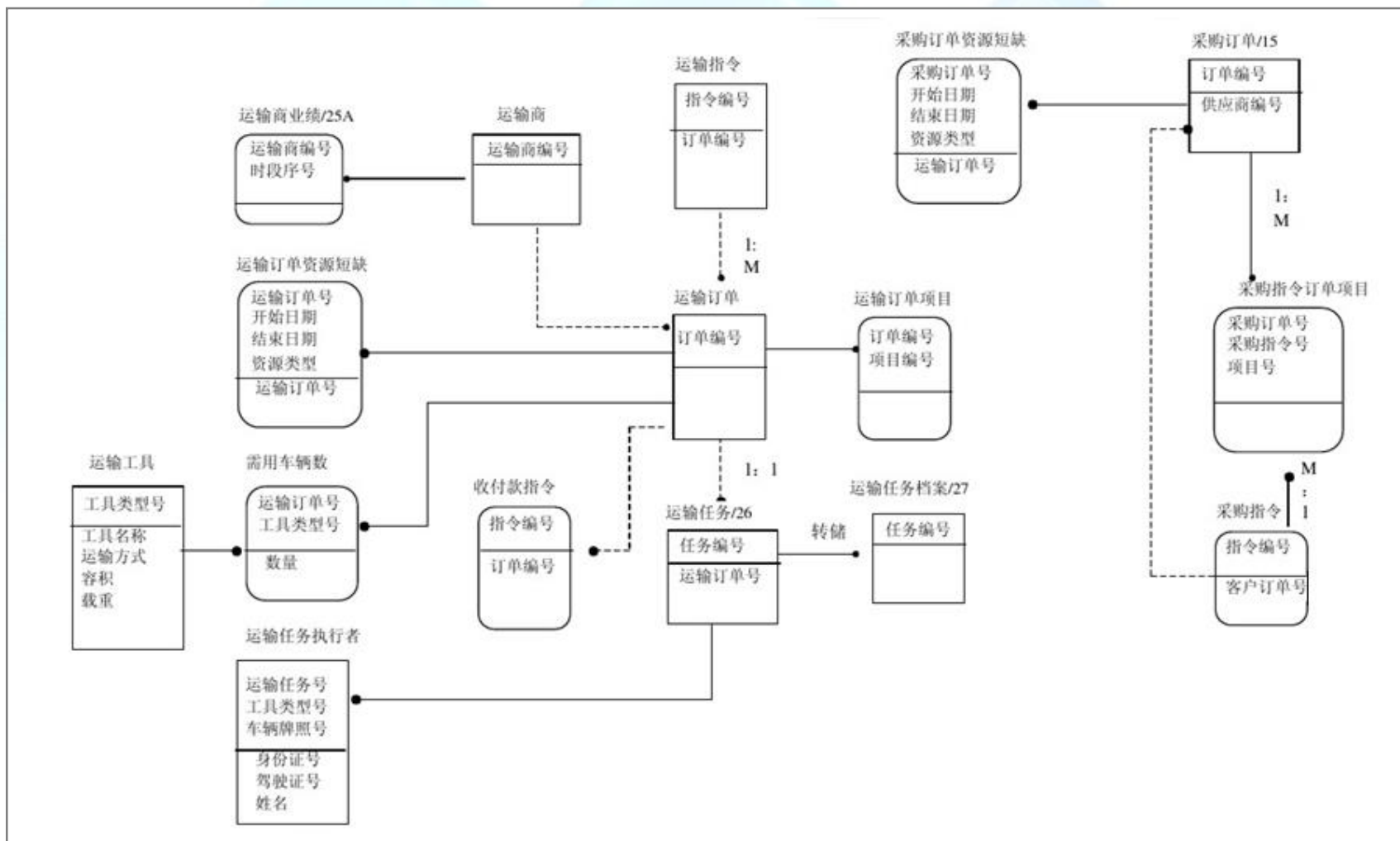
数据分析建模 -- ERD



采用“乌鸦脚 (Crow's Foot)”形式的 ERD

结构化的系统分析方法

数据分析建模 – IDEF1X图





本章主要内容

1. 结构化方法vs面向对象方法
2. 结构化的系统分析方法
3. 数据流图（DFD）
4. 数据字典（DD）
5. 数据分析（ERD、IDEF1X）

DFD建模 - 基于数据流的系统分析

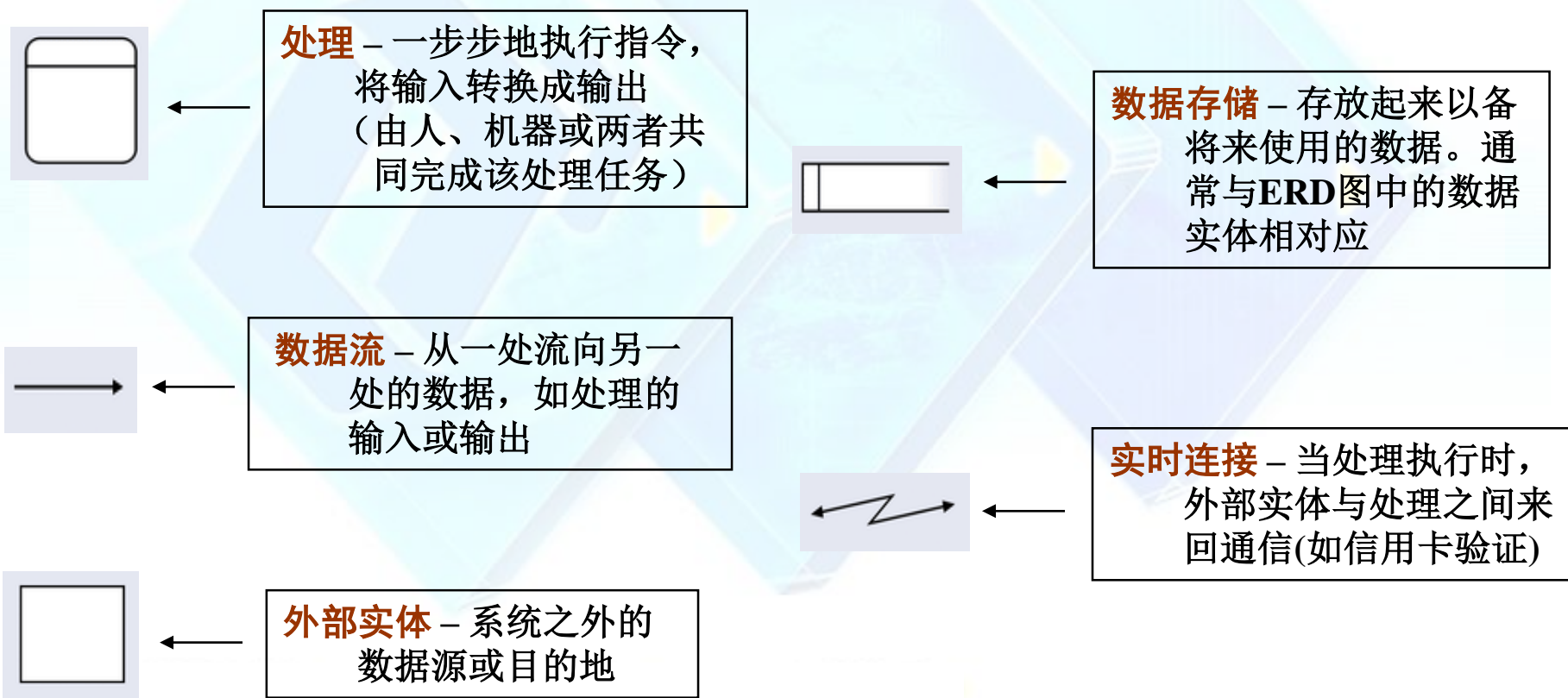
数据流图：用处理、外部实体、数据流以及数据存储来表示系统需求的图表

DFD的特点

- 图形元素少且符号简单易懂
- 较充分表达系统的主要需求：输入、输出、处理和数据存储
- 最终用户、管理人员和系统开发人员只需稍加培训即可读懂DFD图，方便交流

DFD建模 - 基于数据流的系统分析

DFD数据流图的符号说明

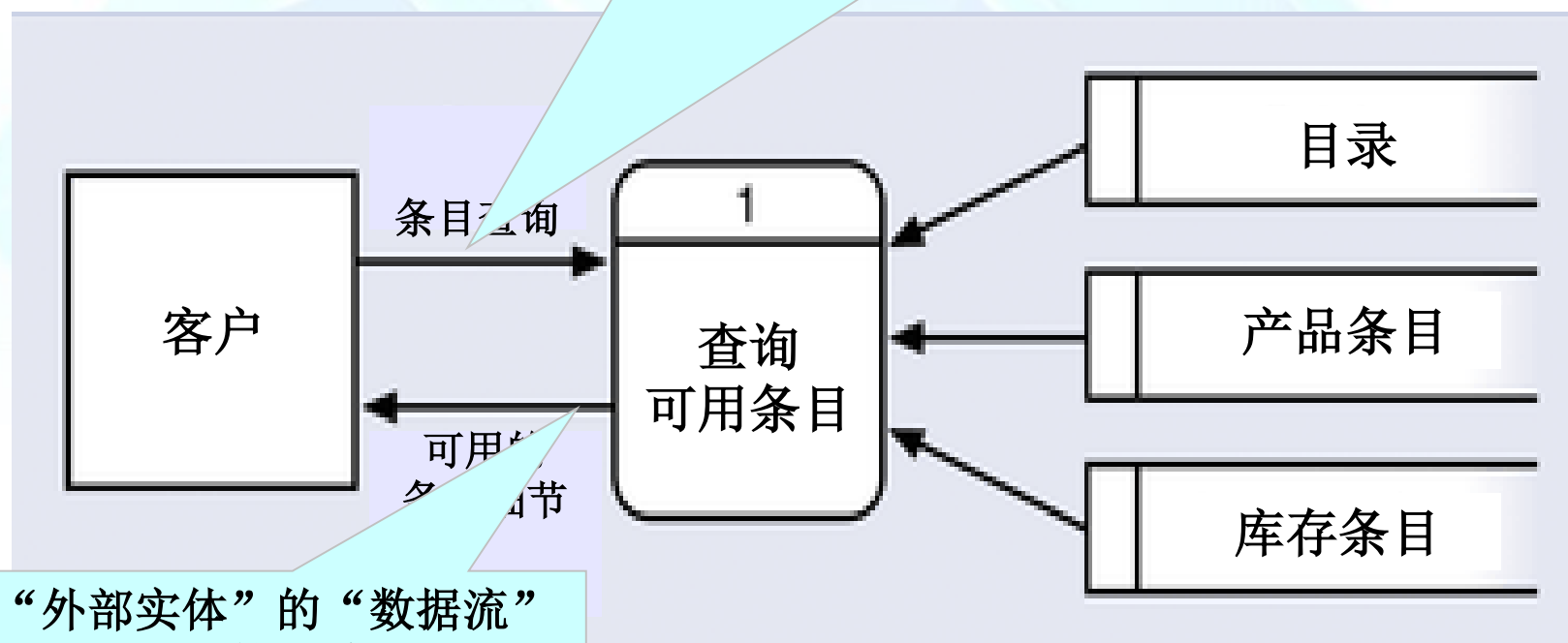


DFD建模 - 基于数据流的系统分析

数据流图例子:

从外部实体出发的“数据流”可以是:

1. 外部实体发出的“处理请求”，即一个事件
2. 外部实体给出的“输入数据”



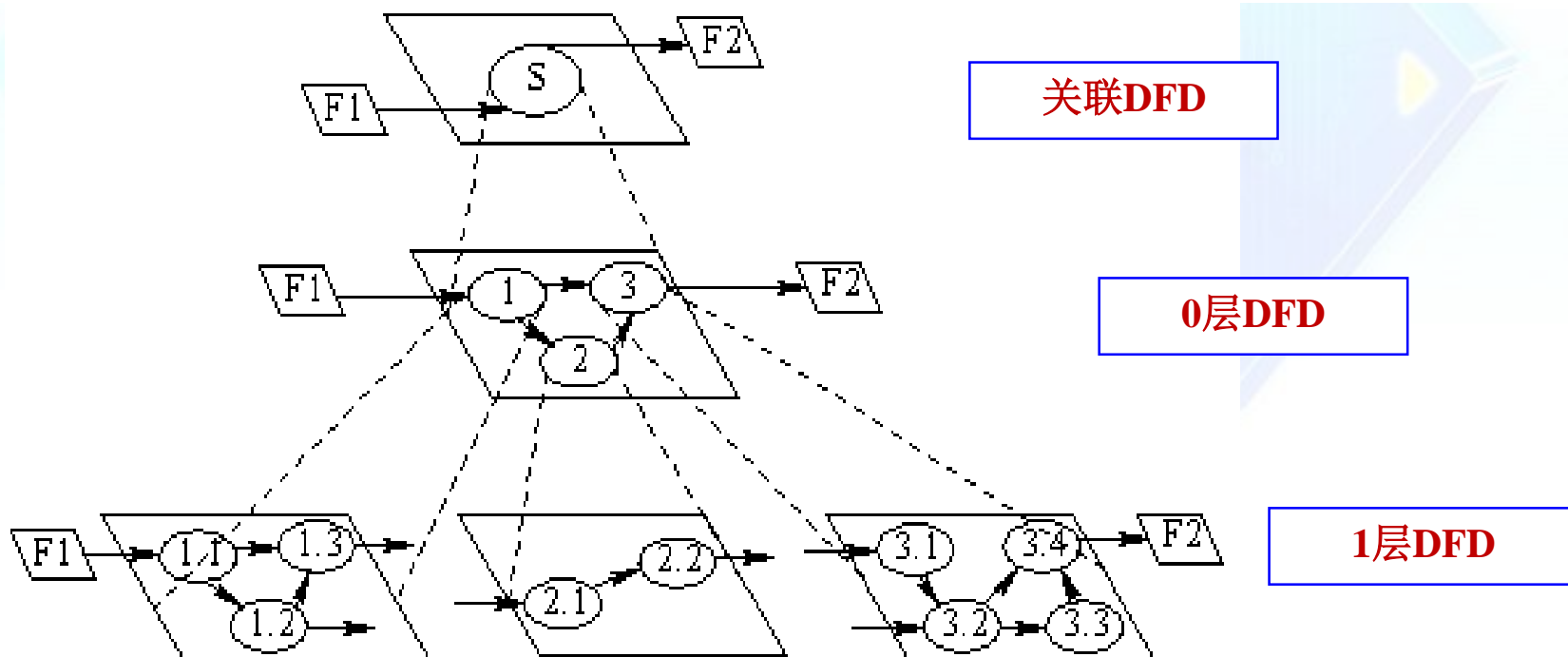
指向“外部实体”的“数据流”一般是“处理”的反馈或处理结果

显示处理“查询可用条目”的DFD

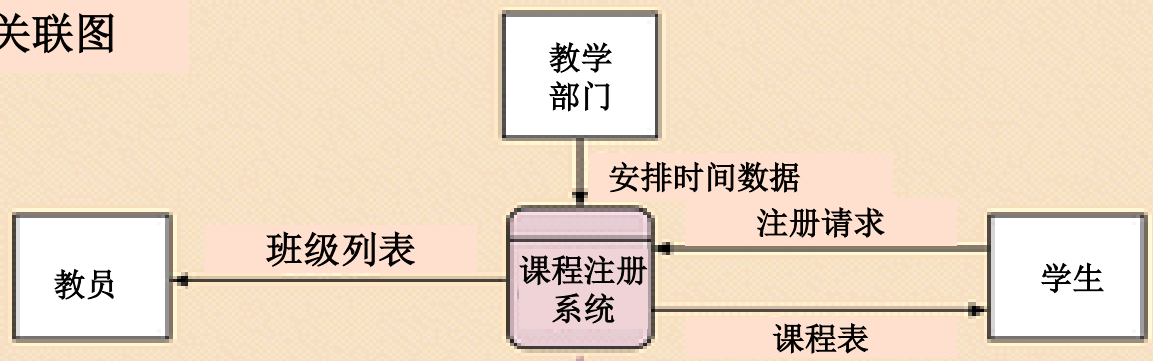
DFD建模 - 基于数据流的系统分析

DFD图可以描述高层次的具有高度概括的系统处理
也可以描述低层次的具有更详细分解的系统处理

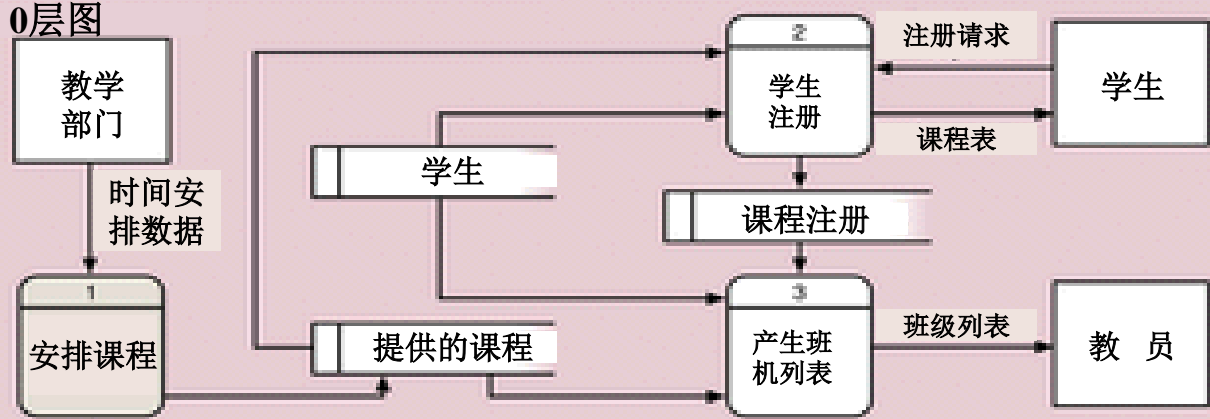
抽象层次：把系统分解成一个逐步细化的分层集合的建模技术



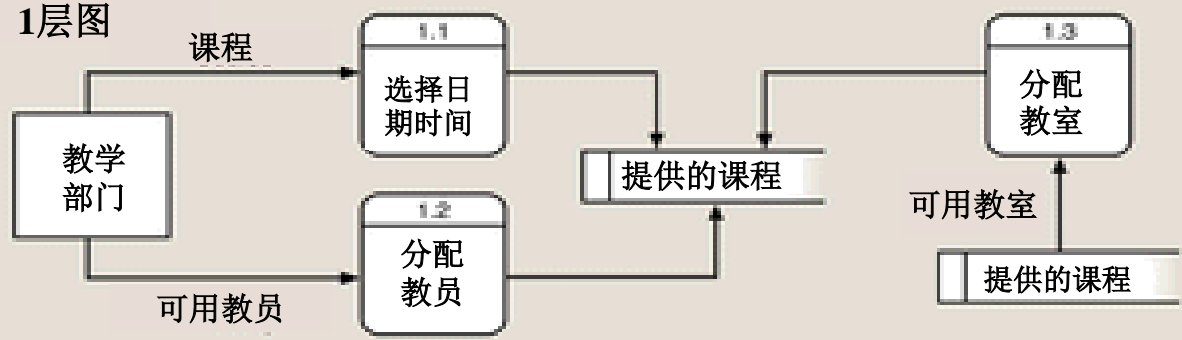
关联图



0层图



1层图



课程注册系统的DFD抽象层次

DFD建模 - 基于数据流的系统分析

关联DFD图

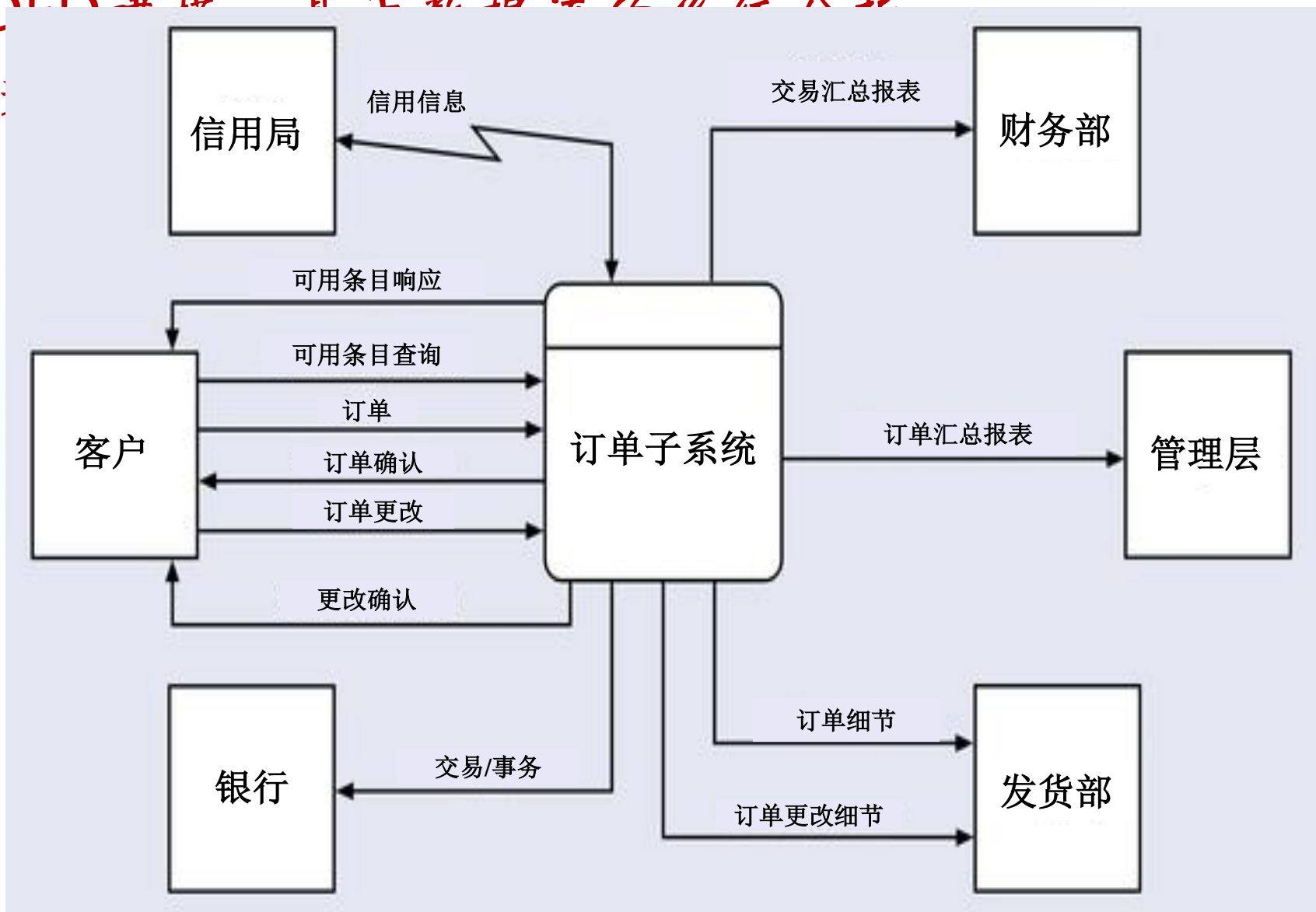
关联图：在单个处理符号中概括系统内所有处理活动的DFD

关联图在表达系统边界时用处很大

系统的范围是通过单个的处理和外部实体所表示的事物来定义的
数据存储不画在关联图中是因为它本身被认为是系统内部的内容

当一个系统响应事件较多时，常常将系统分成多个子系统，并为每个子系统创建一张关联图

DFD图及其与数据流图的关系



订单子系统的关联DFD图

DFD建模 - 基于数据流的系统分析

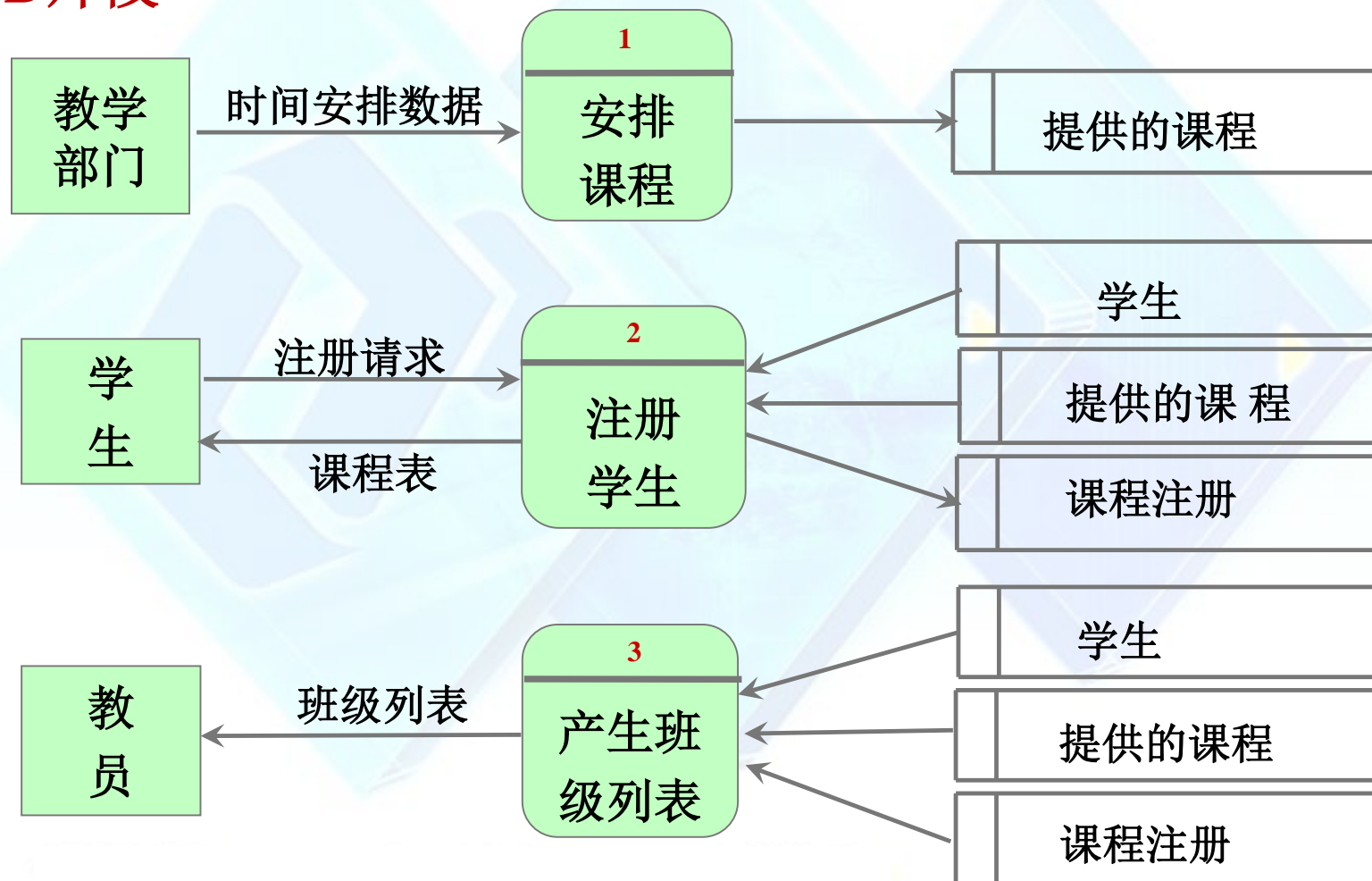
DFD片段

DFD片段：用一个单一处理符号表示系统响应一个事件的**DFD**

- 在**DFD**片段中，展示了处理、外部实体和内部数据存储之间的交互细节
- 每个**DFD**片段仅显示要响应该事件的相关的那些数据存储
- 一个**DFD**片段是为事件表中的每一个事件创建的

DFD建模 - 基于数据流的系统分析

DFD片段



课程注册系统的DFD片段

哈工大计算学部/软件学院

DFD建模 - 基于数据流的系统分析

DFD的0层图

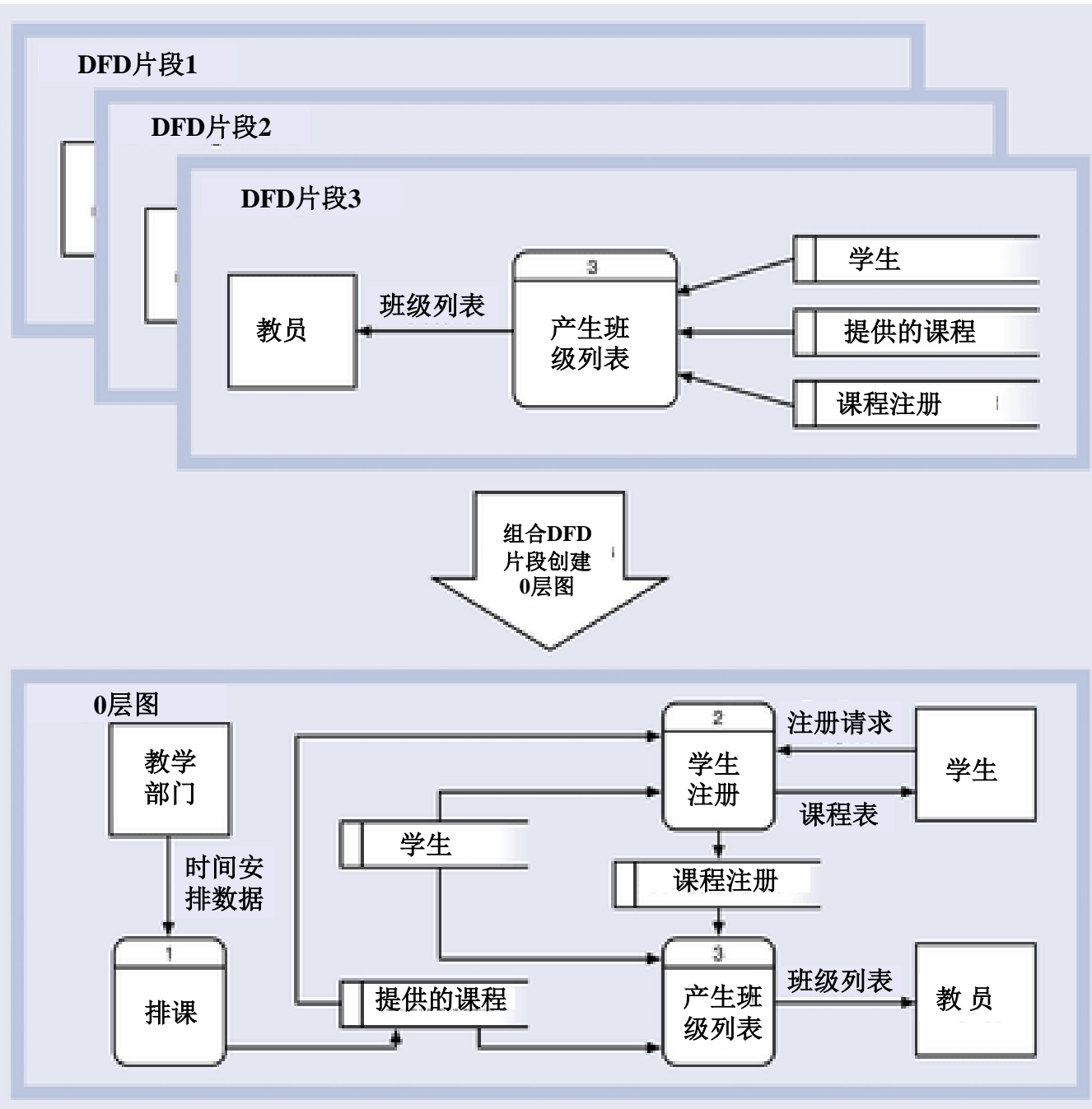
DFD的0层图：

将一个系统或子系统的所有DFD片段组合到一个单个的DFD图中，这样的DFD图称为**事件分离的系统模型/0层图**

0层DFD图：“处理”的编号为i

DFD建模

DFD的0层图



DFD建模 - 基于数据流的系统分析

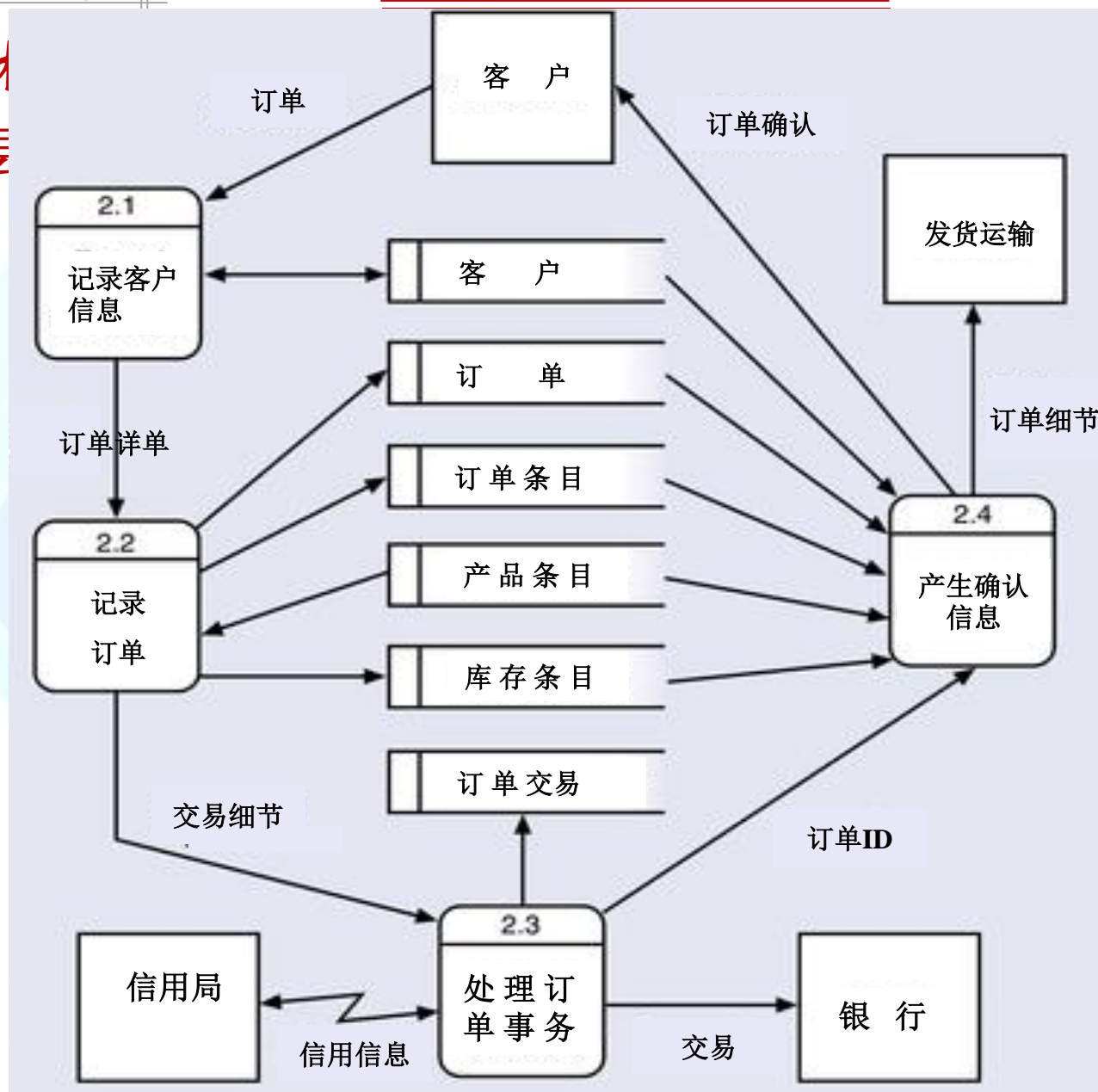
DFD的层次

分解“处理” -- 理解“处理”的细节：

一个DFD片段中的处理也可以包含多个更小的处理
有时分析员需要将该DFD片段进一步细化，生成更详细层次的DFD图

1层DFD图： 将0层DFD中的处理进一步细化等到的DFD图
“处理”的编号为“**i.j**”

2层DFD图： 将1层DFD中的处理进一步细化等到的DFD图
“处理”的编号为“**i.j.k**”

DFD 建立
DFD的层

DFD建模 - 基于数据流的系统分析

DFD的质量评估

高质量的DFD:

可读性强、内部一致、能够准确描述系统需求

措施:

- ◆ 最小化复杂度
- ◆ 保证数据流一致性

最小化复杂度:

就是使每幅DFD图尽量简单易懂, 避免信息超量

信息超量:

当太多的信息同时显现时所发生的难以理解的情况

DFD建模 - 基于数据流的系统分析

DFD的质量评估

最小化复杂度

措施：

采用分层结构将DFD划分为小的且相对独立的子集
这样可以逐级阅读、考察DFD

构造DFD图的 7 ± 2 规则：

- ◆ 单个DFD中不应有超过 7 ± 2 个处理
- ◆ 单个DFD中不应超过 7 ± 2 个数据流进出同一个处理/数据存储

接口最小化：

- ◆ DFD中各个元素之间的连接数越少越好

DFD建模 - 基于数据流的系统分析

DFD的质量评估

保证数据流一致性

数据流一致性表现在三个方面：

- ◆ 一个“处理”和该“处理”被详细分解后在数据流内容上应该一致
- ◆ 对一个“处理”，有数据流入则必须有相对应的数据流出
- ◆ 对一个“处理”，有数据流出则必须有相对应的数据流入

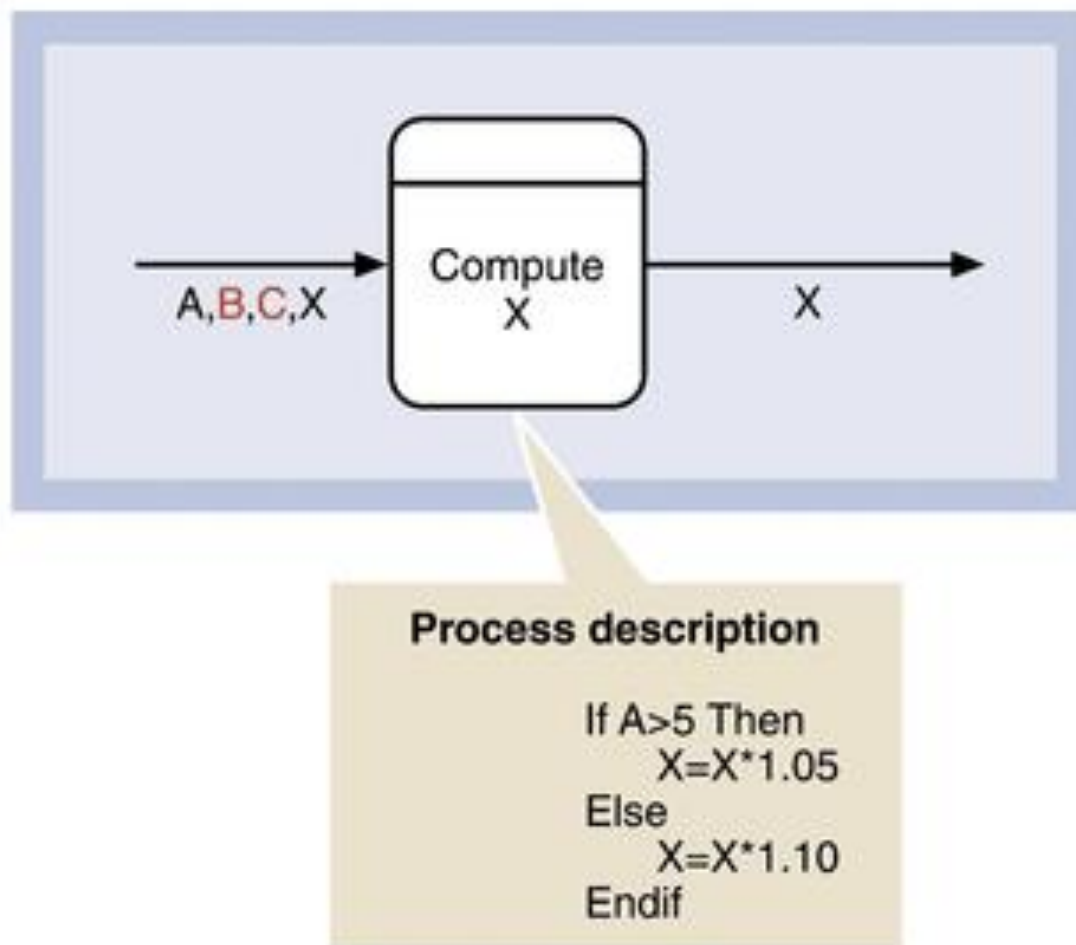
DFD建模 - 基于数据流的系统分析

DFD的质量评估

黑洞 --- 带有输入数据的但并不用其产生输出数据的
处理或数据存储

DFD建模 - 基于数据流的系统分析

DFD的质量评估



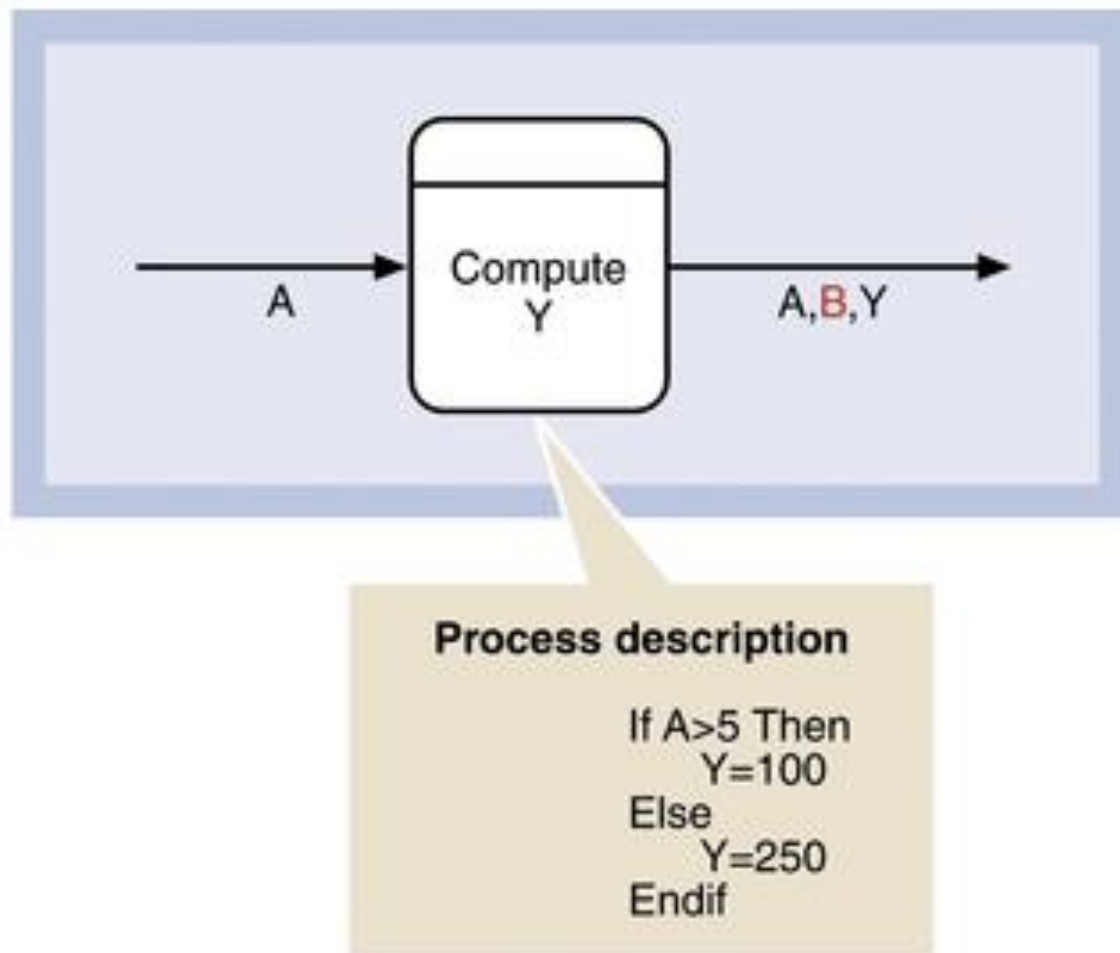
DFD建模 - 基于数据流的系统分析

DFD的质量评估

奇迹 --- 没有足够数据元素作为输入或产生来源的一个
处理或数据存储

DFD建模 - 基于数据流的系统分析

DFD的质量评估



有不可能的数据输出的处理——一个奇迹

哈工大计算学部/软件学院

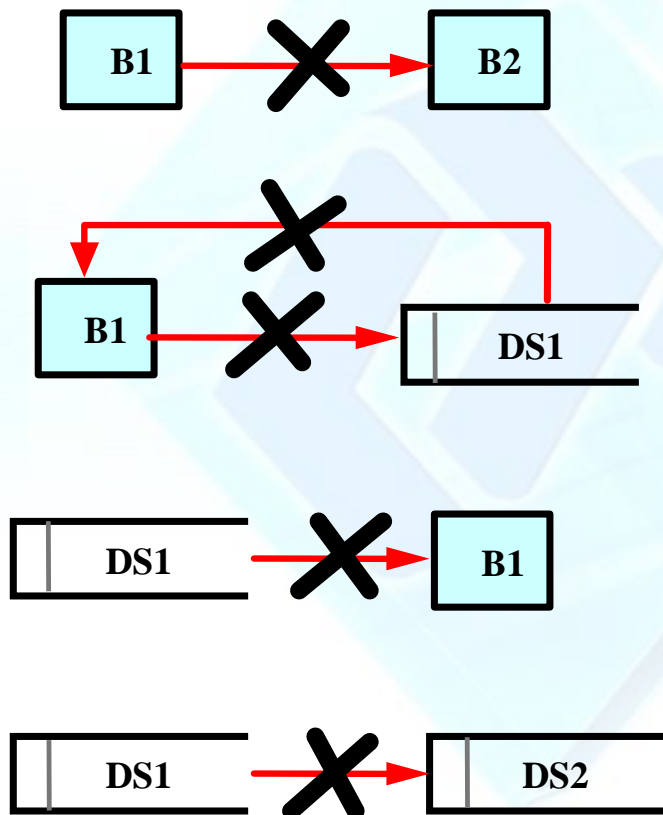
DFD建模 - 基于数据流的系统分析

DFD的质量评估

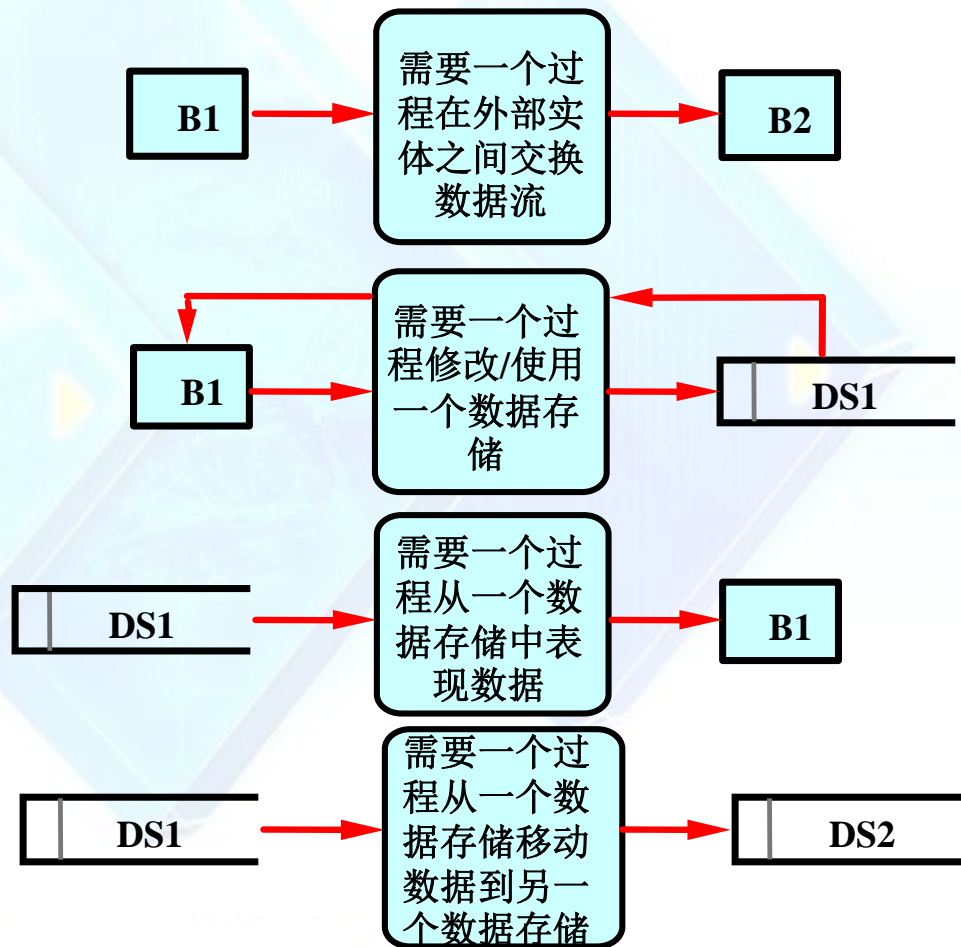
DFD图典型错误

DFD建模 - 基于数据流的系统分析

非法的数据流

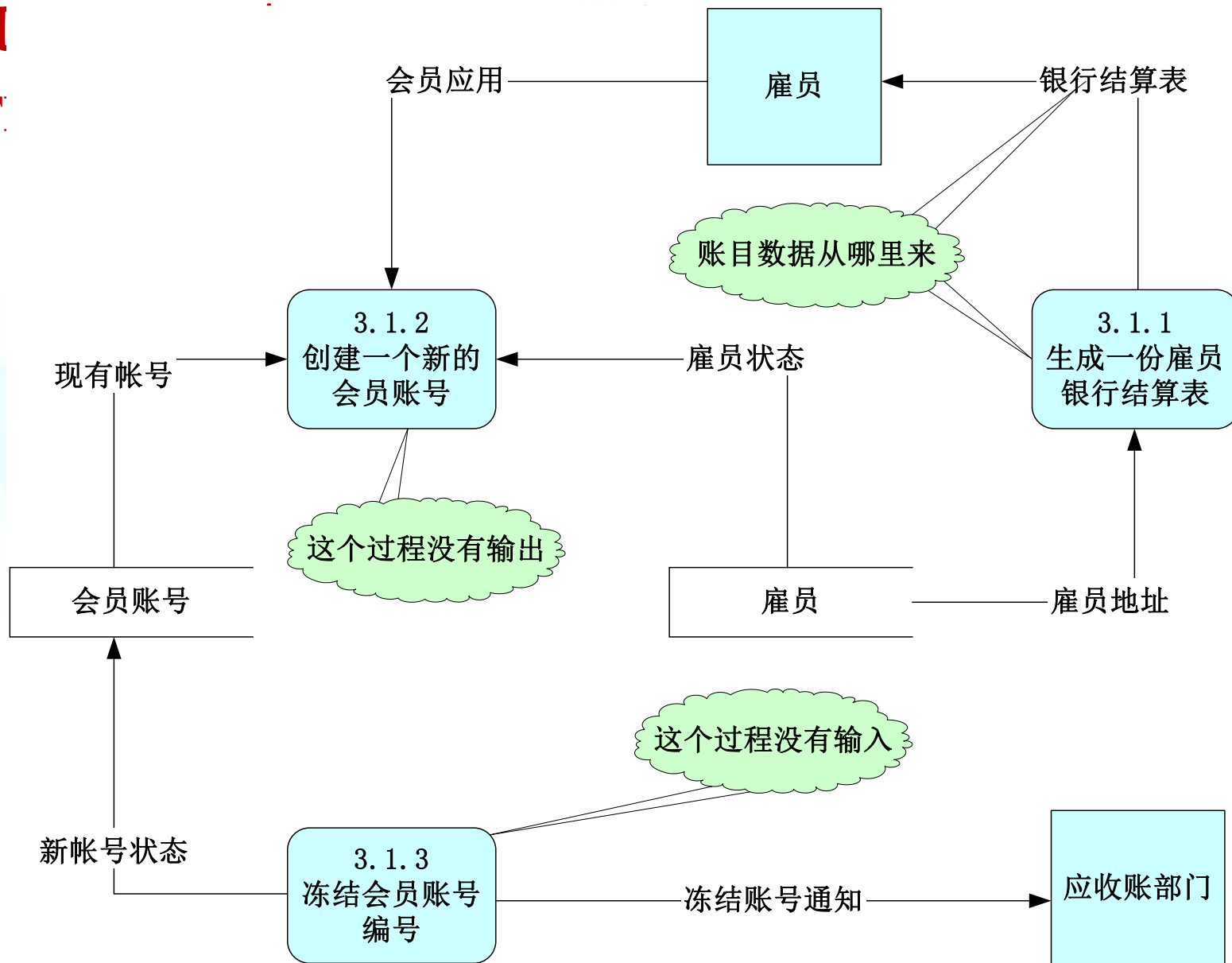


改正错误后的数据流



DFI

DF



DFD建模 - 基于数据流的系统分析

DFD模型总结

- ❖ 在关联图中不画数据存储
- ❖ 数据流不反映处理顺序，显示数据通过系统的流程，因此“处理”可以并行工作
- ❖ “处理/数据存储”既要有输入，又有输出
 - ◆ 若输入数据流不完全用来产生输出数据流，称之为黑洞
 - ◆ 若输出数据流不完全依赖于输入数据流，称之为奇迹

DFD建模 - 基于数据流的系统分析

DFD细节内容描述

在传统方法中，DFD图在一个图中描述了三种元素：

- 处理 -- 每个最底层“处理”需要详细描述其细节
- 数据流 -- “数据流”包含的数据元素需要定义
- 数据存储 -- “数据存储”需要在ERD中定义

使用“数据字典 – DD”来描述，见下一小节

DFD建模 - 基于数据流的系统分析

DFD细节内容描述 - 处理的细节描述

1. “处理” 细分解，层层分解，直到可详细描述细节
2. 结构化语言/伪代码
3. 决策表/决策树

“处理” 细分解的方法前面已经提到

DFD建模 - 基于数据流的系统分析

DFD细节内容描述描述 - 处理的细节描述

结构化语言/伪代码:

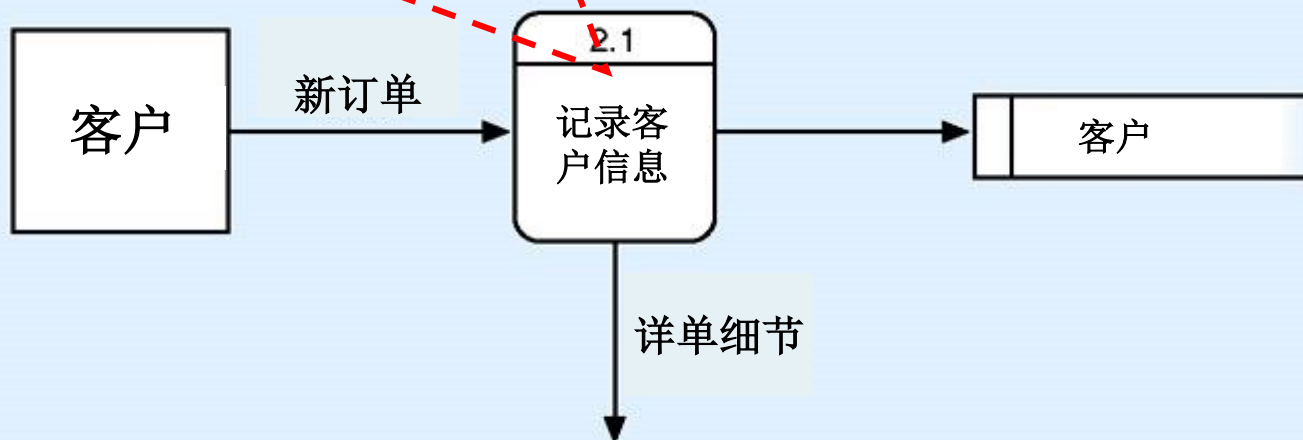
- 一种描述“处理”的规范方法，将结构化编程技术和叙述性英语结合起来
- 结构化英语适合用来描述处理步骤清楚、控制逻辑相对简单的情况

DFD

DFD

Process 2.1 - Record Customer Information

```
Ask if customer has an account (or has made a previous order)
If customer has an account then
    Ask for identification information
    Query database with identifying information
    Copy query response data to Order details
Else
    Create an empty Customer record in the database
    Ask customer for Customer attributes
    Update empty Customer record with Customer attributes
Endif
Ask customer for order information for first item
While more order items Do
    Update Order details with order information
Endwhile
```



DFD建模 - 基于数据流的系统分析

DFD细节内容描述描述 - 处理的细节描述

结构化语言的特点

- ◆ 介于自然语言和形式语言之间的语言
- ◆ 书写规则：
 - 使用短句
 - 多层缩进
 - 将结构化编程技术和叙述性语言结合
 - 无确定语法
 - 可分层、嵌套

DFD建模 - 基于数据流的系统分析

DFD细节内容描述描述 - 处理的细节描述

决策表和决策树

决策表：“处理”逻辑的一种表格形式的表示方法，其中包括决策变量、决策变量值、行为或公式

决策树：用树形结构组织起来的线条对“处理”逻辑进行图形化的描述

DFD建模 - 基于数据流的系统分析

DFD细节内容描述描述 - 处理的细节描述

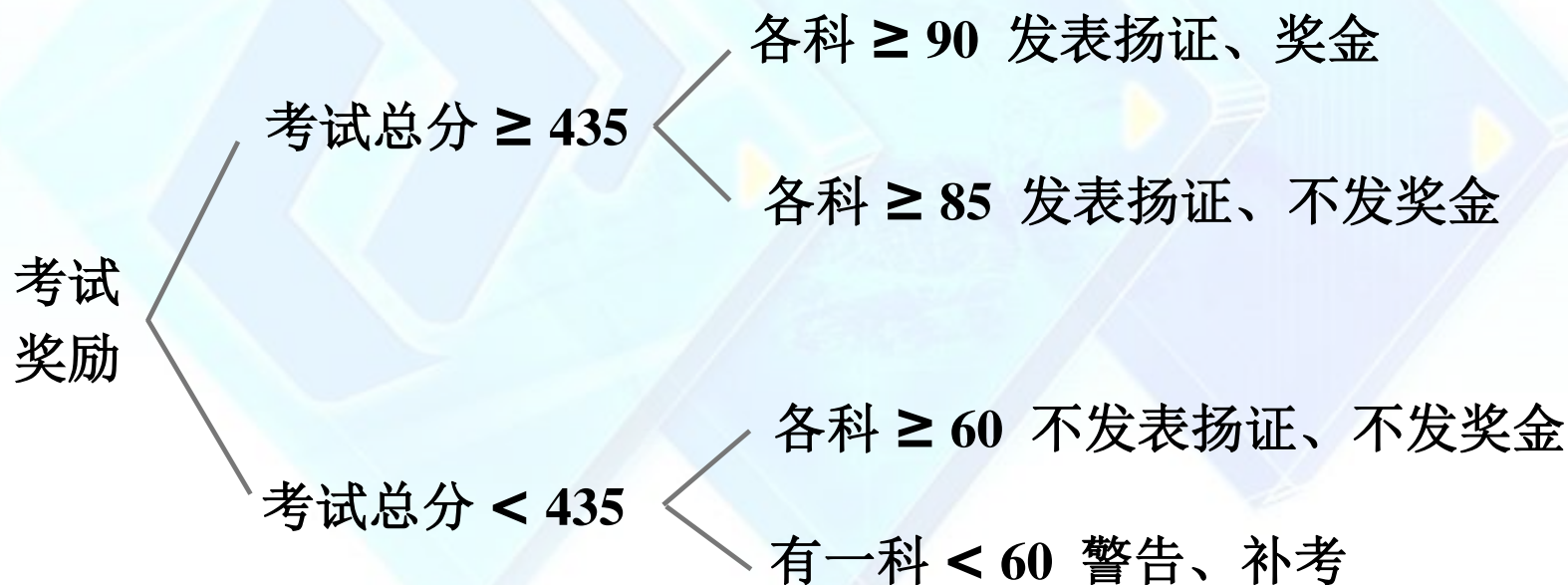
决策表

		规 则			
条 件	考试总分	≥ 435	≥ 435	< 435	< 435
	单科成绩	≥ 90	≥ 85	≥ 60	< 60
动 作	发表扬证	√	√	×	×
	发奖金	√	×	×	×
	警告	×	×	×	√

DFD建模 - 基于数据流的系统分析

DFD详细内容描述描述 - 处理的细节描述

决策树





本章主要内容

1. 结构化方法vs面向对象方法
2. 结构化的系统分析方法
3. 数据流图（DFD）
4. 数据字典（DD）
5. 数据分析（ERD、IDEF1X）

数据字典 (DD) - DFD 细节内容描述

数据字典是数据分析的描述模型，包括：

数据项定义，数据结构定义，数据流描述，数据存储描述

数据项定义： 定义特定数据项的组成和意义

数据结构定义： 定义有数据项组成的表达数据的基本数据

数据流描述： 描述数据流的数据构成，并指明其来源或去向

数据存储描述： 描述保存在存储介质上的数据文件或数据库表的格式和内容

广义的数据字典，也可以包含对“处理”的描述

数据字典 (DD) - DFD 细节内容描述

数据项定义

数据项：数据的基本单位

- 数据项名
- 数据项说明
- 数据类型
- 长度
- 取值范围
- 语义定义
- 与其他数据项的关联

数据字典 (DD) - DFD 细节内容描述

数据项定义

数据元素条目

名称：学号

说明：本校学生编号

数据值类型：离散

类型：数字

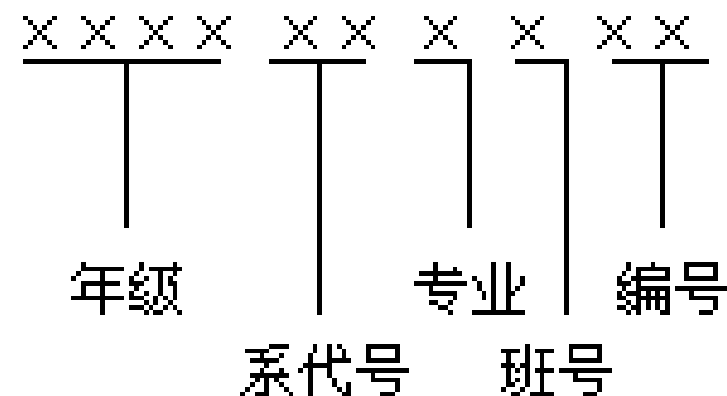
长度：10

有关数据结构：学生成绩
学生登记卡
选课卡

总编号：1—001

编号：001

有关编码说明：



数据字典 (DD) - DFD 细节内容描述

数据结构定义

数据结构：由数据项组成，它给出了数据基本结构单位

- 数据结构名
- 数据结构说明
- 数据结构组成：{ 数据项/数据结构 }
- 数据结构约束

数据字典 (DD) - DFD 细节内容描述

数据结构定义

数据结构条目

名称：学生登记卡

总编号：2-001

说明：新生入学时填写基本信息的卡片

编号：001

有关数据流，数据存储：

学籍表

结构：学号

姓名

性别

出生日期

入学日期

民族

家庭住址

联系电话

数据字典 (DD) - DFD 细节内容描述

数据流定义

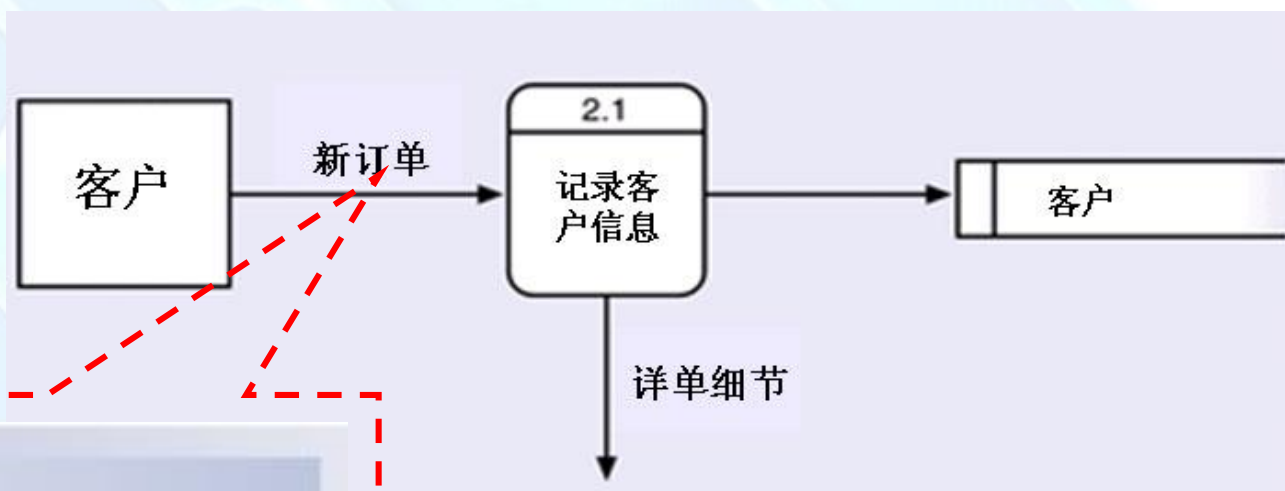
数据流：数据结构在系统中的流通过径

- 数据流名
- 数据流说明
- 数据流来源
- 数据流去向
- 数据流组成： { 数据结构 }
- 平均流量
- 高峰流量

数据字典 (DD) - DFD 细节内容描述

数据流定义

数据流是数据元素的集合，数据流定义就是列出其包含的所有数据项



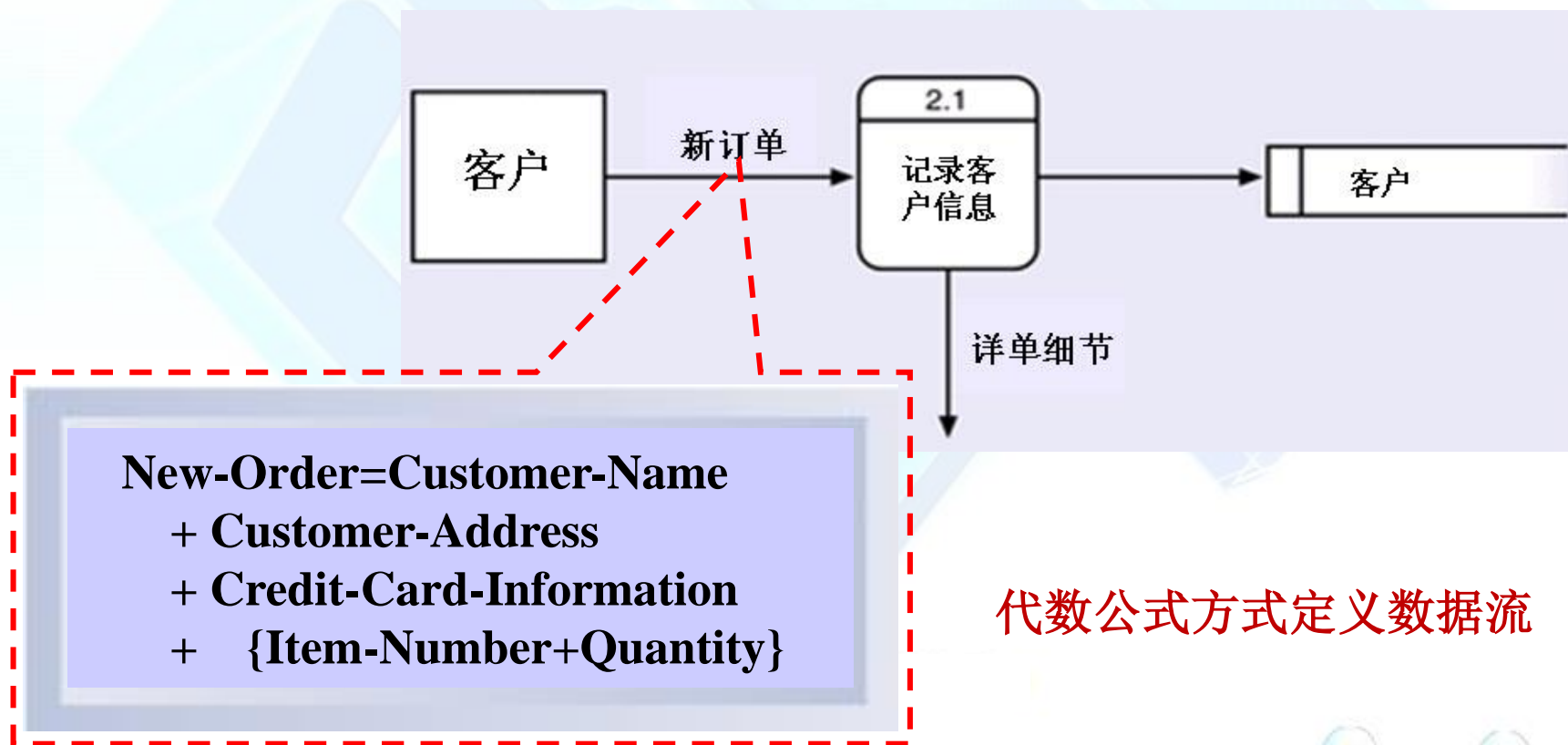
Customer-Name
Customer-Address
Credit-Card-Information
Item-Number
Quantity

数据元素还需进一步的定义：说明类型、长度等，一般在设计阶段详细定义

数据字典 (DD) - DFD 细节内容描述

数据流定义

数据流是数据元素的集合，数据流定义就是列出其包含的所有数据项



数据字典 (DD) - DFD 细节内容描述

数据流定义

数据流条目

名称：新生信息

总编号：3-001

说明：新生入学时带来的基本
信息

编号：001

流通量：1000 份/学期

数据流来源：招生办

数据流去向：P1.1

包含的数据结构：学号

姓名

性别

出生日期

入学日期

民族

数据字典 (DD) - DFD 细节内容描述

数据存储定义

数据存储：数据结构保存或停留之处，数据文件或数据库表

- 数据存储名
- 数据存储说明
- 输入的数据流
- 输出的数据流
- 数据存储组成： { 数据结构 }
- 数据量
- 存取频度
- 存取方式

数据字典 (DD) - DFD 细节内容描述

数据存储定义

数据存储条目

名称：学籍表

总编号：4- 001

说明：存储学生所有信息的记录

编号：001

结构：基本信息

有关数据流：

奖惩记录

P1.1 D1

学生动态

P1.2 D1

考试成绩

P2.1 D1

D1 P3.1

D1 P3.2

数据字典（DD） - DFD细节内容描述

数据处理（广义DD）

数据处理：给出处理的流程和说明信息

- 数据处理名
- 数据处理说明
- 输入数据： {数据结构}
- 输出数据： {数据结构}
- 处理过程简要描述

数据字典 (DD) - DFD 细节内容描述

数据处理 (广义DD)

处理过程条目

名称：信息输入

总编号：5-001

说明：输入新生的基本
信息

编号：P1.1

输入：招生办 P1.1

输出：P1.1 D1

处理：将由招生办来的学生登
记卡的内容作为基本信
息输入到学籍记录中。



本章主要内容

1. 结构化方法vs面向对象方法
2. 结构化的系统分析方法
3. 数据流图（DFD）
4. 数据字典（DD）
5. 数据分析（ERD、IDEF1X）

数据分析（ERD、IDEF1X）

实体关系图 – ERD

- ◆ 对DFD中的数据存储的分析非常重要，单独进ERD建模
- ◆ 数据存储需求包括数据实体、数据实体的属性以及它们之间的关系
- ◆ 分析员在数据建模的过程中，常常对ERD进行细化的工作，即处理多对多的关系以及细化属性说明
- ◆ 由于关系数据库中不能直接实现多对多的关系，因此必须建立一个单独的表，来消除多对多的关系
- ◆ **关联实体** – 解决上述问题的人为增加的数据实体，它一定包含两端数据实体的关键字

数据分析 (ERD、IDEF1X)

实体关系图 – ERD

◆ 概念ERD

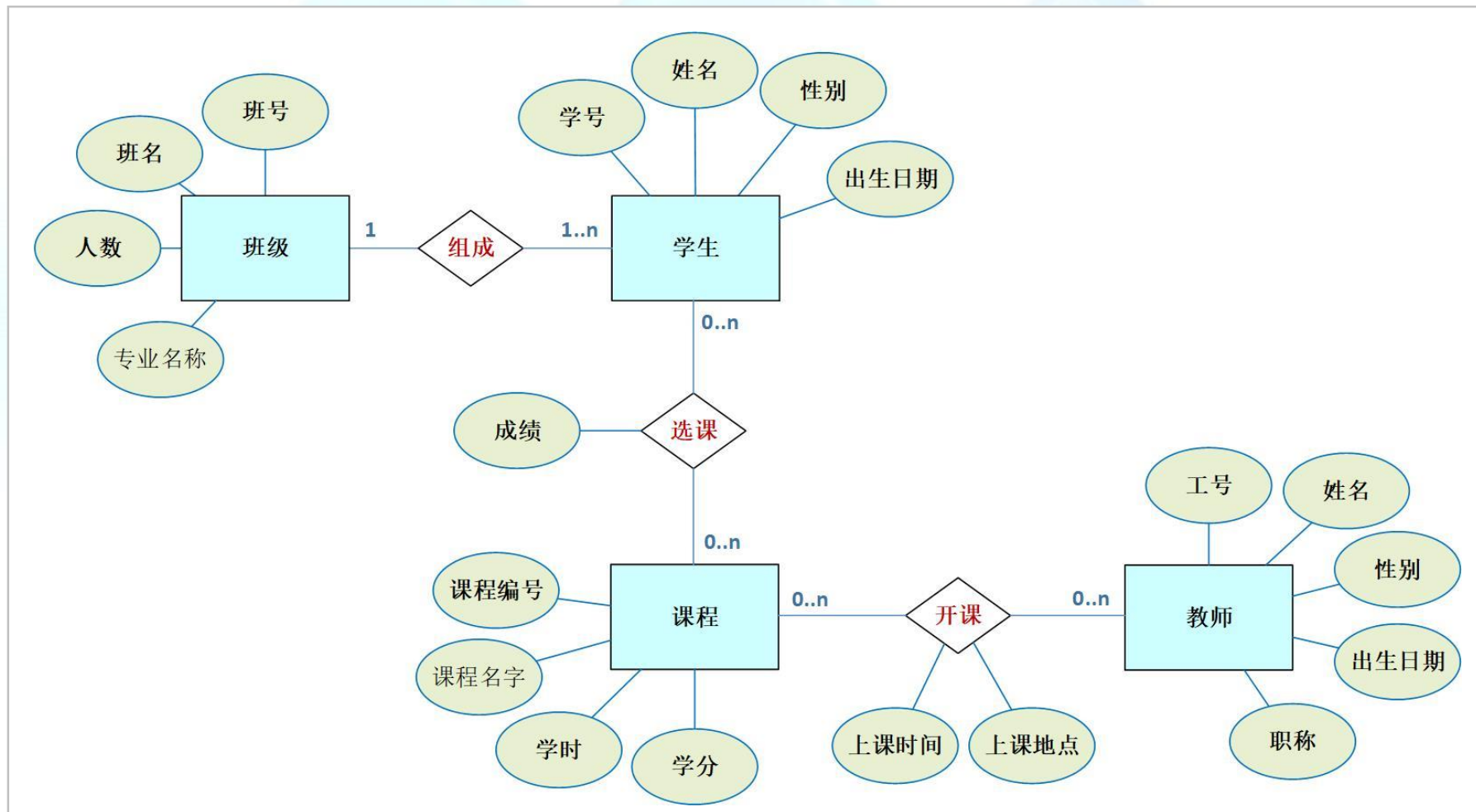
◆ 逻辑ERD

◆ 物理ERD

ERD形式	概念ERD	逻辑ERD	物理ERD
实体 (名称)	√	√	√
关系	√	√	√
关联实体		√	√
属性 (名称)		√	√
属性的类型、 长度、说明		随意	√
主键			√
外键			√

数据分析 (ERD、IDEF1X)

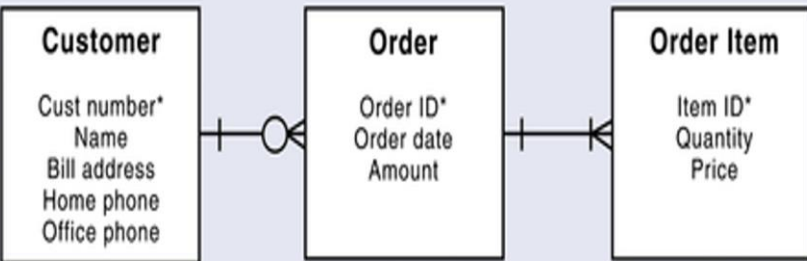
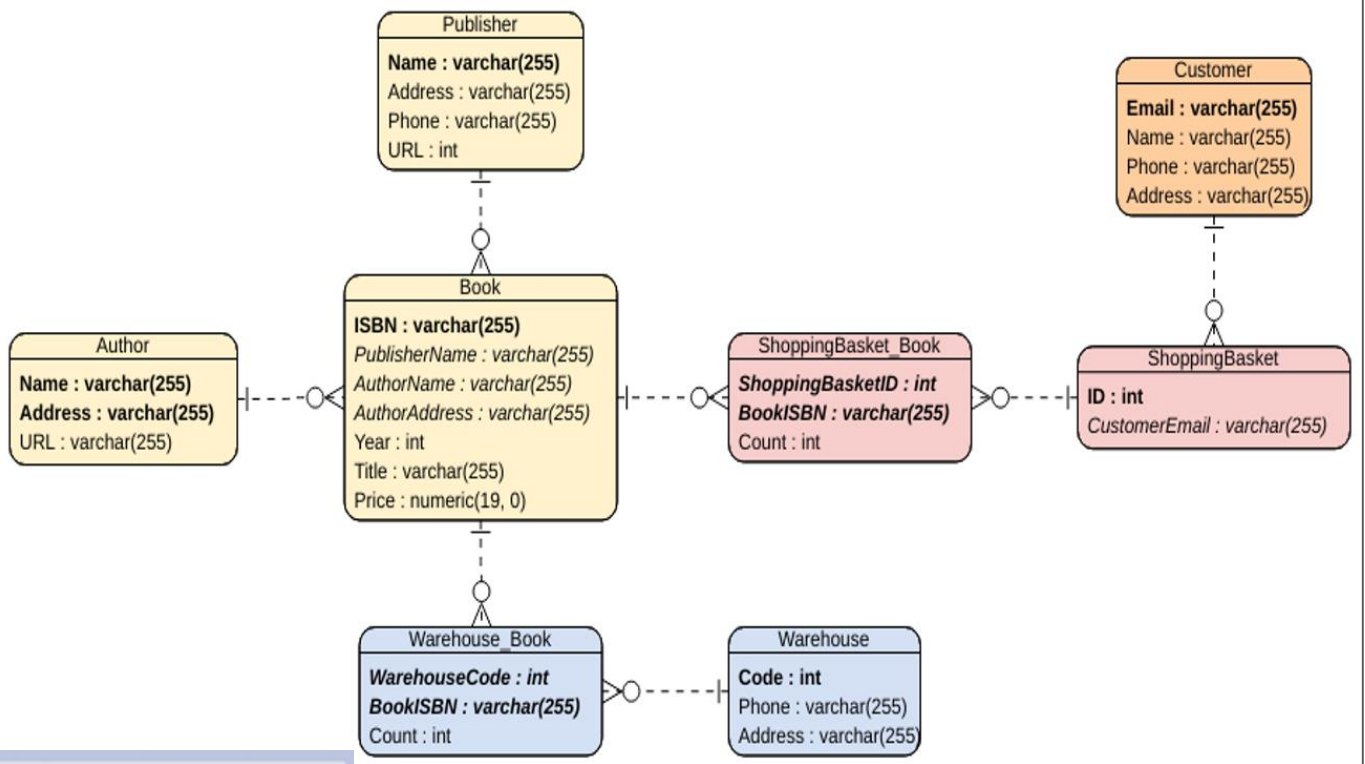
实体关系图 – ERD



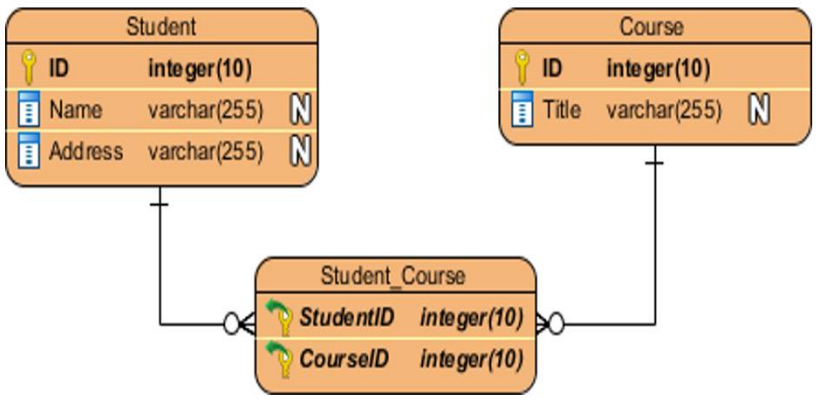
传统的标准画法的ERD (Chen's ERD)

哈工大计算学部/软件学院

数据分析 (E) 实体关系图 - ER

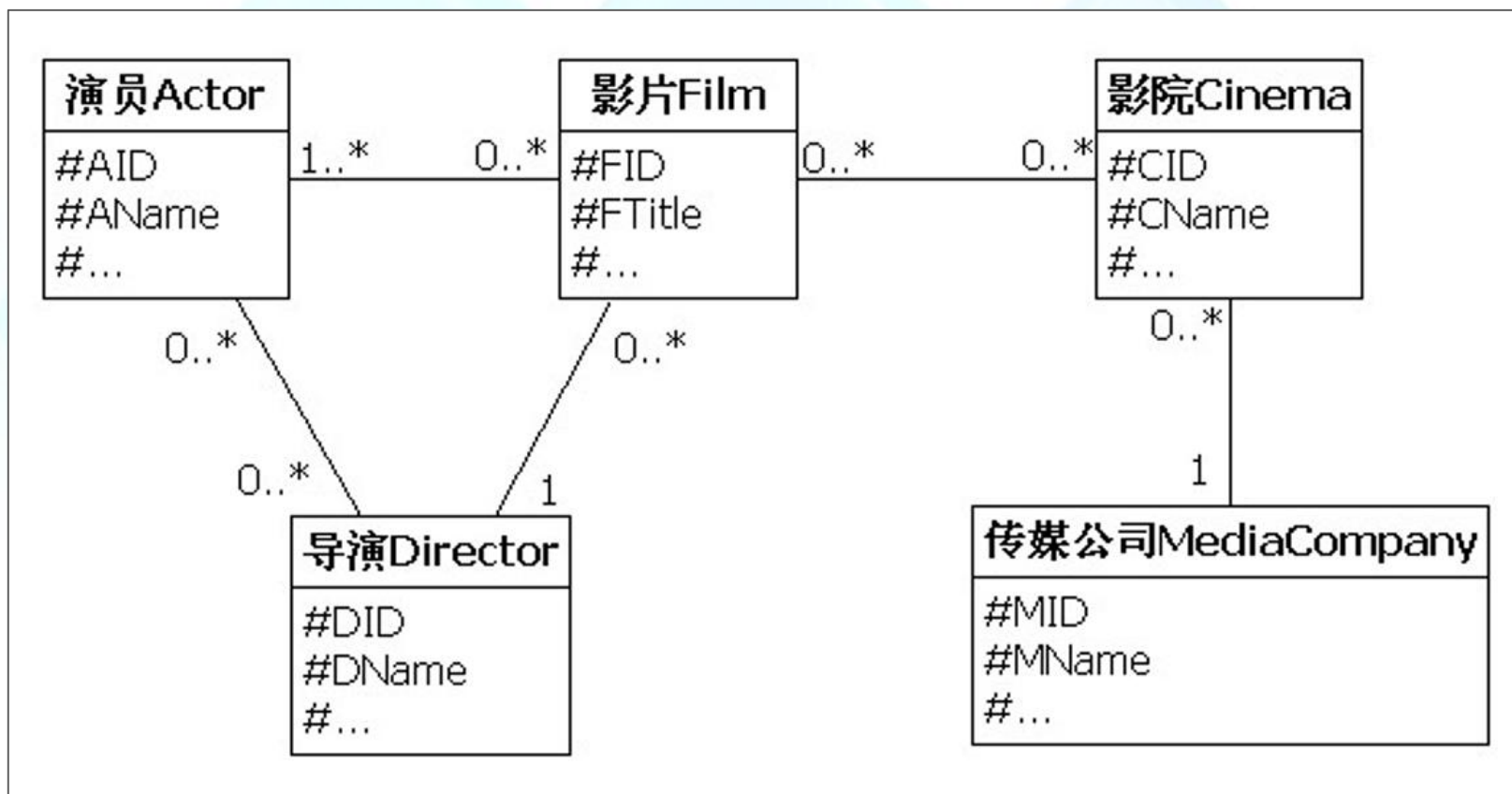


*表示标识符或关键字



数据分析 (ERD、IDEF1X)

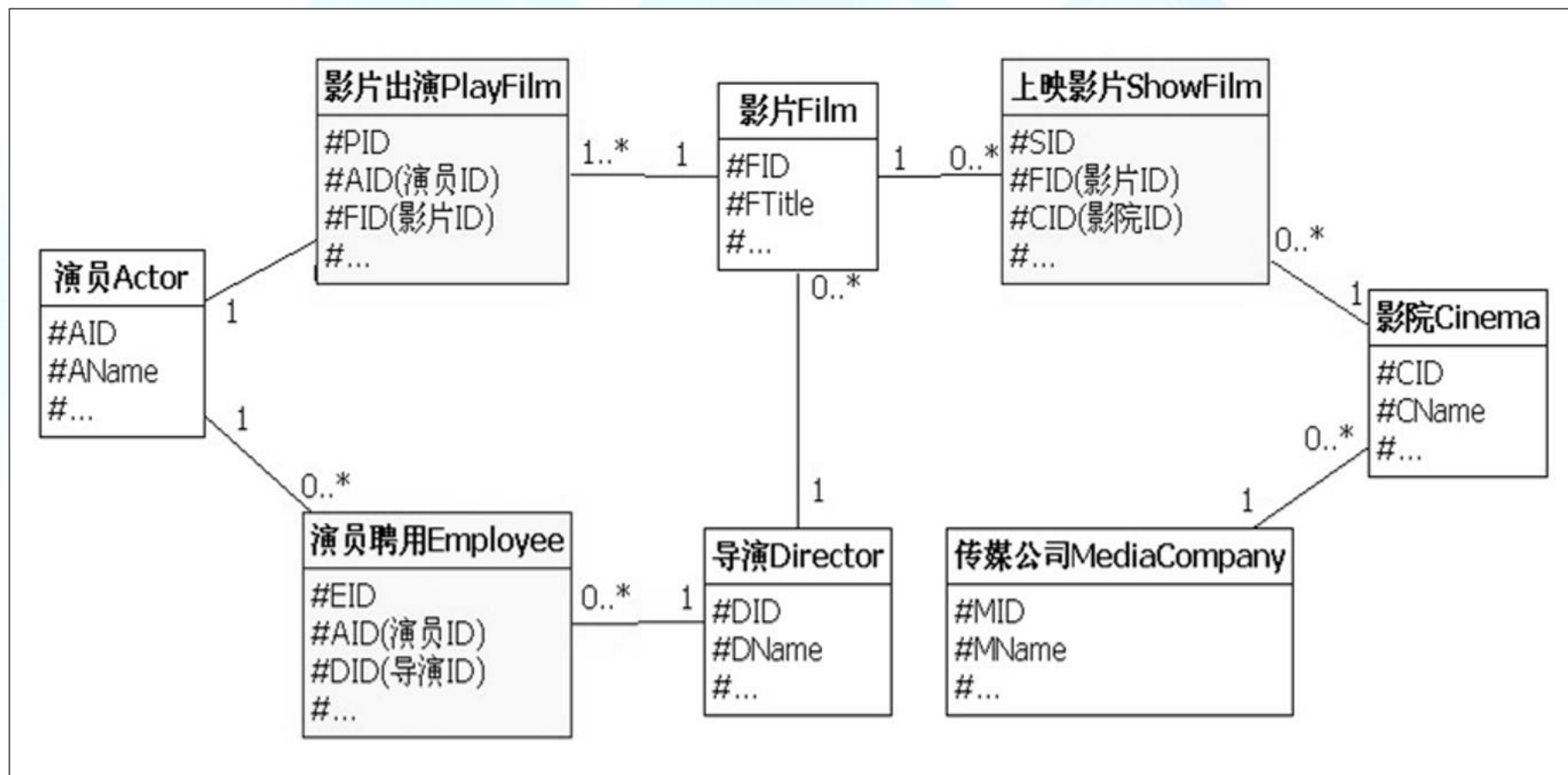
实体关系图 – 类IDEF1X图



含有*..*关系的ERD (类IDEF1X形式)

数据分析 (ERD、IDEF1X)

实体关系图 – 类IDEF1X图

消除*..*关系增加了**关联实体**的ERD（类IDEF1X形式）