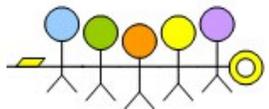
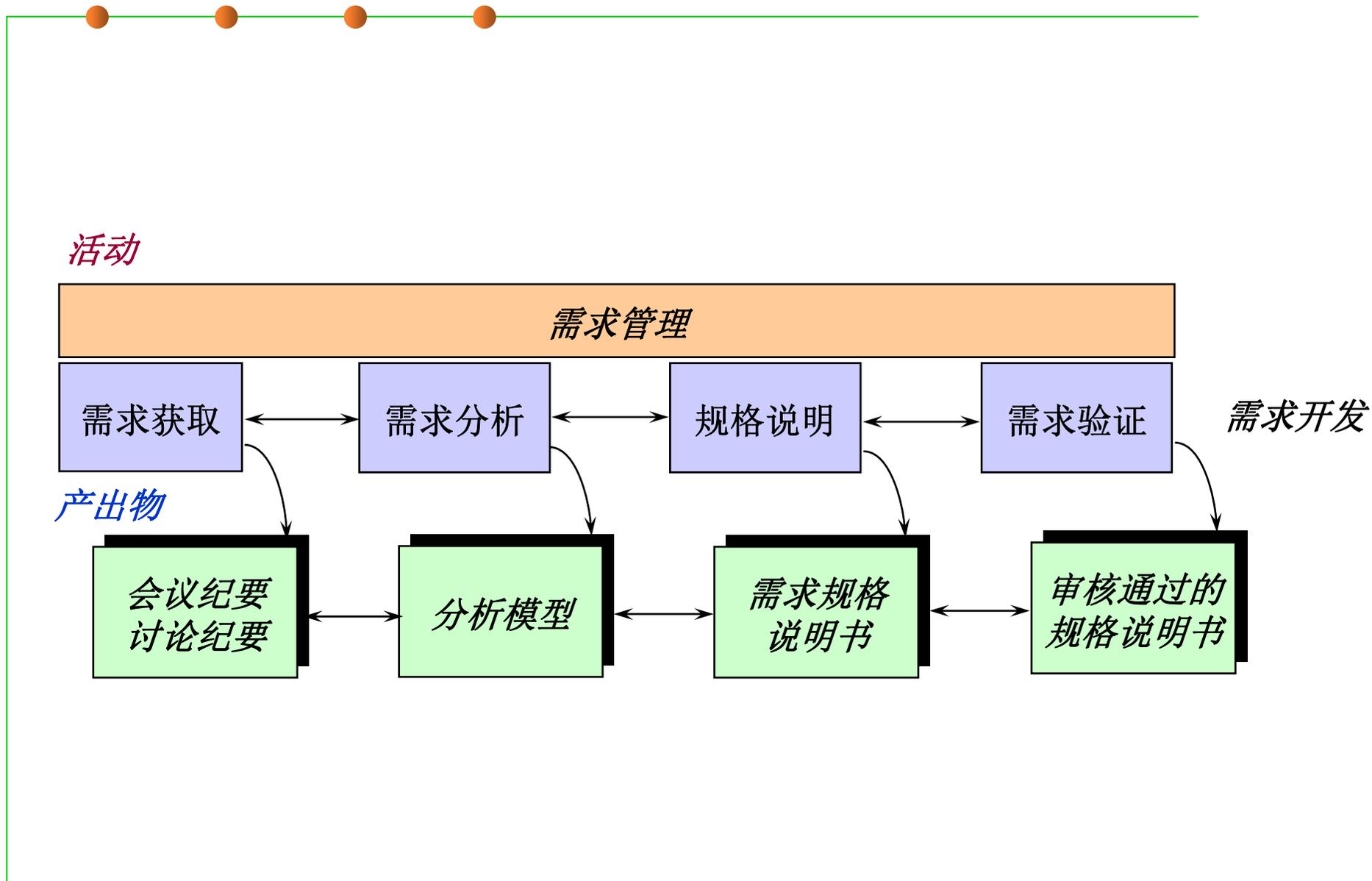


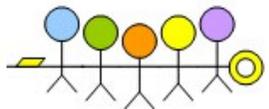
# 主要内容

- 7.1 需求分析方法概述
- 7.2 结构化需求分析
- 7.3 数据流图(DFD)
- 7.4 数据字典(DD)
- 7.5 结构化语言、决策表与决策树
- 7.6\* 实体-联系图(E-R)
- 7.7\* 状态转换图



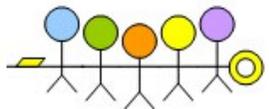
# 需求工程的总体流程





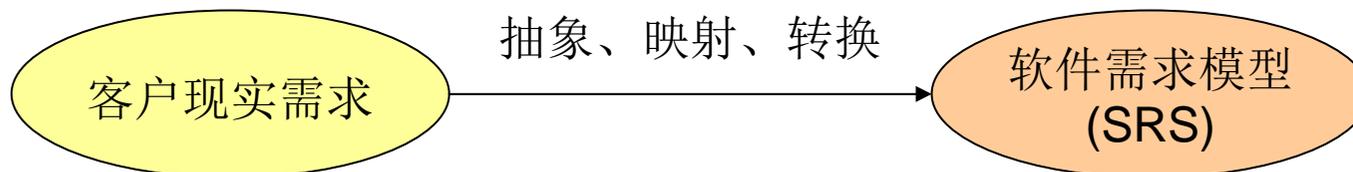
## 7.1 需求分析方法概述

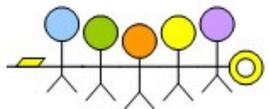




## 需求分析的本质

- **需求分析(Requirement Analysis)**: 对收集到的需求进行提炼、分析和审查, 为最终用户所看到的系统建立概念化的分析模型。
  - 定义系统的边界
  - 分析需求可行性
  - 确定需求优先级
  - 建立需求分析模型
  - 创建数据字典
- **本质:**

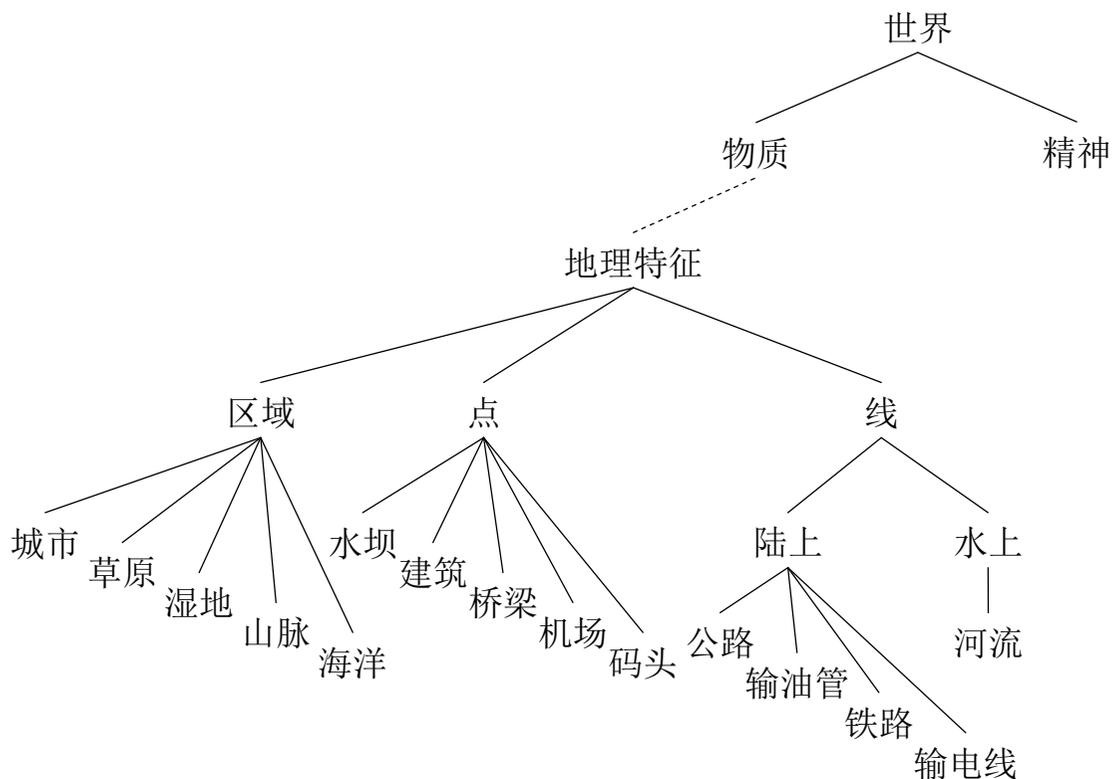


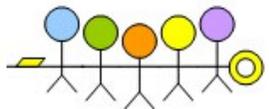


# 需求分析的基本思想：抽象

## ■ 抽象：透过现象看本质

- 抓住事物的本质，捕获问题空间的“一般/特殊”关系是认识、构造问题的一般途径。

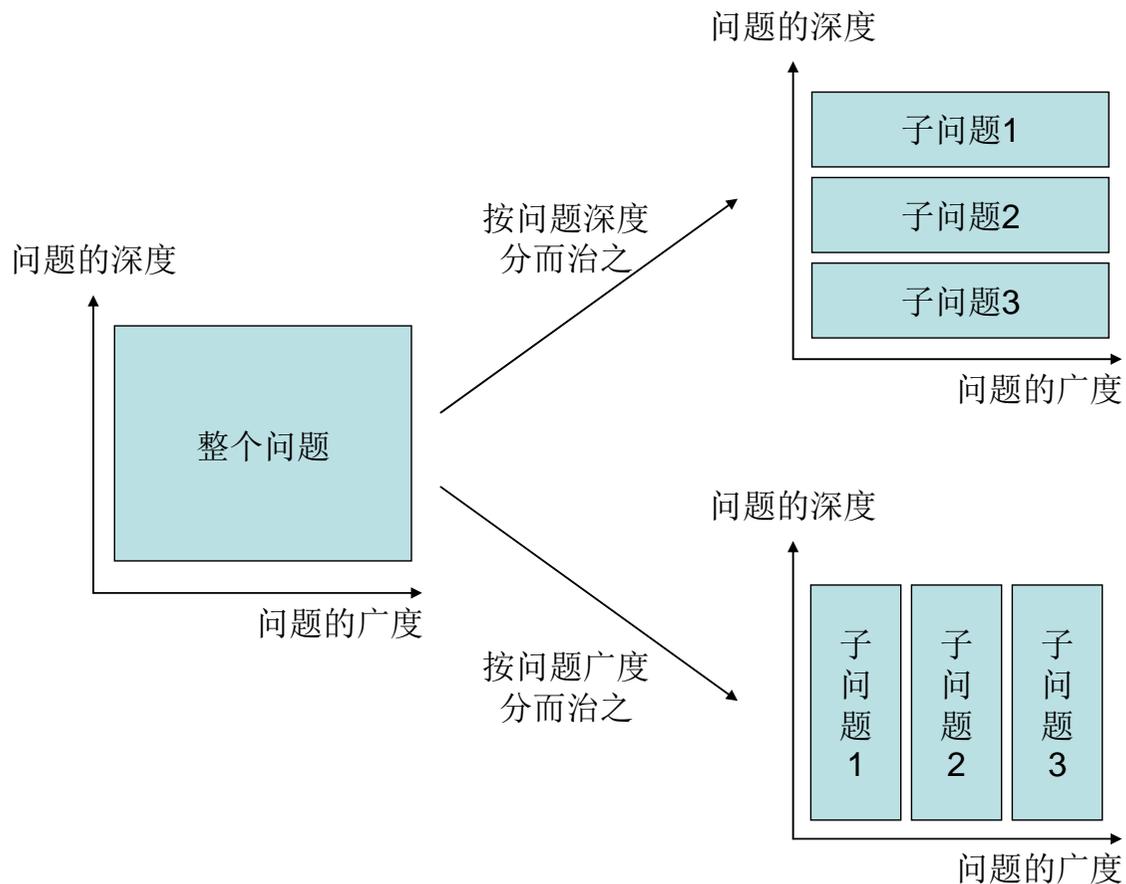


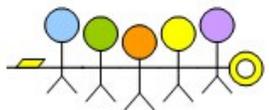


# 需求分析的基本思想：划分

## ■ 划分：分而治之

— 分离问题，捕获问题空间的“整体/部分”关系是降低问题复杂性的基本途径。

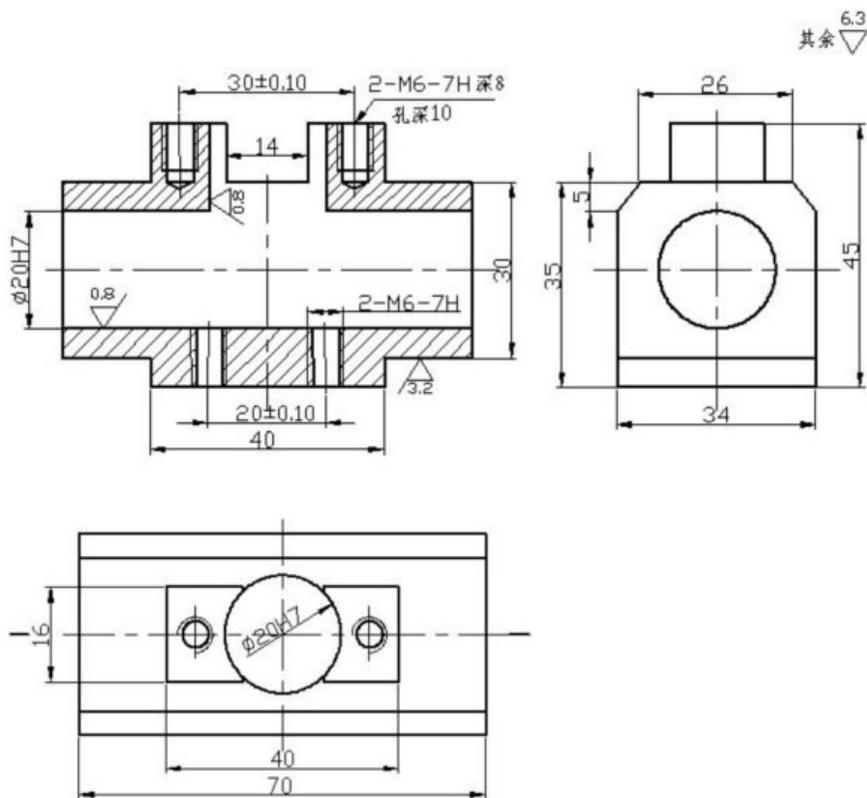


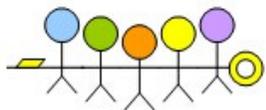


## 需求分析的基本思想：投影

### ■ 投影：从不同视角看问题

— 捕获并建立问题空间的多维视图是描述问题的基本手段。





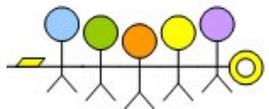
## 需求分析的基本思想：建模

### ■ 建模：规格严格、功夫到家

- 采用规范的描述方法，将模糊的、不确定的用户需求表达为清晰的、严格的模型，作为进一步设计与实现的基础。
- 模型的作用：
  - 增强对需求的理解
  - 检测不一致性、模糊性、错误和遗漏
  - 在项目的参与者之间更高效的交流

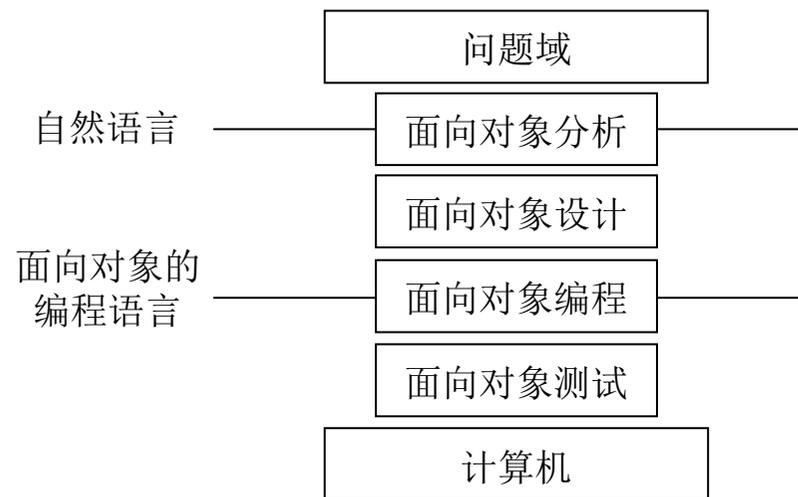
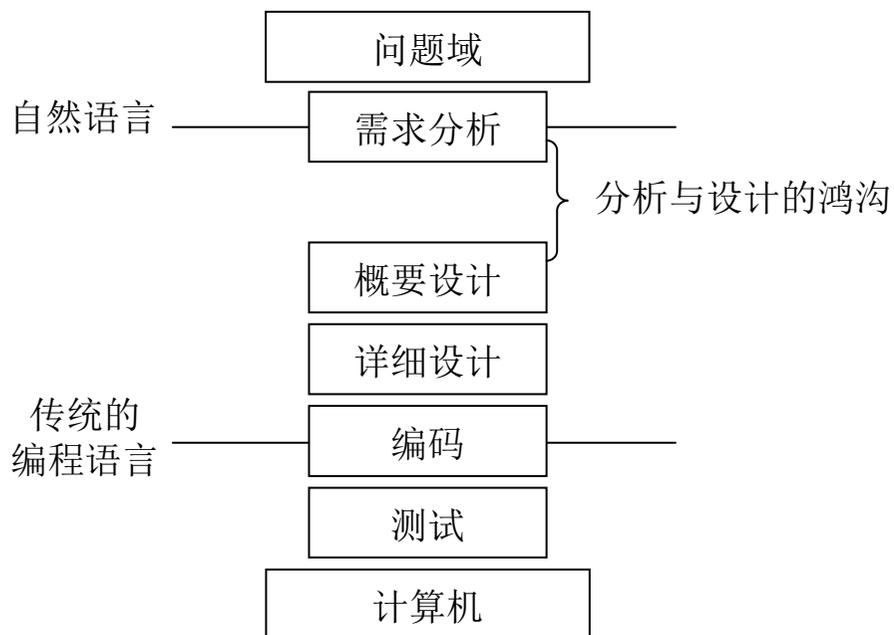
### ■ 两种模型形态：

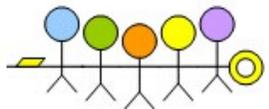
- 形式化的数学模型(formal mathematical model)
- 非形式化的图形化模型 (informal graphical model)



# 需求分析方法

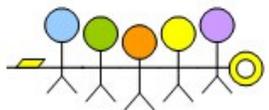
- 两种主要需求分析方法：
  - 结构化分析与设计方法(Structured Analysis and Design Technique, SADT)
  - 面向对象分析与设计方法(Object-Oriented Analysis and Design, OOAD)





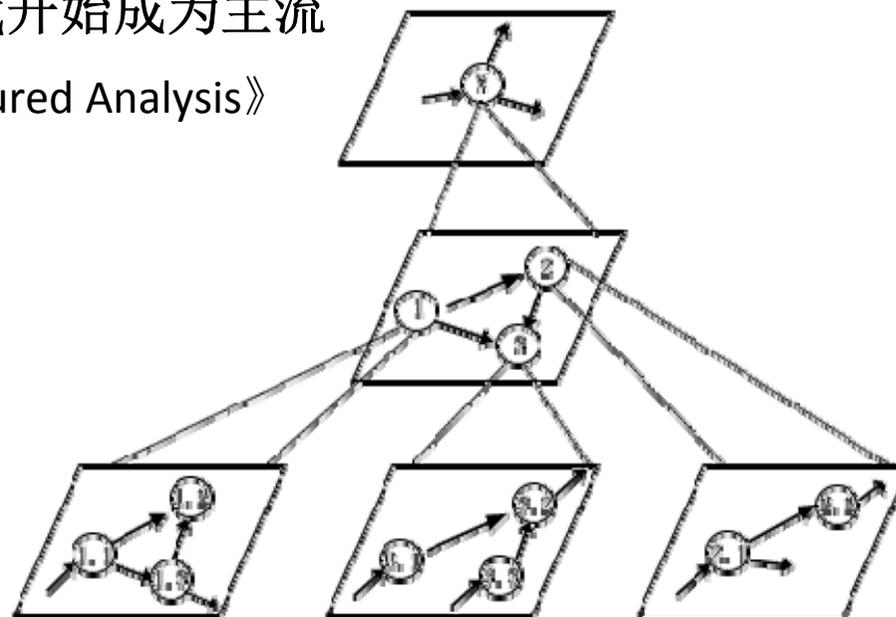
## 7.2 结构化需求分析方法

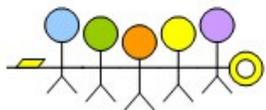




## 结构化需求分析方法的起源

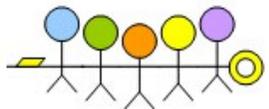
- **结构化分析方法(SA):** 将待解决的问题看作一个系统, 从而用系统科学的思想方法(抽象、分解、模块化)来分析和解决问题。
- 起源于结构化程序设计语言(事先设计好每一个具体的功能模块, 然后将这些设计好的模块组装成一个软件系统);
- 最早产生于**1970年代中期**, **1980年代**开始成为主流
  - Yourdon于1989年出版《Modern Structured Analysis》
- 核心思想:
  - 自顶向下的分解(top-down)



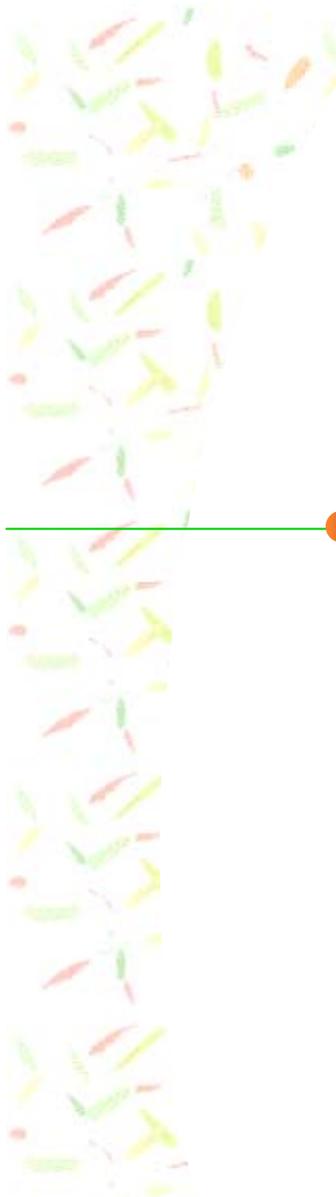


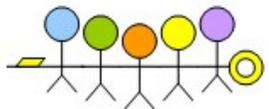
## 结构化方法的模型

- 结构化需求分析方法通常需建立以下模型：
  - 数据流图(Data Flow Diagram, DFD)
    - 描述系统由哪些部分组成、各部分之间有什么联系等
  - 数据字典(Data Dictionary, DD)
    - 定义了数据流图中每一个数据元素
  - 结构化语言(Structured Language)
  - 判定表或判定树(Decision Table/Tree)
    - 详细描述数据流图中不能被再分解的每一个加工的内部处理逻辑
  - 实体联系图(Entity-Relationship Diagram, E-R)
  - 状态转换图(State Transition Diagram, STD)



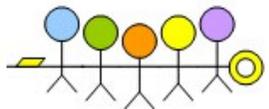
## 7.3 数据流图(DFD)





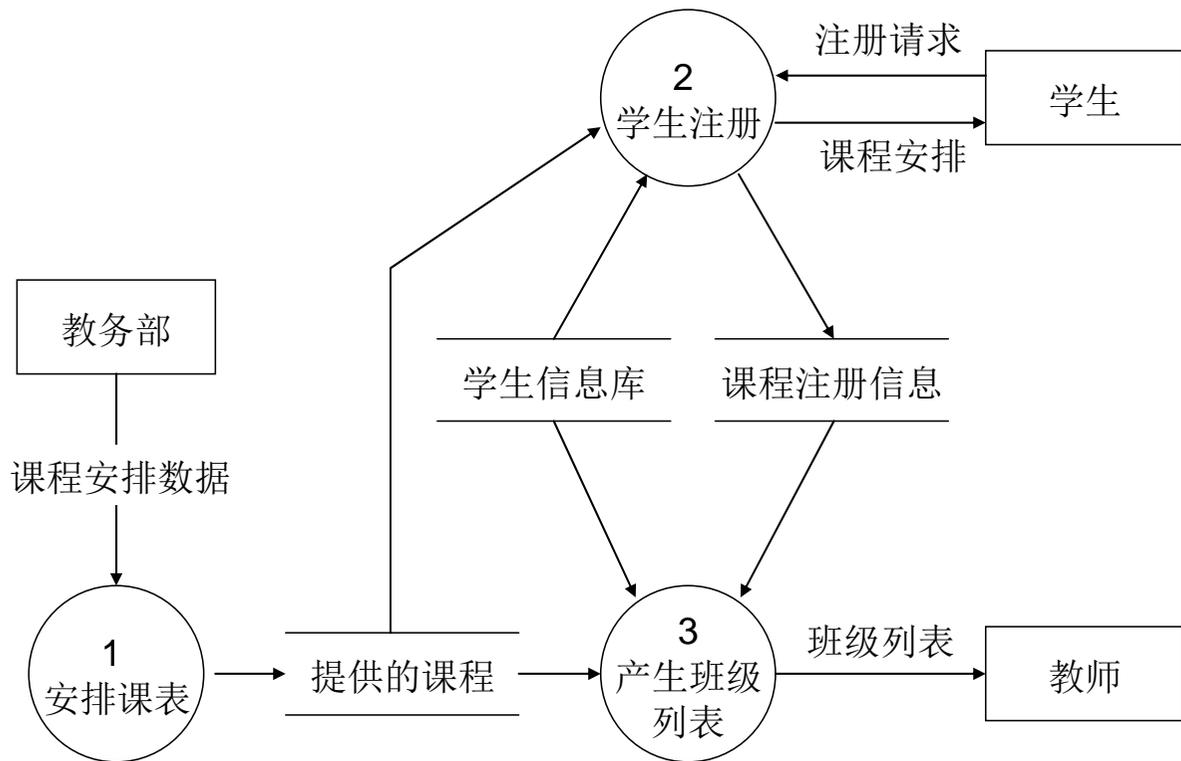
## 数据流图(DFD)

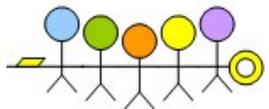
- 7.3.1 DFD的作用
- 7.3.2 DFD的模型元素及图形化表示
- 7.3.3 DFD的层次性
- 7.3.4 绘制DFD应遵循的约束
- 7.3.5 DFD应用案例
- 7.3.6 DFD树



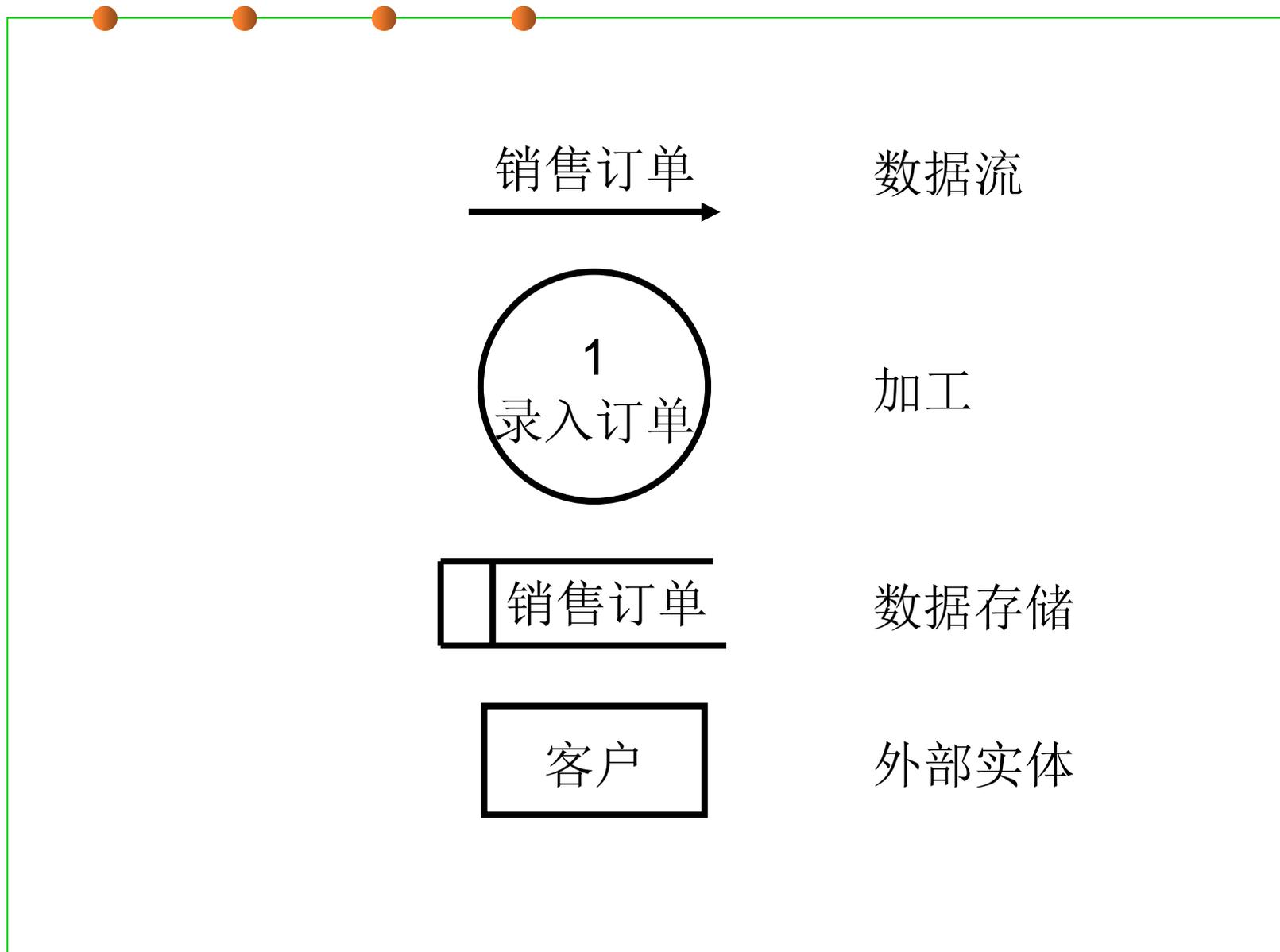
## 7.3.1 数据流图(DFD)

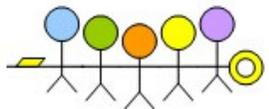
- **数据流图(Data Flow Diagram, DFD):** 结构化系统分析的基本工具
  - 描绘数据在系统中各逻辑功能模块之间的流动和处理过程，是一种功能模型
  - 主要刻画“功能的输入和输出数据”、“数据的源头和目的地”





## 7.3.2 DFD的主要元素

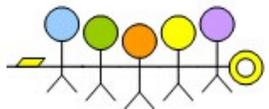




## DFD的主要元素(1): 加工

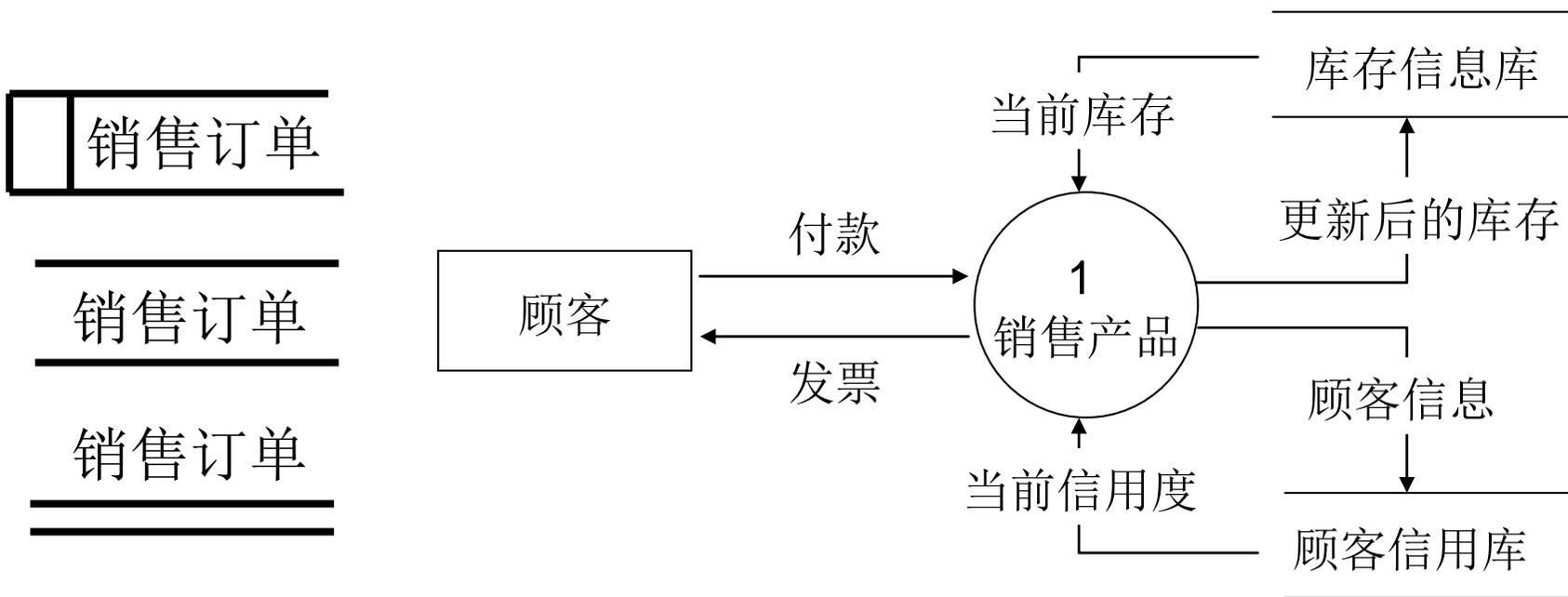
- **加工(又称数据处理, data processing): 对数据流进行某些操作或变换。**
  - 收集、排序、选择、聚集、分析等
  - 加工要有名字, 通常是动词短语, 简明地描述完成什么事情
  - 在分层的数据流图中, 加工还应编号
  - 三种类型: 计算机自动加工、手工加工、人机协作的加工

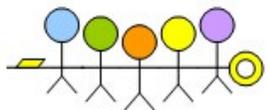




## DFD的主要元素(2): 数据存储

- **数据存储(data storage, 也称文件):** 需要在外存储器上保存的数据, 它可以是数据库文件或任何形式的数据组织。
  - 以名词命名





## DFD的主要元素(3): 外部实体

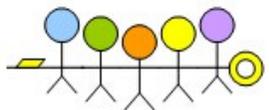
- **外部实体(external entity):** 本系统外部环境中的实体(包括人员、组织或其他软件系统)
  - 也称为“数据源点/数据终点”，表示产生数据的源头或消费数据的终点
  - 以名词短语命名
  - 不能直接访问数据存储

客户

学生

库存系统

旅行社



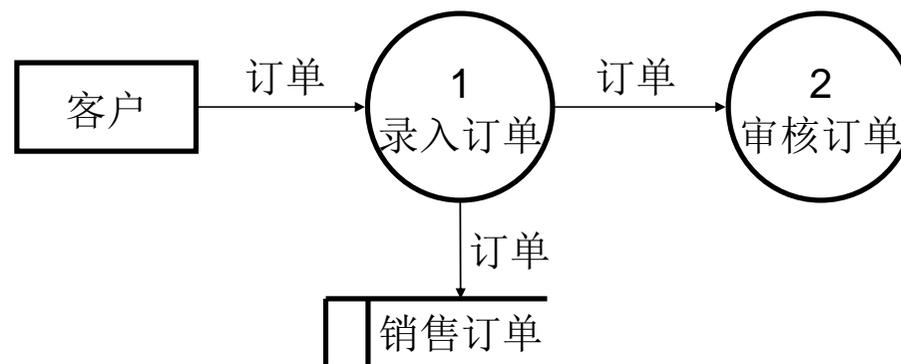
## DFD的主要元素(4): 数据流

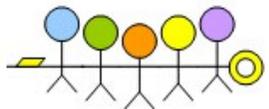
- **数据流(data flow): 数据在系统内传播的路径**
  - 由一组成分固定的数据组成。
  - 由于数据流是流动中的数据，所以必须有流向
  - 应用名词或名词短语命名
  - 可能是纸张上的数据、电子数据、通过网络传输的数据等

销售订单



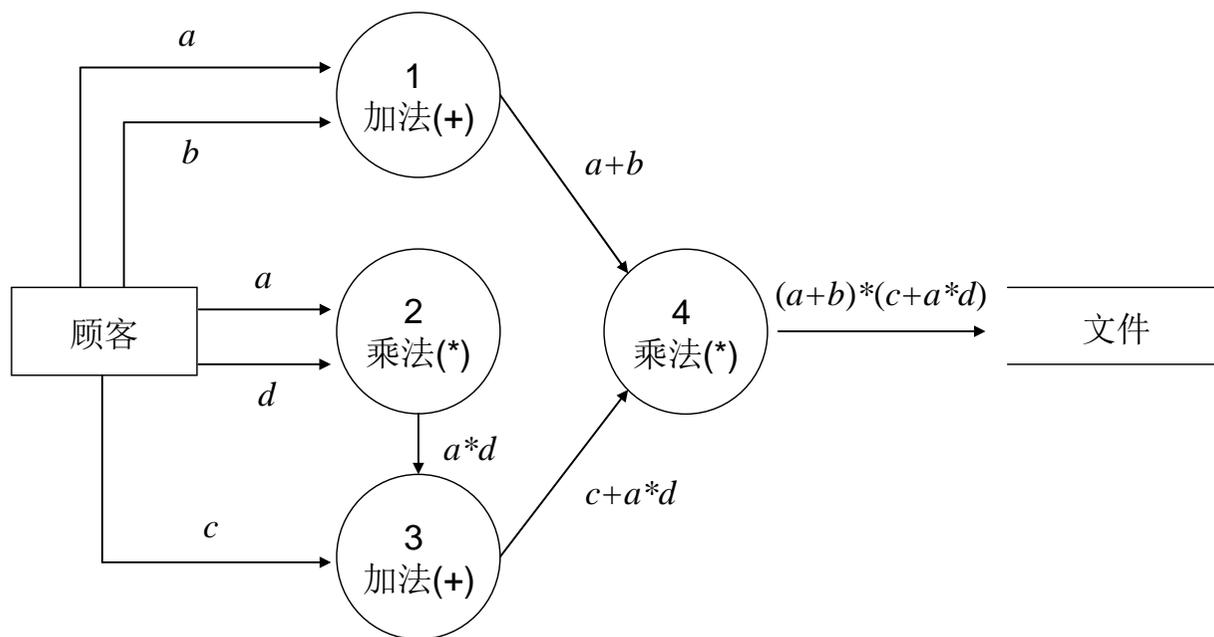
- 可能存在于:
  - 外部实体与加工之间;
  - 加工与加工之间;
  - 加工与数据存储之间

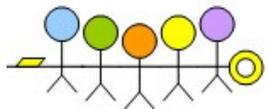




## DFD的简单练习

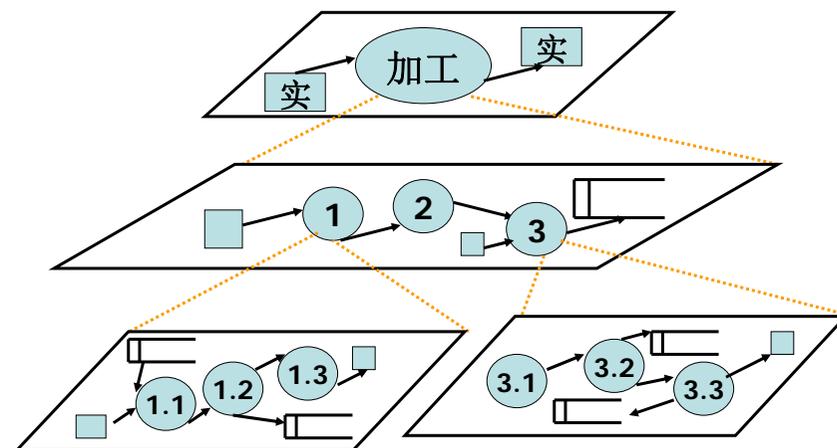
- 背景：用户输入 $a$ 、 $b$ 、 $c$ 、 $d$ 四个值，系统计算 $(a+b)*(c+a*d)$ ，并将结果输出到一个文件中存储。
- 问题：绘制该系统的DFD

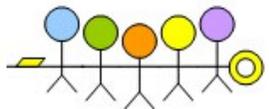




## 7.3.3 DFD的层次性

- DFD的层次性：自顶向下的分解(top-down)
- DFD的两种类型：
  - 环境关联DFD图(Context-level DFD, 或Context Diagram): 也称顶层DFD图, 描述了系统与外部环境之间的数据输入/输出关系;
  - 系统内部DFD图(Inner-level DFD): 描述系统内部各功能模块之间的数据流动关系
    - 0-层DFD图
    - 1-层DFD图
    - ...
    - N层DFD图

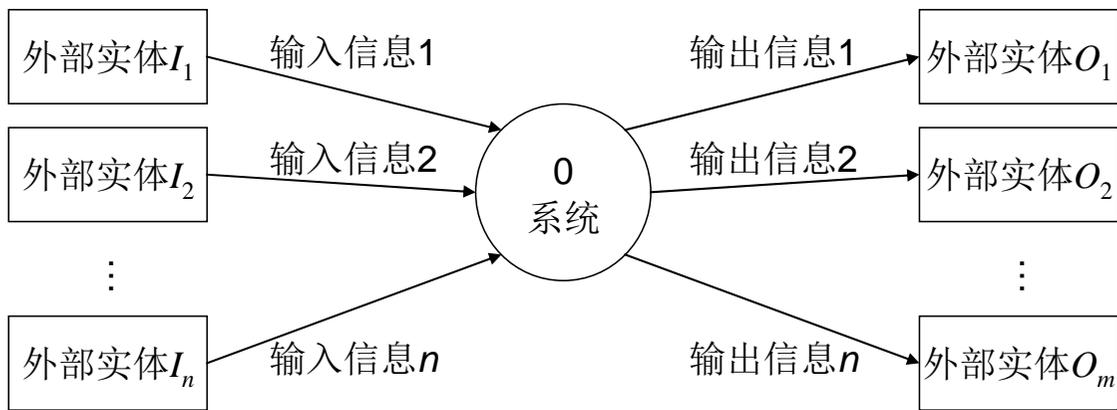


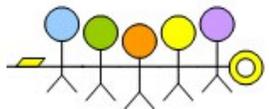


# 顶层DFD

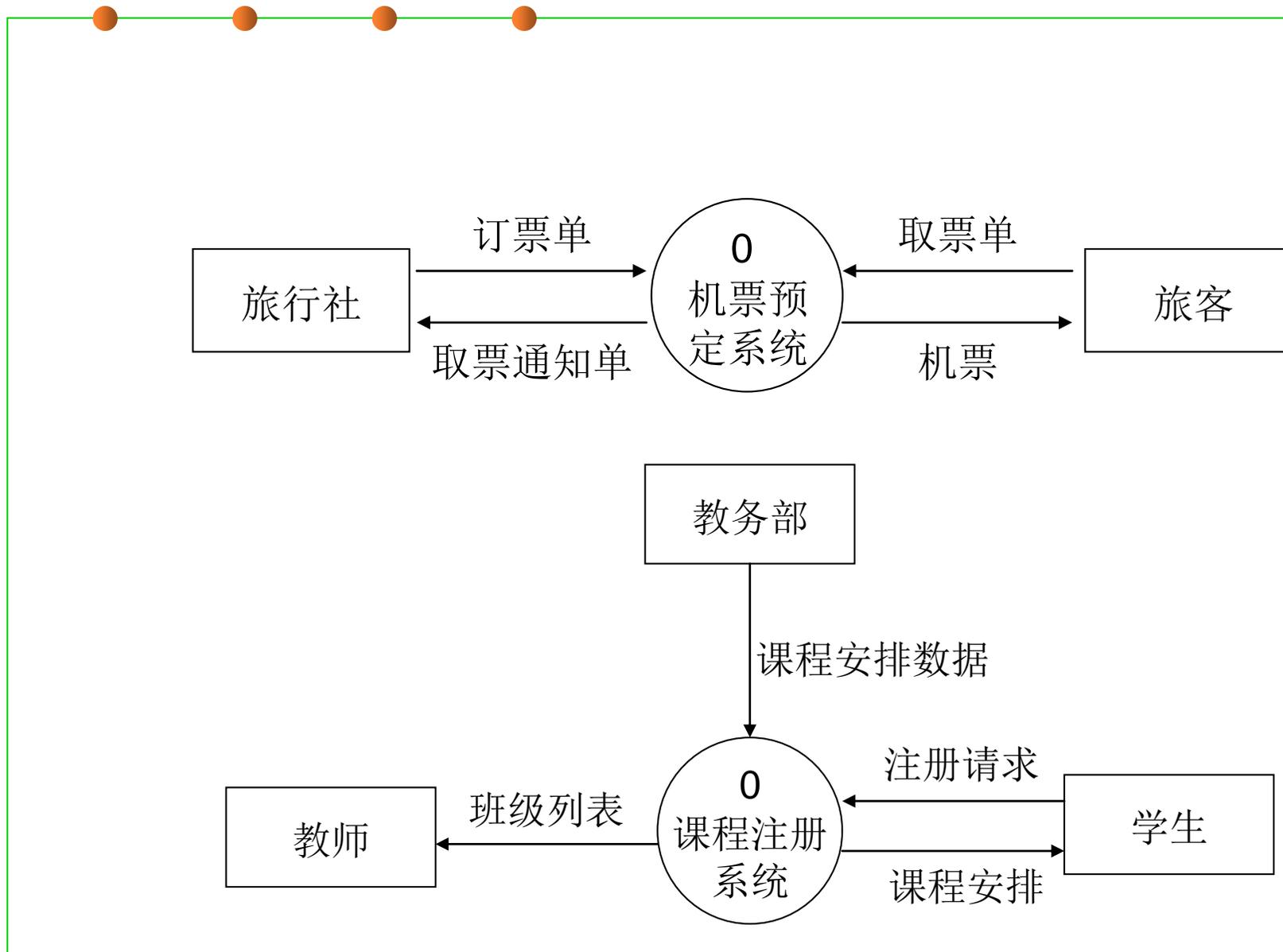
## ■ 顶层DFD图(关联图)

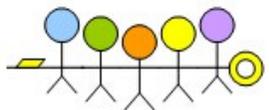
- 通过系统和外部世界之间的联系来描述系统的范围
- 确定了通过某一接口与系统相连的外部实体，同时也确定了外部实体和系统之间的数据流
- 只包含一个加工，用以表示被开发的系统，然后考虑该系统有哪些输入数据、输出数据流
- 编号：0





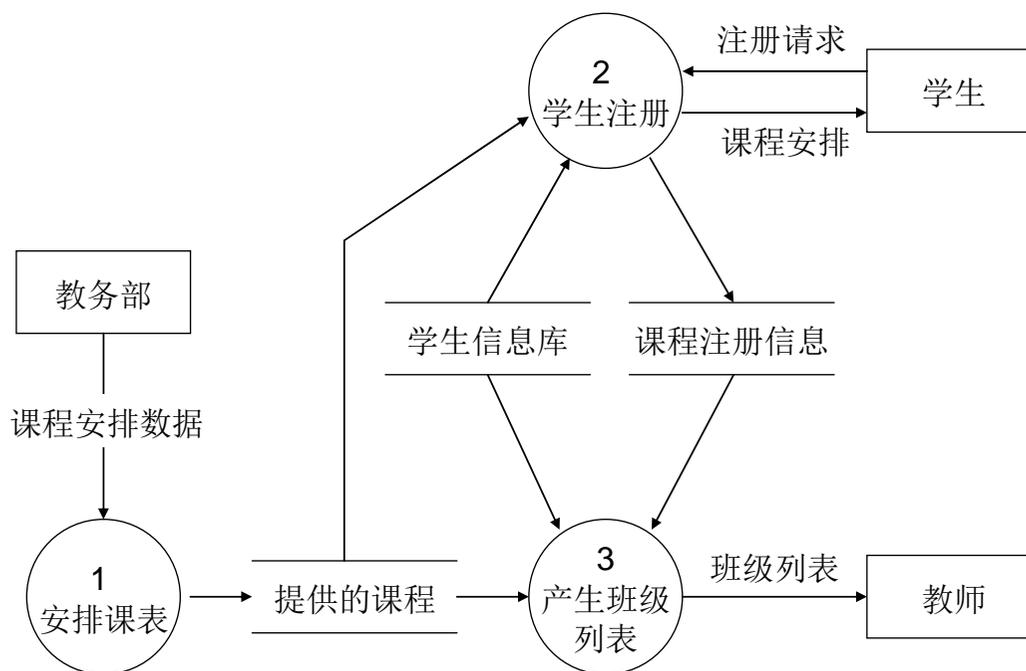
## 示例：顶层DFD

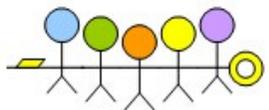




## 0层DFD

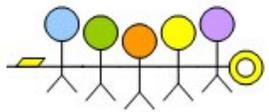
- 将顶层DFD图中的系统分解为若干个子系统，决定每个子系统间的数据接口和活动关系，得到**0层DFD图**；
- 编号：1、2、...、n





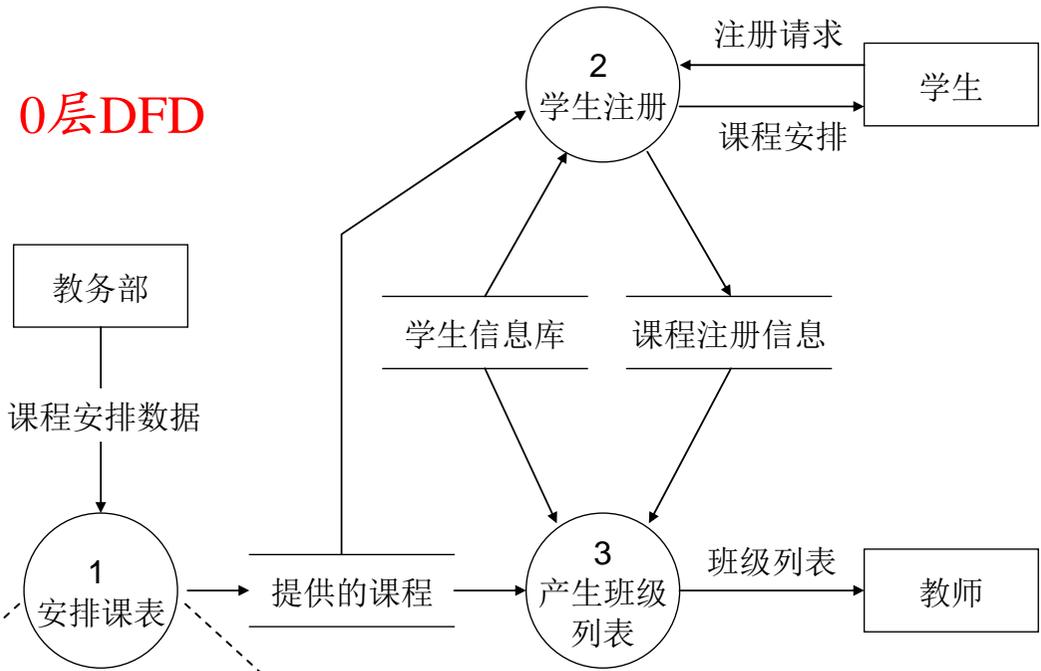
## 底层DFD

- 针对0层DFD中的每一个子系统，对其继续分解得到更细化的加工，进而逐渐向下构造得到1层DFD、2层DFD、...、n层DFD，一直到不能或不需再分解为止。
- 最底层DFD中的加工称为“基本加工”。
- 编号：
  - 1层DFD: 1.1、1.2、...、1.n
  - 2层DFD: 1.1.1、1.1.2、...、1.1.n
  - ...

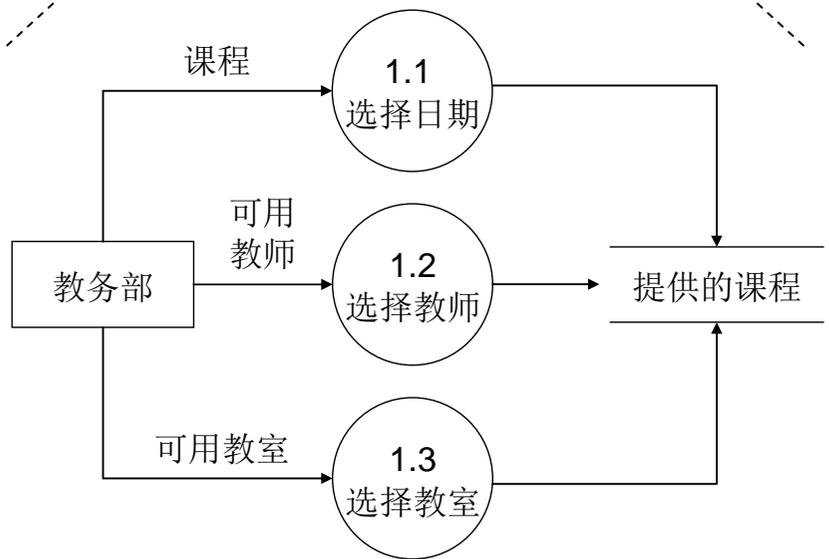


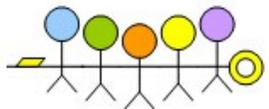
# 底层DFD

## 0层DFD

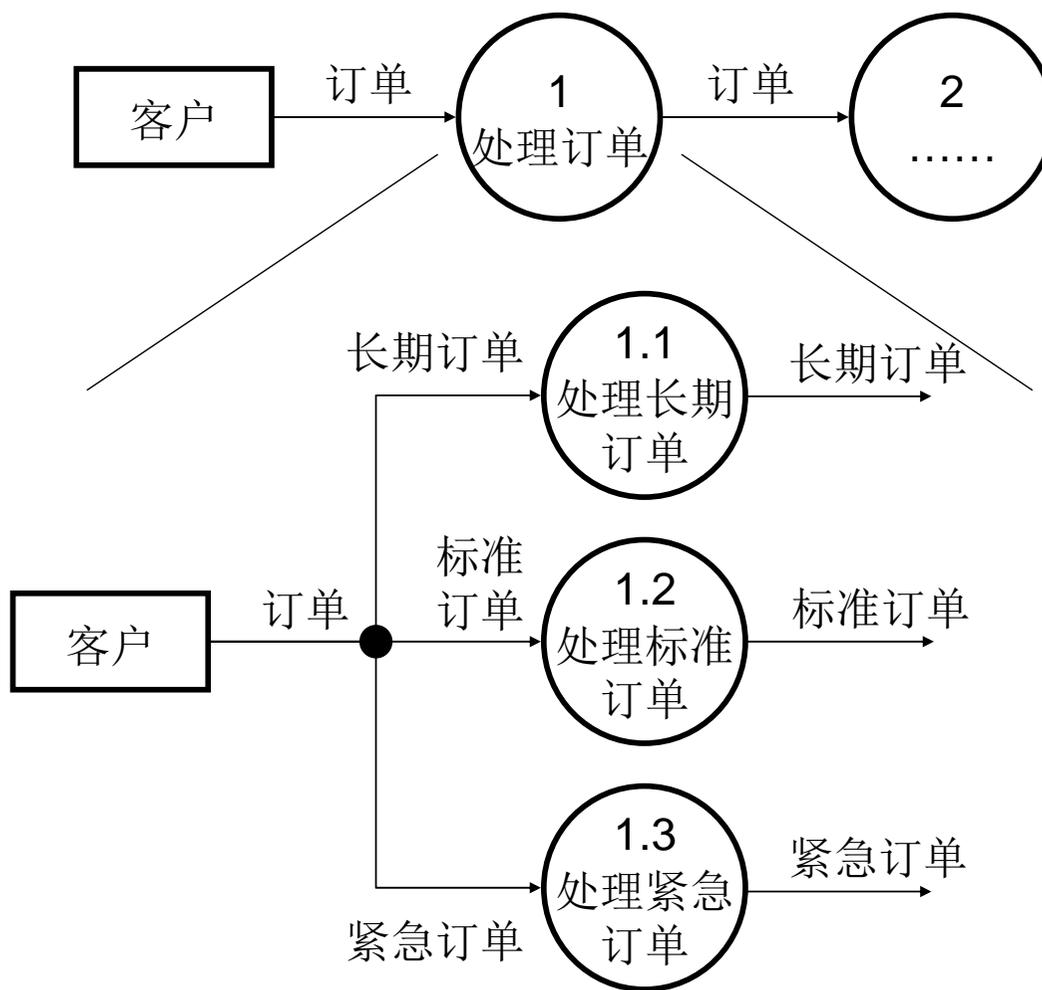


## 1层DFD





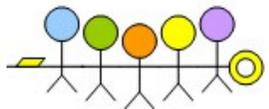
# 数据流的分解





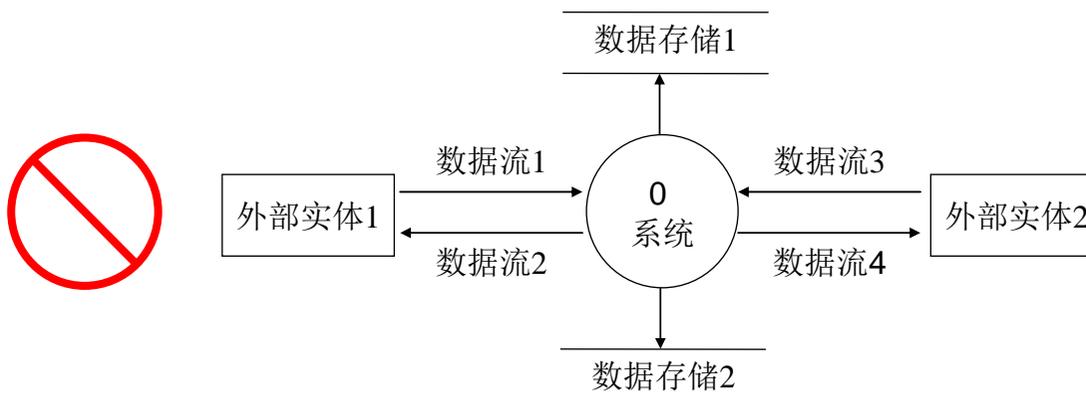
## 如何识别数据流

- 通过识别“事件”来识别数据流，进而识别得到加工、数据存储
- 事件的分类：
  - **外部事件(External events)**: 外部实体与系统进行交互(顾客下订单、供应商的货物到达)
  - **决策事件(Decision events)**: 需要外部实体为系统某些业务做出决策(是否接受订单)
  - **时间性事件(Temporal events)**: 由时间所触发的周期性时间(每月25号编制下月计划、每天17点盘点库存)
  - **状态事件(State events)**: 由某些数据的变化所自动触发的事件(当库存量下降到100以下时，启动采购流程)



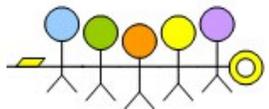
## 7.3.4 绘制DFD的一些基本原则

- 把数据存储放在0层数据流图或更低层子图上，不要放在顶层的关联图上



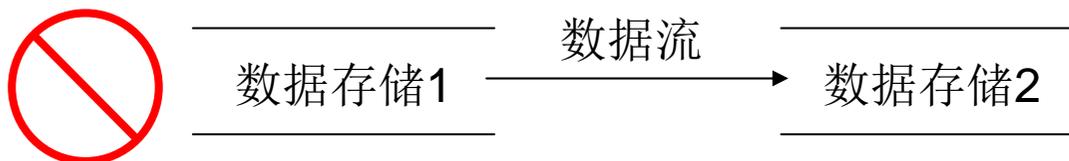
- 使用数据流图时，不要试图让数据流图反映处理的顺序，忽略系统的运行时的时间特性
- 加工通过数据存储进行通讯，而尽量避免从一个过程直接流到另一过程



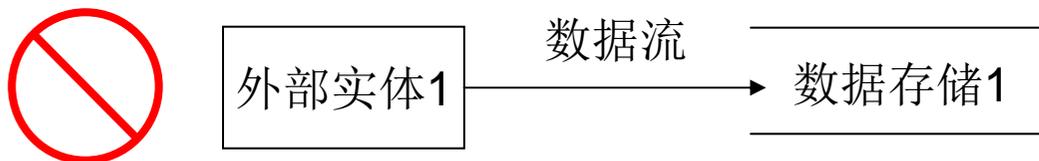


## 绘制DFD的一些基本原则

- 数据不能直接由一个数据存储直接流到另一个数据存储



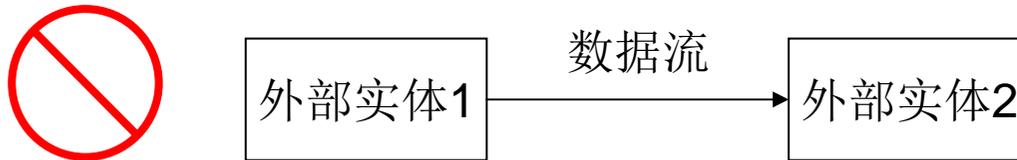
- 数据不能直接从一个外部实体直接流到一个数据存储

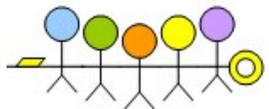


- 数据不能直接从一个数据存储直接流到一个外部实体



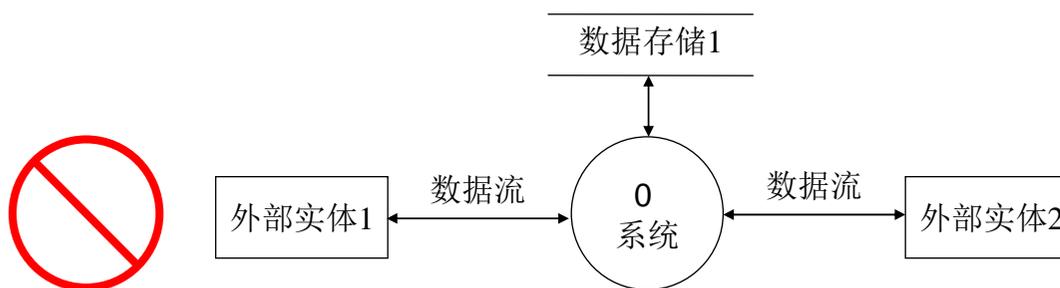
- 数据不能直接在外部实体之间流动



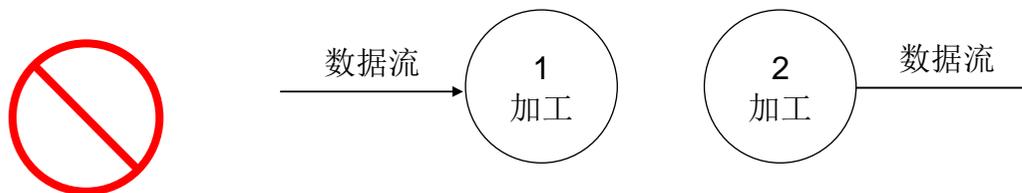


## 绘制DFD的一些基本原则

- 数据流是单向的

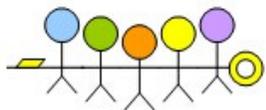


- 任何加工必须有输入和输出数据流



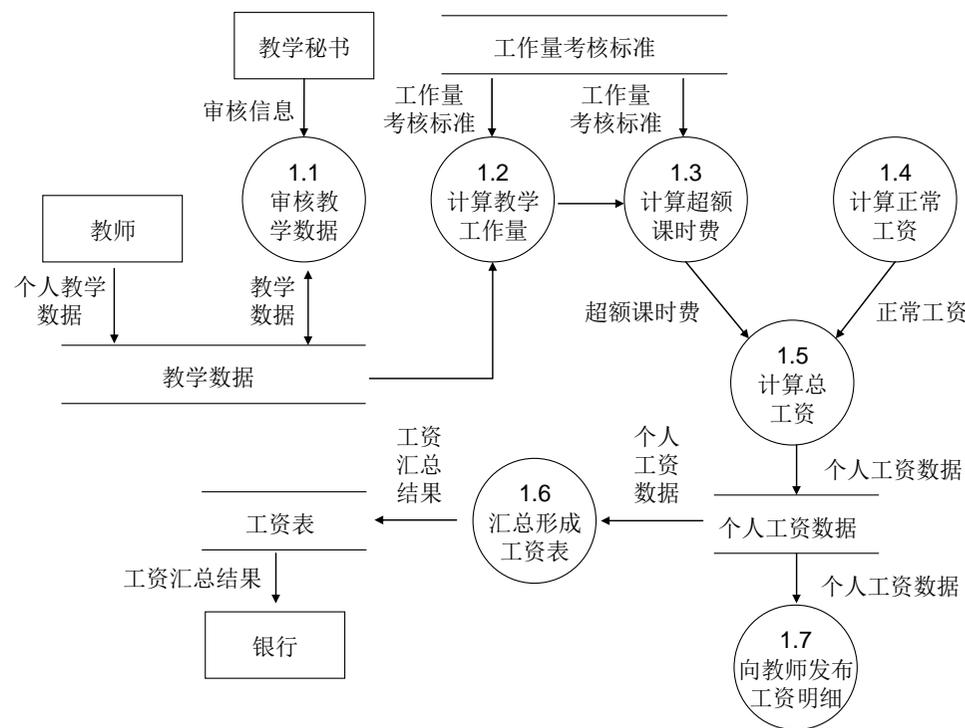
- 对现有加工进行持续的分解和组合，直到所有加工之间达到较高的聚合度；
- 尽量将每一张DFD上的所有元素数目控制在7-12个。

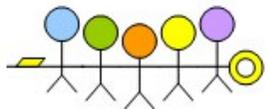




## 错误的DFD

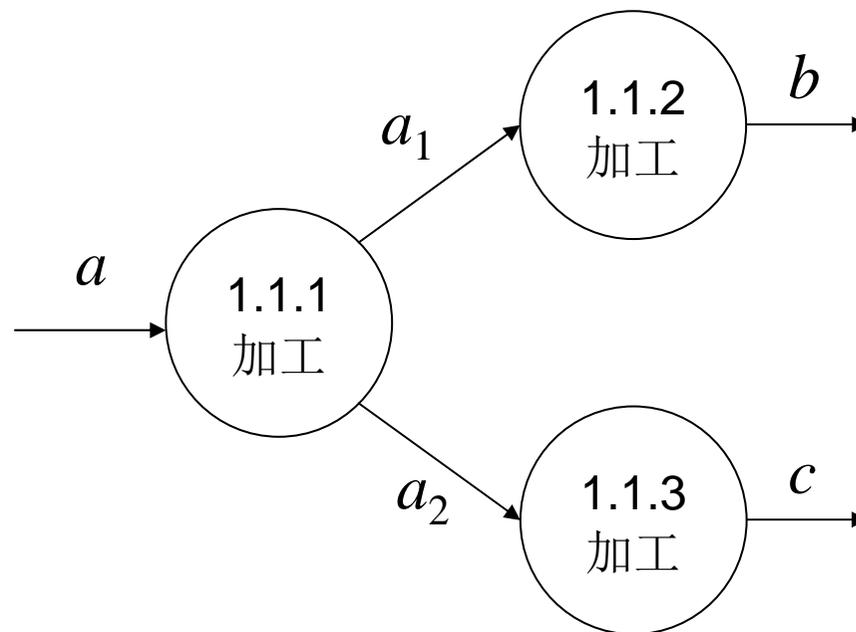
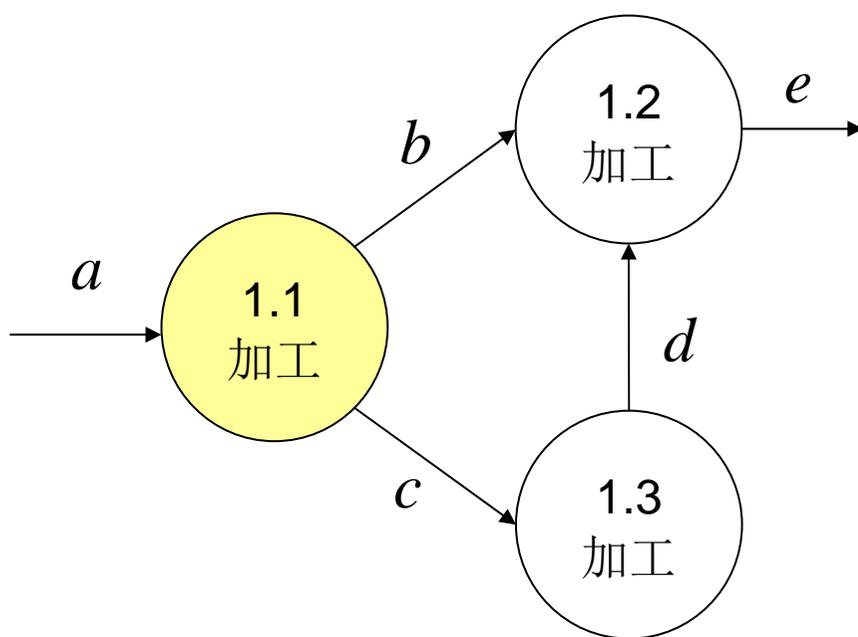
- 教师将个人教学数据录入系统，教学秘书对数据进行审核，发现错误并进行修改。
- 系统根据工作量考核标准和教师的教学数据，计算教师的工作量。
- 教师的工资分为两部分：正常工资和超额课时费。
- 对于前者，系统读取事先已存在的工资标准加以计算；对于后者，系统根据教师的工作量和工作量考核标准来计算得到。
- 得到这两部分工资数据之后，系统计算教师的总工资，并将所有教师的工资数据汇总形成工资表，发送给银行，由银行进行工资发放。
- 与此同时，系统需要向每个教师发布其个人的工资计算结果。

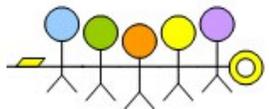




## 父图与子图的平衡

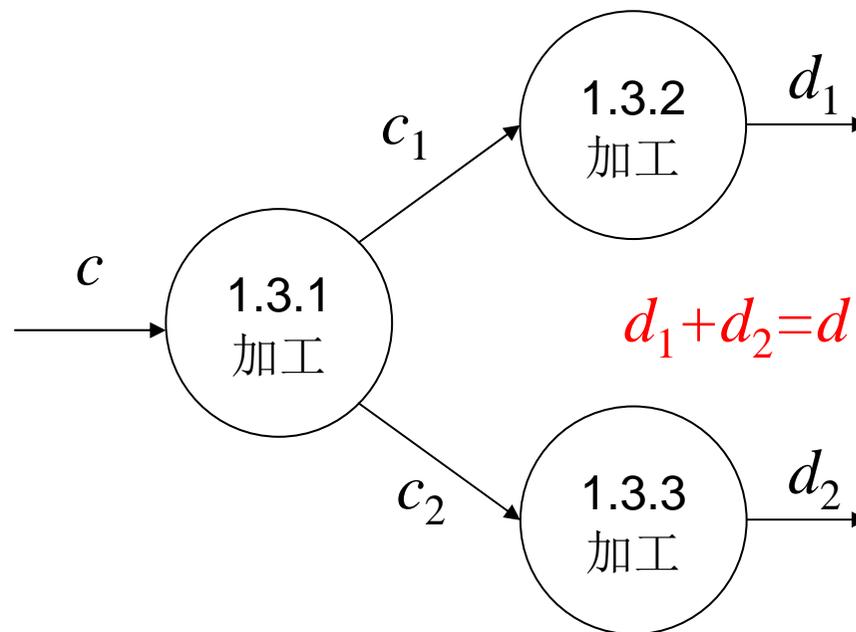
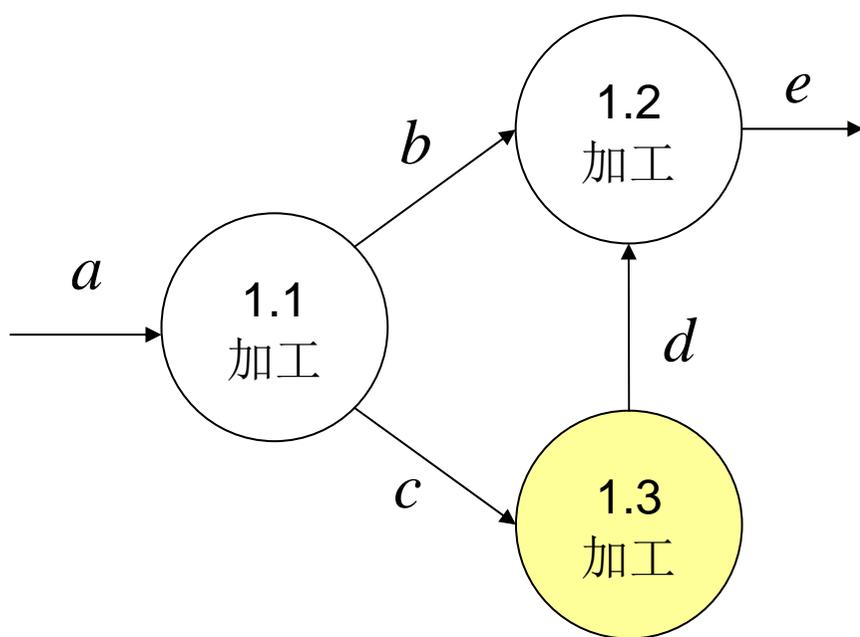
- 下层DFD中的输入输出数据流同上层DFD中相应加工的输入输出数据流必须一致，此即父图与子图的平衡。

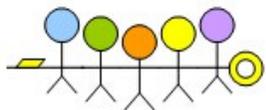




## 父图与子图的平衡

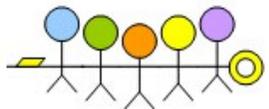
- 数据流本身可以分解，但其包含的数据内容应保持平衡



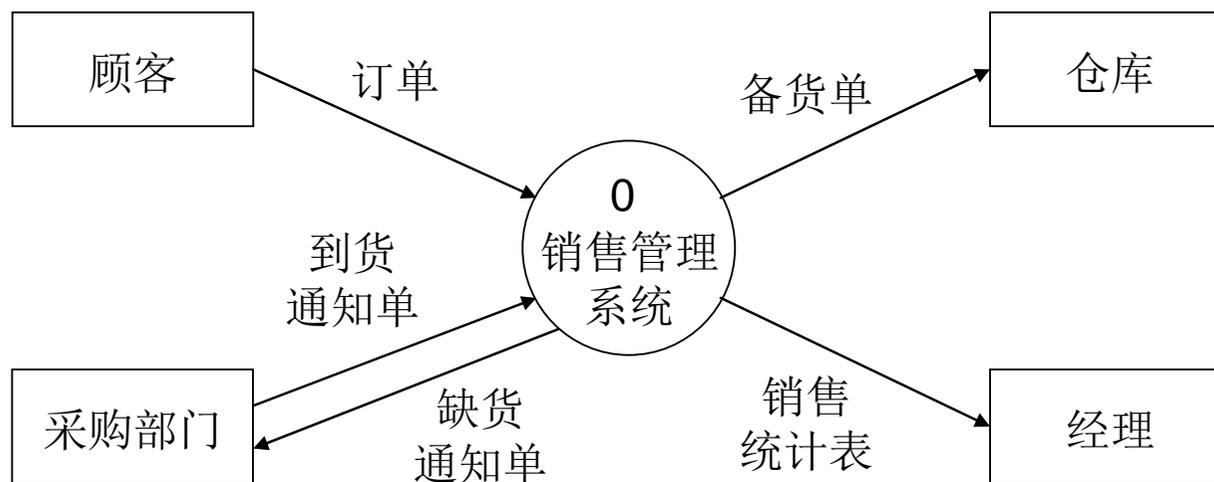


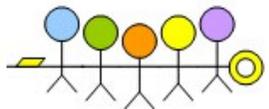
## 7.3.5 DFD实例：销售系统

- 某企业销售管理系统：
  - 接受顾客的订单，检验订单，若库存有货，进行供货处理，即修改库存，给仓库开备货单，并且将订单留底；若库存量不足，将缺货订单登入缺货记录。
  - 根据缺货记录进行缺货统计，将缺货通知单发给采购部门，以便采购。
  - 根据采购部门发来的进货通知单处理进货，即修改库存，并从缺货记录中取出缺货订单进行供货处理。
  - 根据留底的订单进行销售统计，打印统计表给经理。
  
- 绘制上述系统的顶层、0层、1层DFD图

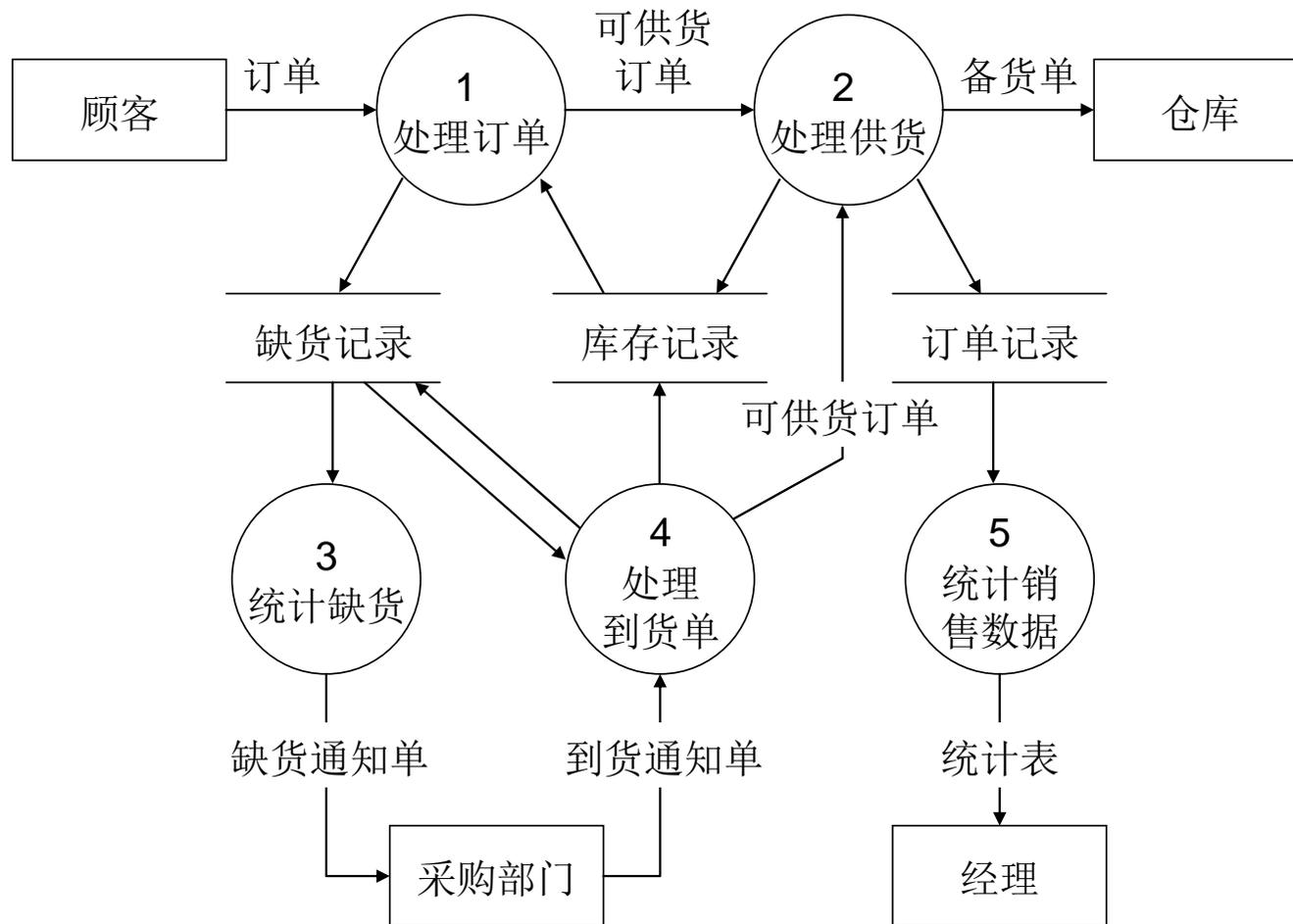


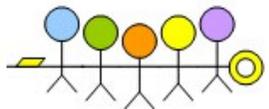
# 销售系统顶层DFD



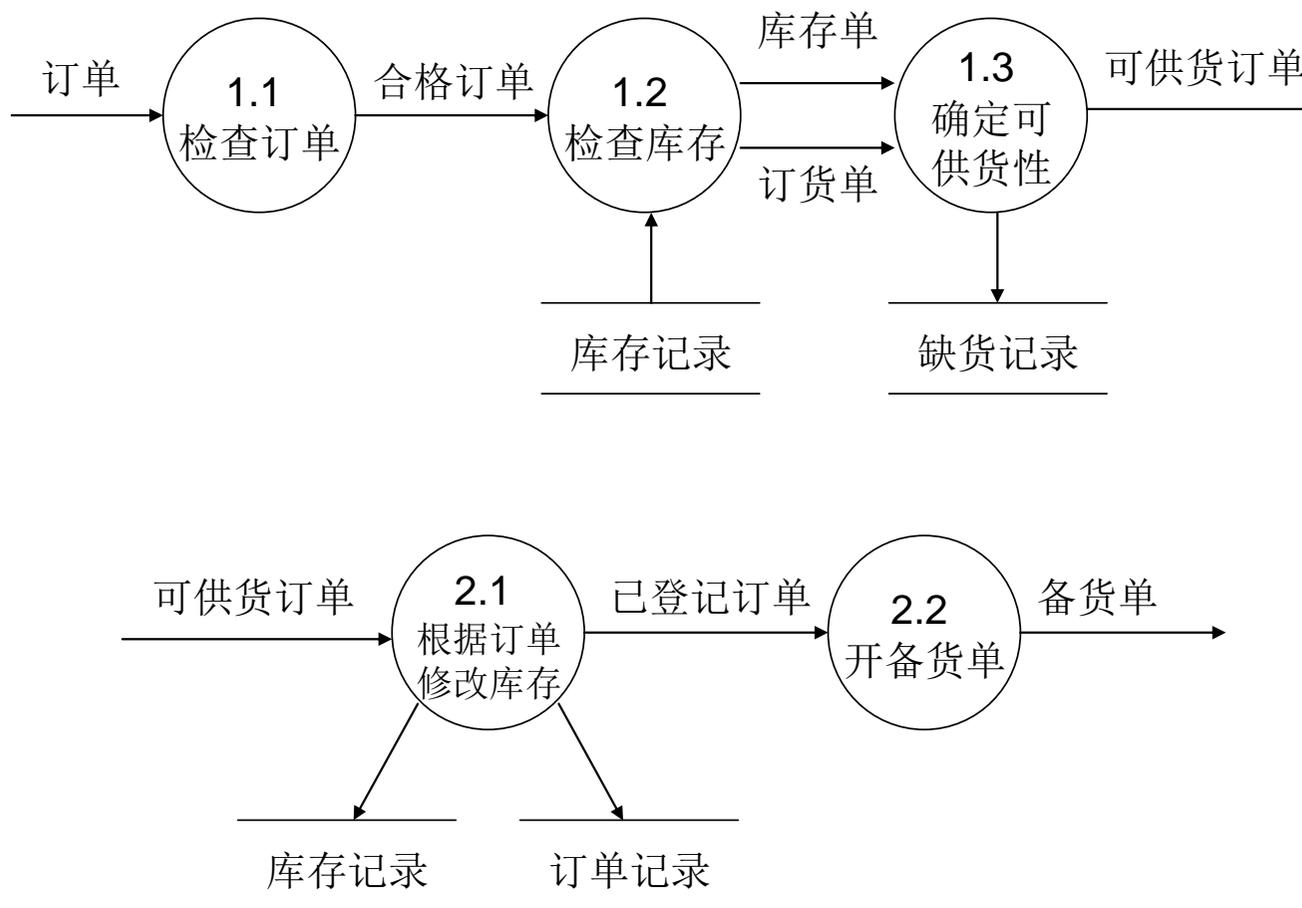


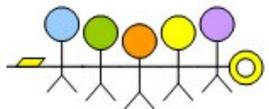
# 销售系统0层DFD



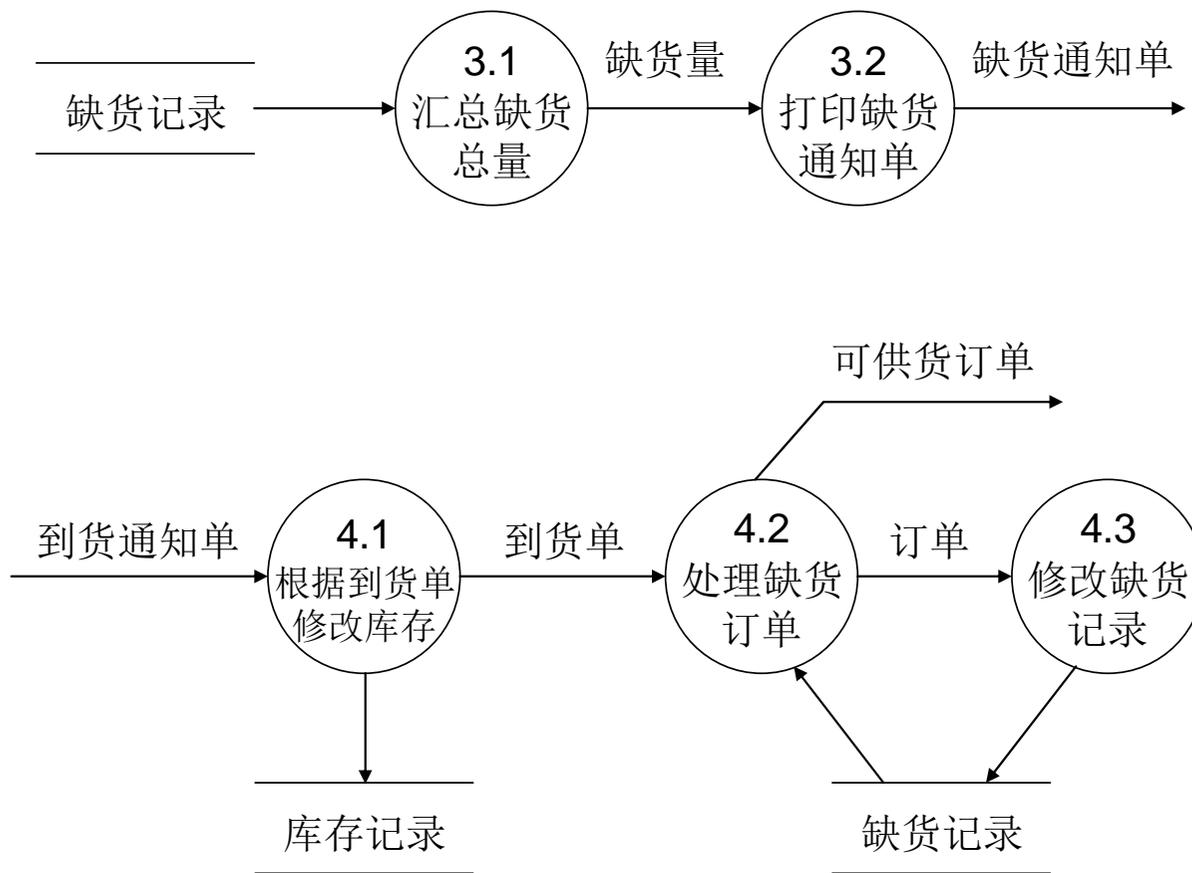


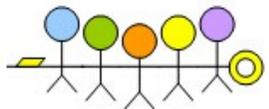
# 销售系统1层DFD



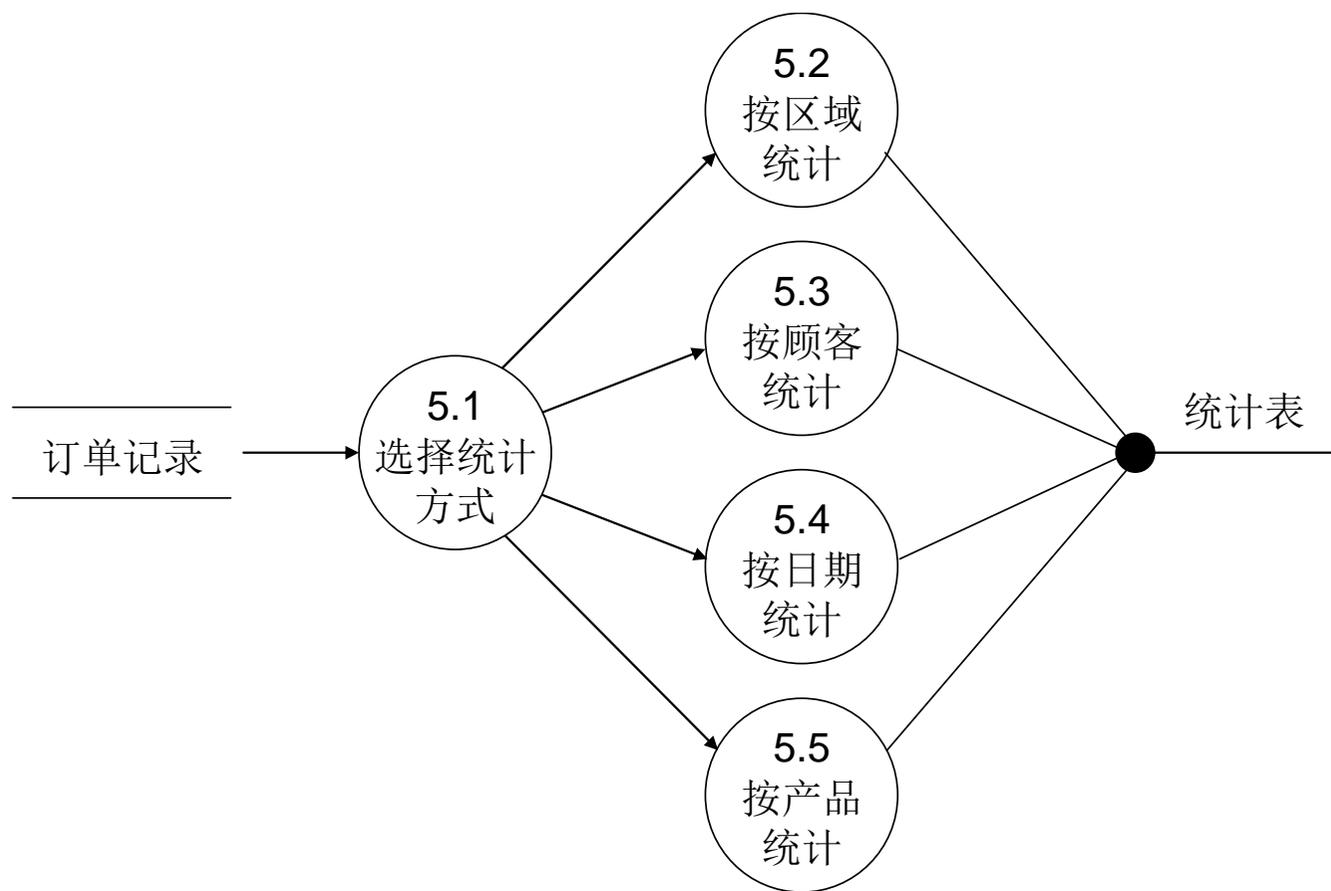


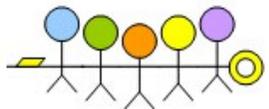
# 销售系统1层DFD



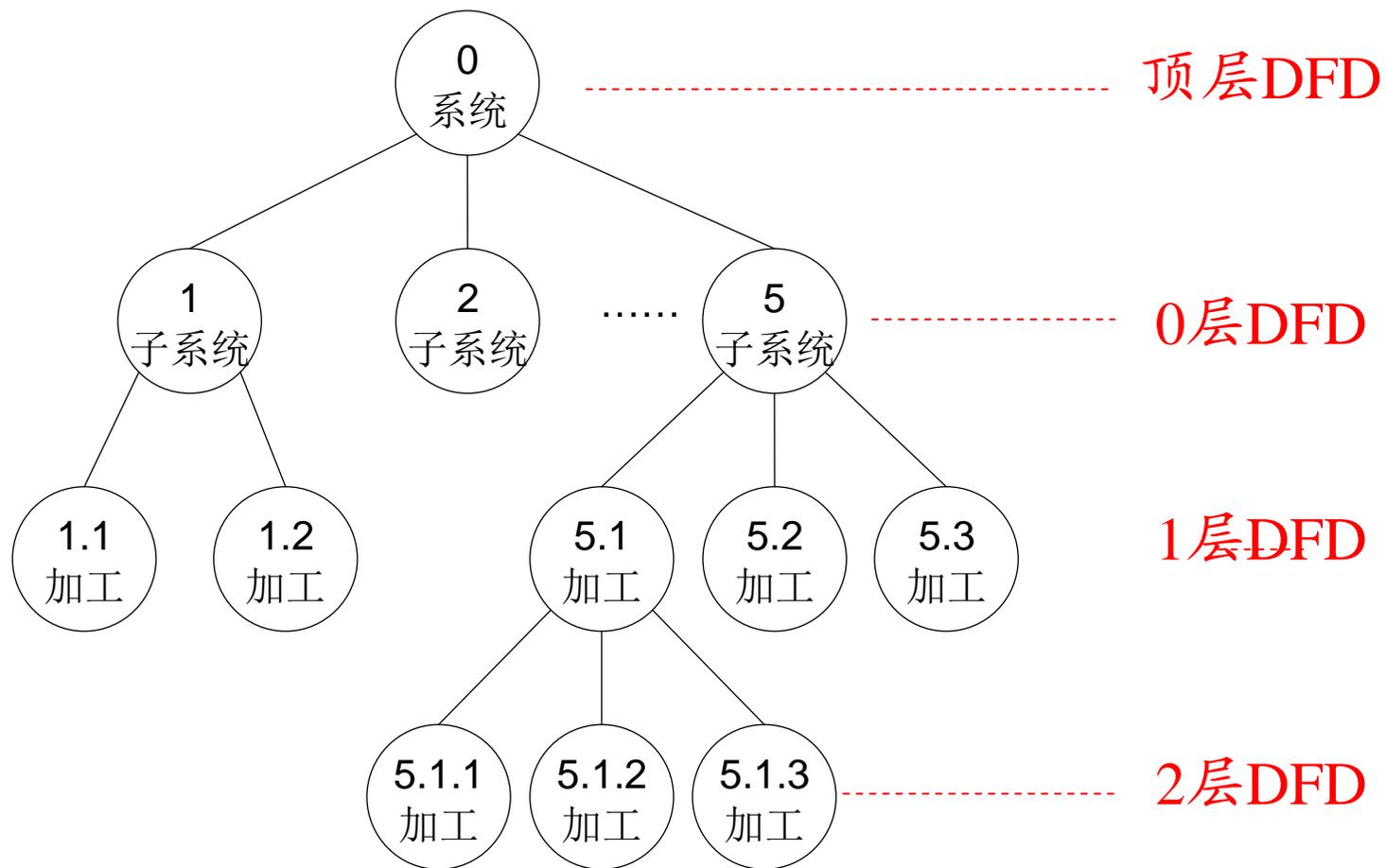


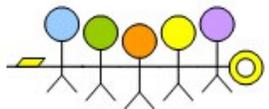
# 销售系统1层DFD





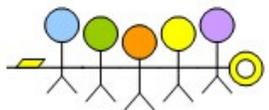
## 7.3.6 DFD树





## 7.4 数据字典(DD)

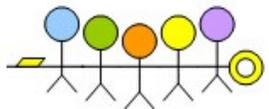




## 数据字典(DD)

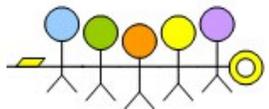
- DFD只是绘制了系统各功能之间的数据流动和处理关系，还需进一步考虑各数据的具体内容。
- 采用数据字典(Data Dictionary)作为描述工具
- 对于DFD中出现的所有被命名的图形元素(数据流、数据项、数据存储、加工)在DD中作为一个词条加以定义，使得每一个图形元素的名字都有一个确切的解释。
  - DD所有的定义应是严密的、精确的，不可有半点含混，不可有二义性。





## 数据字典的格式

符号	含义	示例
=	被定义为	
+	与	<b>x=a+b</b>
[... ...]	或	<b>x=[a b]</b>
$\left. \begin{matrix} \{ \dots \} \\ m \\ n \end{matrix} \right\}$	重复(最多m次, 最少n次)	$\left. \begin{matrix} \mathbf{x}=\{\mathbf{a}\} \\ 5 \\ 2 \end{matrix} \right\}$
{...}	重复	<b>x={a}</b>
(...)	可选	<b>x=(a)</b>
“...”	基本数据元素	<b>x=“a”</b>
..	连接符	<b>x=1..9</b>

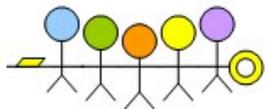


## 数据字典的格式：示例1

国内机票=姓名+身份证号+日期+航班号+起点+终点+费用

姓名=[ {字母}<sup>18</sup><sub>1</sub> | {汉字}<sup>5</sup><sub>2</sub> ]

航班号=[CA|MU|CZ|SC|FM] + “1000”..“9999”



## 数据字典的格式：示例2

存折=户名+开户行号+账号+开户日期+性质+(印密)+存取记录

户名={字母}

开户行号="001".."999"

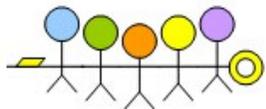
账号="00000001".."99999999"

开户日期=年+月+日

性质="1".."6"

印密="0"

存取行=日期+(摘要)+支出+存入+余额+操作员+复核员



## 数据项的定义

- 数据项：不可再分解的数据单位，包括：
  - 名称
  - 描述
  - 数据类型
  - 长度(精度)
  - 取值范围及缺省值
  - 计量单位
  - 相关数据元素及数据结构

### 示例

数据项名称：产品编码

别名：Product\_No

简述：用来唯一标识产品的文字

类型：字符串

长度：10

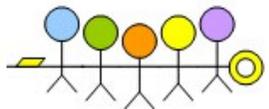
取值范围及含义：

第1位：进口/国产

第2-4位：类别

第5-7位：规格

第8-10位：品名编号



## 数据流的定义

- 数据流来源
- 数据流去向
- 数据流的组成
- 流动属性描述：
  - 频率、数据量等

### 示例

数据流名称：订单

别名：无

简述：顾客订货时填写的项目

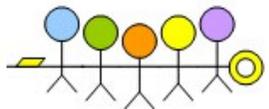
来源：顾客

去向：加工1“检验订单”

数据流量：1000份/每周

组成：编号+订货日期+顾客编号+地址+电话+  
          银行账号+货物名称+规格+数量

频率：平均40条/天



## 数据存储的定义

- 文件名
- 描述
- 数据结构
- 数据存储方式
- 关键码
- 存取频率和数据量
- 安全性要求

### 示例

数据存储名称：库存记录

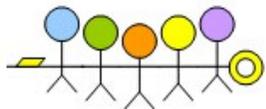
别名：无

简述：存放库存所有可供产品的信息

组成：产品编号+名称+生产厂家+单价+库存量

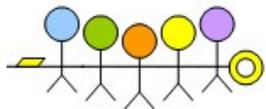
组织方式：索引文件，以产品编号为关键字

查询要求：要求能随时查询



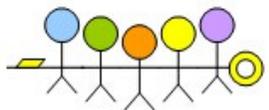
## 示例：数据字典(1)

- 某酒店所提供的电话服务系统的功能为：
  - 客人可以通过拨分机号联络酒店内的其他房间，也可拨外线号码与酒店外联络。
  - 分机号从8201至8299。
  - 外线号码需先拨0，然后加拨市话号码或长途电话号码。
  - 长途电话号码由区号和市话号码组成，其中区号可以为010、021~029、0300~0999中的任意一个数字串。
  - 市话号码是任意7位或8位长度的数字串。



## 示例：数据字典(1)

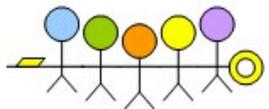
- “电话号码”的数据字典：
  - 电话号码=[分机号|外线号码]
  - 分机号=8201..8299
  - 外线号码=0+[市话号码|长途电话号码]
  - 市话号码={数字}
  - 长途电话号码=区号+市话号码
  - 区号=[010|021..029|0300..0999]



## 示例：数据字典(2)

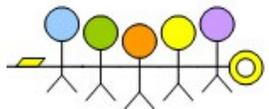
名字： 零件编号  
别名： 编码  
描述： 唯一的标识库存清单中一个特定零件  
定义： 零件编号={字符}<sup>8</sup>  
位置： 库存清单  
        采购订单  
        订货报表

名字： 采购订单  
别名： 采购单  
描述： 由各部门采购人员定期向供应商发出的单据  
定义： 采购订单=供应商ID+供应商名称+采购日期+总金额+  
        {零件编号+零件名称+数量+价格}  
位置： 订货报表



## 7.5 结构化语言、判定表(树)

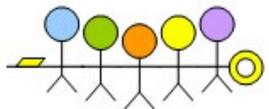




## 还剩下什么没有描述？

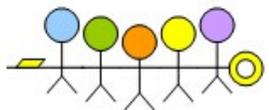
- **DFD**: 描述数据在功能模块之间的流动；
- **DD**: 描述数据的具体格式；
- 还剩下什么？

——“数据”是如何被“加工”所处理的？



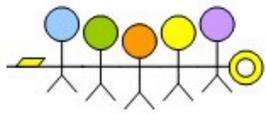
## 处理逻辑的描述方式

- 结构化语言(Structured English)
- 判定表 (Decision Table)
- 判定树 (Decision Tree)



## 结构化语言

- 介于自然语言和形式语言之间的一种半形式语言，类似于算法设计时使用的“伪代码”(Pseudo Code)
- 在自然语言基础上加了一些限定，使用有限的词汇和有限的语句来描述加工逻辑
- 可分成外层和内层两层：
  - 外层：用来描述控制结构，采用顺序、分支、循环三种基本结构
  - 内层：一般是采用祈使语句的自然语言短语，使用数据字典中的名词和有限的自定义词



## 结构化语言示例

**IF customer does more than \$50,000 business**

**THEN**

**IF the customer wasn't in debt to us the last 3 months**

**THEN discount = 15%**

**ELSE (he was in debt to us)**

**IF customer has been with us for more than 20 years**

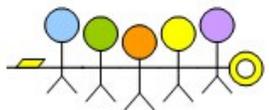
**THEN discount = 10%**

**ELSE (20 year or less)**

**THEN discount = 5%**

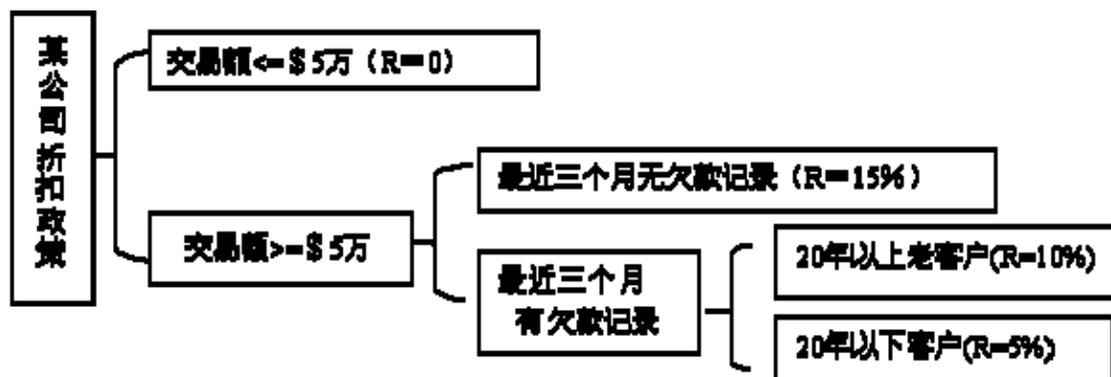
**ELSE (customer does \$50,000 OR Less)**

**THEN discount = 0**

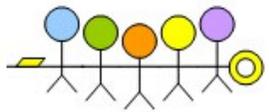


## 决策树与决策表

- 用决策树(decision tree)来描述一个加工的逻辑处理过程，其基本思路与结构化英语类似，但是更直观易懂。

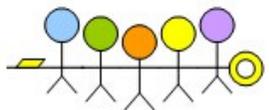


- 决策树(决策表)适用于描述具有复杂分支判断条件的加工处理过程。



## 7.6\* 实体联系图(E-R)



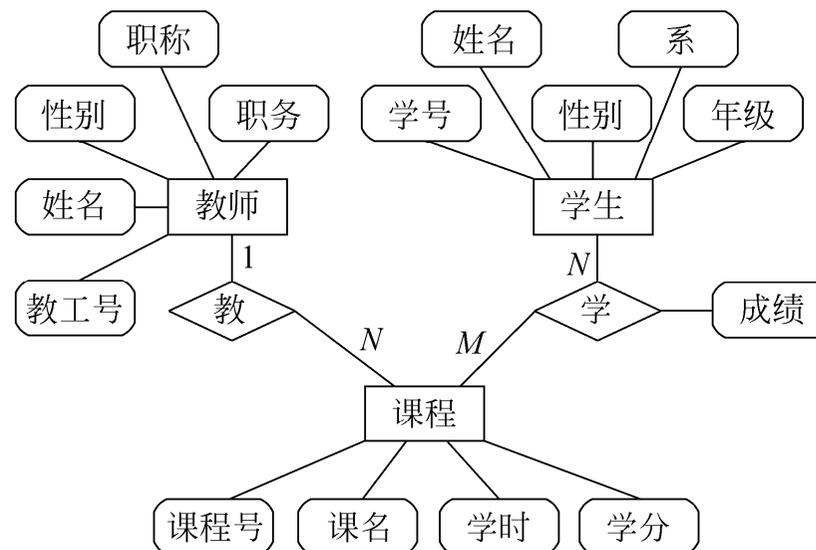


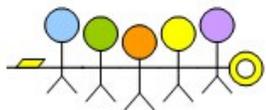
# 实体联系图(E-R)

## ■ 实体联系图(Entity-Relationship Diagram)

- 目标：把用户的数据要求清除、准确的描述出来，需建立一个概念性的数据模型(或称信息模型)。
- 形式：系统中包含哪些数据实体(entity)、每个实体所具备的主要属性(attribute)、这些实体之间存在哪些关联关系(relationship)；

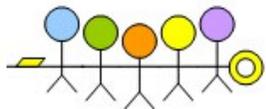
关于E-R的细节，  
在《数据库系统》课程中学习





## 实体(Entity)

- **E-R图中的“实体”(entity)也称“数据对象”(Data Object)，是对软件必须理解的复合信息的抽象。**
  - 所谓复合信息是指具有一系列不同性质或属性的事物，仅有单个值的事物(例如，宽度)不是数据对象。
- **可以由一组属性来定义的实体都可以被认为是数据对象：**
  - 外部实体：产生或使用信息的任何事物；
  - 事物：报表；
  - 行为：打电话
  - 事件：响警报
  - 角色：教师、学生
  - 单位：会计科
  - 地点：仓库
  - 结构：文件



## 实体的属性(Attribute)

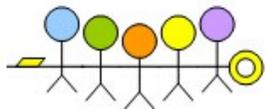
- 属性(attribute)定义了数据对象的性质;
- 实体的某一个或多个属性被定义为“标识符”(identifier), 作为“关键字”(通常简称为“键”);
- 根据对所要解决的问题的理解, 来确定特定数据对象的一组合适的属性。
- 例如“汽车”实体:
  - 生产厂商、品牌、型号、发动机号码、车体类型、颜色、车主姓名、住址、驾驶证号码、生产日期、购买日期、等;



## 联系 (Relationship)

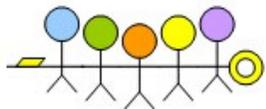
- 数据对象彼此间是有关联的：
  - 教师“教”课程，学生“学”课程；
  - “教”或“学”的关系表示教师和课程或学生和课程之间的一种特定的连接。
- 数据对象彼此之间相互连接的方式称为联系(**relationship**)，也称为关系(**relation**)；
- 分为**3**种类型的联系：
  - 一对一联系(1:1)
  - 一对多联系(1:n)
  - 多对多联系(m:n)

能否举出一些例子？



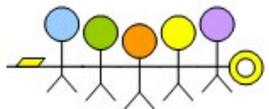
## 联系 (Relationship)

- **一对一联系(1:1)**
  - 一个部门有一个经理，而每个经理只在一个部门任职，则部门与经理的联系是一对一的；
- **一对多联系(1:n)**
  - 教师与课程之间存在一对多的联系“教”，即每位教师可以教多门课程，但是每门课程只能由一位教师来教；
- **多对多联系(m:n)**
  - 学生与课程间的联系(“学”)是多对多的，即一个学生可以学多门课程，而每门课程可以有多个学生来学。

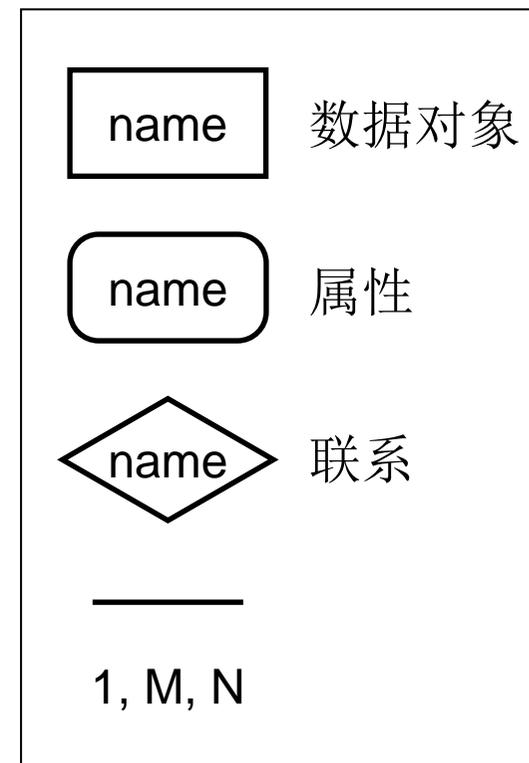
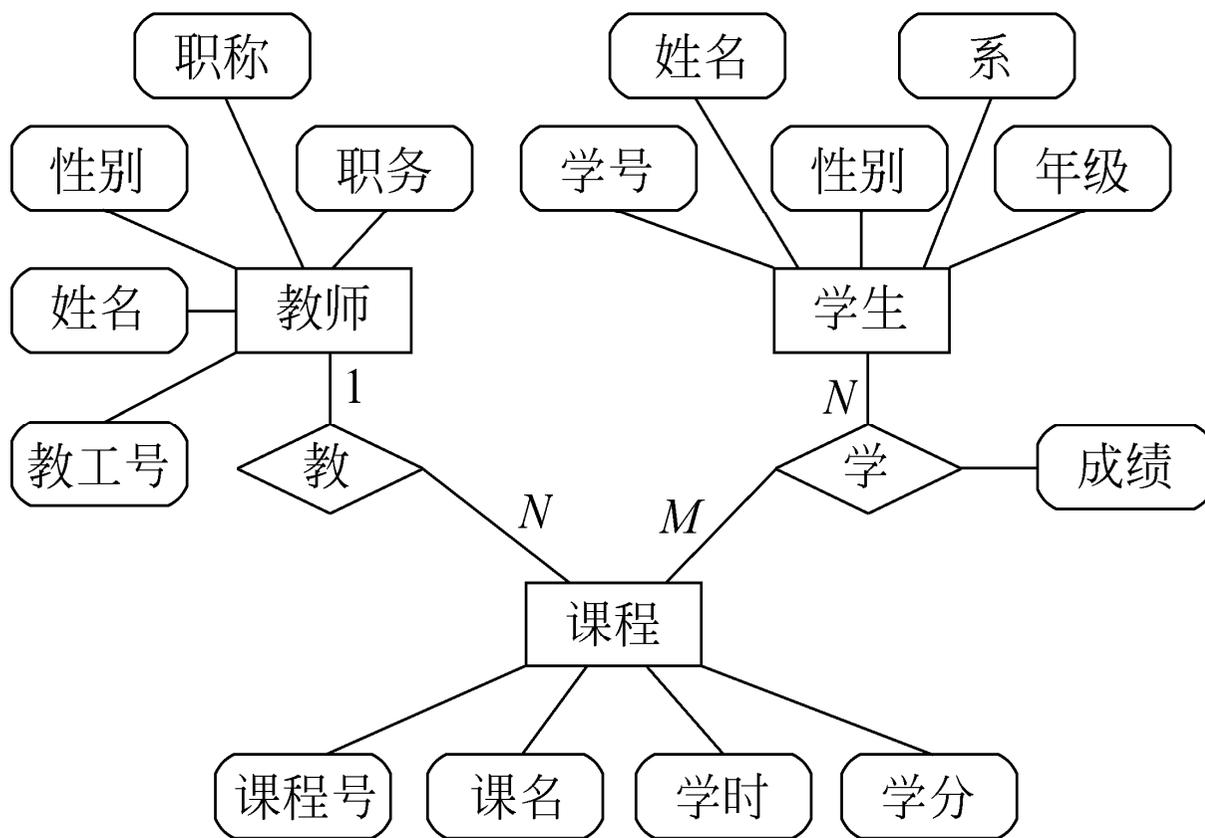


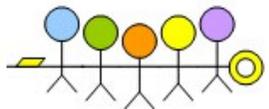
## 联系 (Relationship) 的属性

- 联系也可能有属性。
- 例如，学生“学”某门课程所取得的成绩，既不是学生的属性也不是课程的属性。由于“成绩”既依赖于某名特定的学生又依赖于某门特定的课程，所以它是学生与课程之间的联系“学”的属性



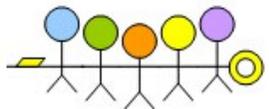
# E-R图的图形化符号





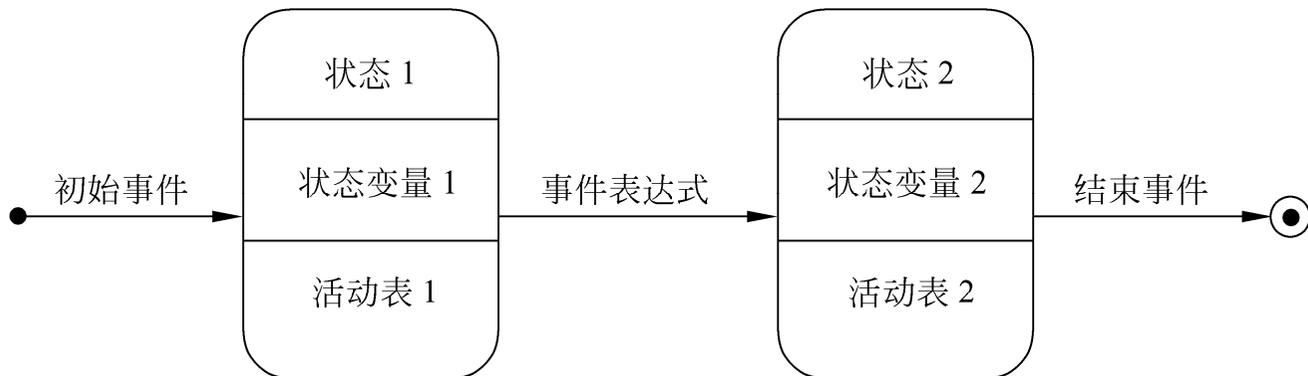
## 7.7\* 状态转换图

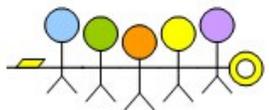




## 状态转换图

- **DFD+DD+结构化英语：功能模型**
- **E-R：信息模型**
- **需求分析阶段还需要建立起软件系统的行为模型：状态转换图(State Transition Diagram)**
  - 通过描绘系统的状态及引起系统状态转换的事件，来表示系统的行为；
  - 指明了作为特定事件的结果系统将做哪些动作。



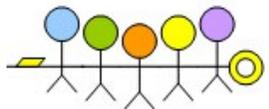


## 状态转换图的基本要素(1)

### ■ 状态(state)

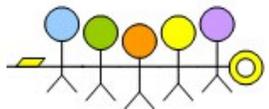
- 是任何可以被观察到的系统行为模式，一个状态代表系统的一种行为模式。
- 规定了系统对事件的响应方式。
- 系统对事件的响应，既可以是做一个(或一系列)动作，也可以是仅仅改变系统本身的状态，还可以是既改变状态又做动作。
- 在状态图中定义的状态主要有：初态(即初始状态)、终态(即最终状态)和中间状态。
- 在一张状态图中只能有一个初态，而终态则可以有0至多个。

### ■ 事件(event)

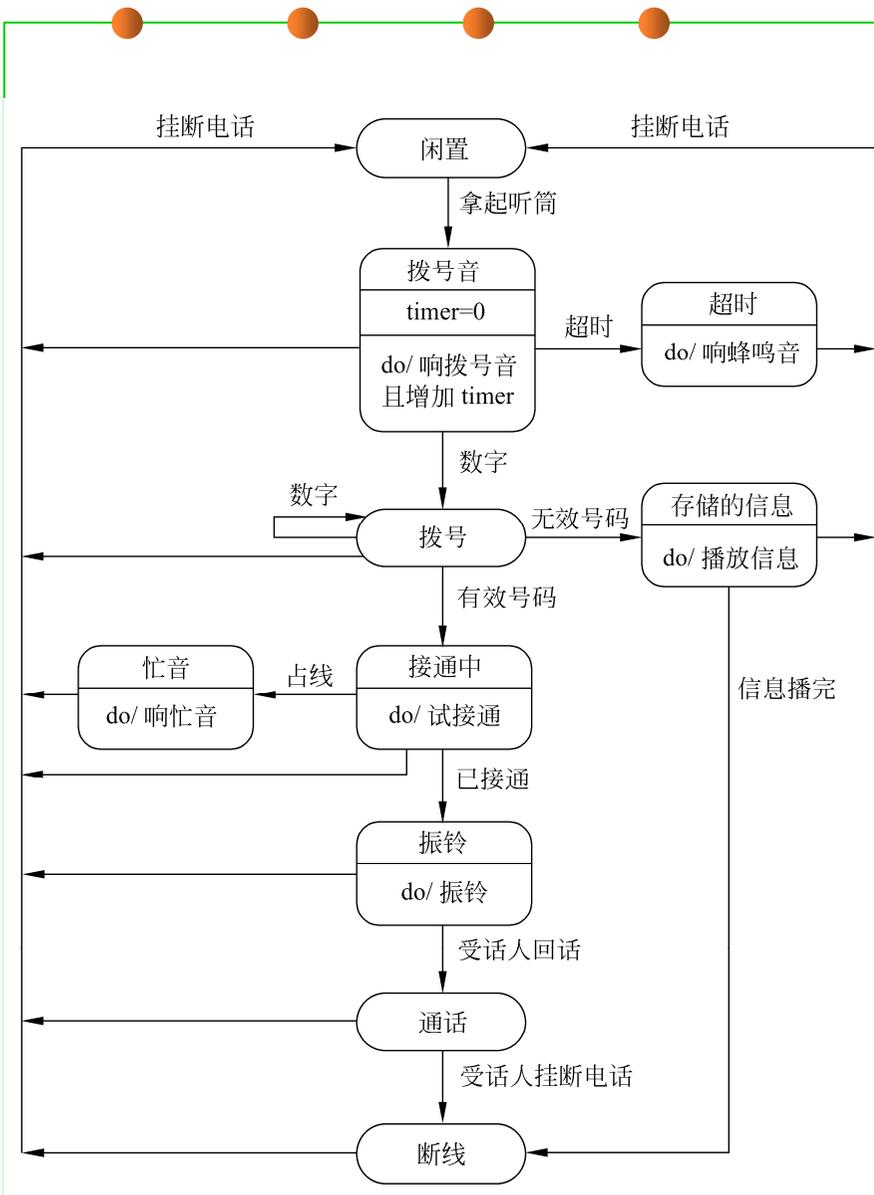


## 状态转换图的基本要素(2)

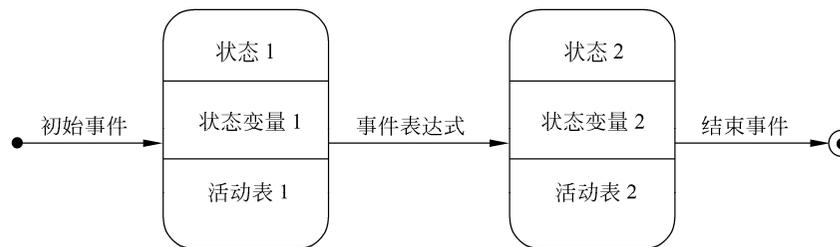
- **状态(state)**
- **事件(event)**
  - 在某个特定时刻发生的事情，它是对引起系统做动作或(和)从一个状态转换到另一个状态的外界事件的抽象。
  - 例如，内部时钟表明某个规定的时间段已经过去，用户移动或点击鼠标等都是事件。
  - 简而言之，事件就是引起系统做动作或(和)转换状态的控制信息。

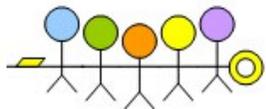


# 状态转换图的图形化符号



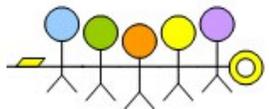
- 无人打电话时电话处于闲置状态;
- 有人拿起听筒则进入拨号音状态，到达这个状态后，电话的行为是响起拨号音并计时;
- 这时如果拿起听筒的人改变主意不想打了，他把听筒放下(挂断)，电话重又回到闲置状态;
- 如果拿起听筒很长时间不拨号(超时)，则进入超时状态;
- ...





## 状态转换图示例

- 复印机的工作过程为：
  - 未接受到复印命令时处于闲置状态，一旦接受到复印命令则进入到复印状态；
  - 完成一个复印命令规定的工作后又回到闲置状态，等待下一个复印命令；
  - 如果执行复印命令时发现缺纸，则进入缺纸状态，发出警告，等待装纸；
  - 装满纸后进入闲置状态，准备接收复印命令；
  - 如果复印时发生卡纸故障，则进入卡纸状态，发出警告等待维修人员来排除故障；
  - 故障排除后回到闲置状态。



## 状态转换图示例

