

# 第6章.过程建模

---

# 主要内容

1. 过程建模
  2. 数据流图DFD
  3. 微规格说明
  4. 数据字典
  5. 模块结构图
-

# 1. 过程建模

## ——结构化建模

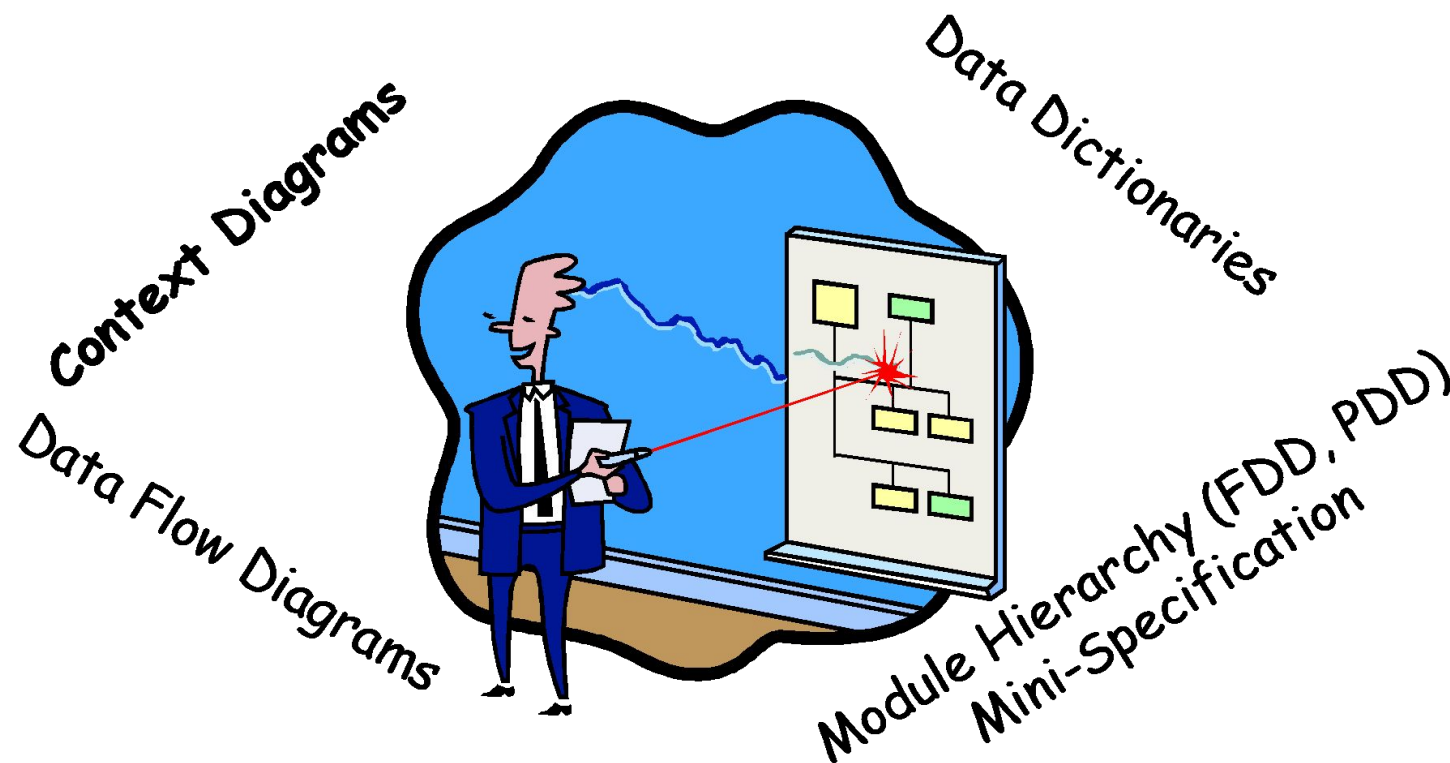
- 怎么理解复杂世界？
  - 复杂—>简单（分解）
    - 简单—>可理解性（最基本单位）
    - 简单（高内聚）
    - 简单 **VS** 简单（低耦合）
  - 简单—>复杂（接口和实现）
- 结构化建模
  - 复杂世界—>复杂处理过程（事情的发生发展）
  - 简单—>过程（可表达的“函数”）
    - 软件“函数”、程序
  - 复杂→简单
    - 功能分解结构
  - 简单—>复杂（函数调用）

---

# 1. 过程建模

- 结构化建模
    - 过程建模，数据建模
  - 过程建模是结构化建模的核心方法
    - 系统是过程的集合
    - 过程可以被分解为子过程
    - 最终的子过程可以被映射为计算实体——函数
    - 所有的系统都是由过程构建的
  - 主要思想
    - 用图形的方式建立和描述良好的过程模型
-

# 1. 过程建模



---

# 主要内容

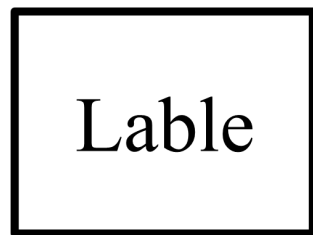
1. 过程建模
  2. 数据流图DFD
    1. 基本元素
    2. 规则
    3. 层次结构
    4. 层次结构的建立
    5. DFD的验证
  3. 微规格说明
  4. 数据字典
  5. 模块结构图
-

## 2.1 DFD的基本元素

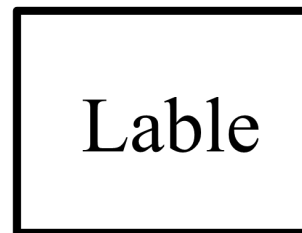
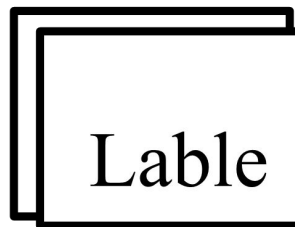
### ■ 外部实体

- 外部实体是指处于待构建系统之外的人、组织、设备或者其他软件系统，它们不受系统的控制，开发者不能以任何方式操纵它们
- 需要进行建模的外部实体是那些和待构建的软件系统之间存在着数据交互的外部实体，它们是待构建系统的数据源或者数据目的地
- 所有的外部实体联合起来构成了软件系统的外部上下文环境

DeMarco-Yourdon



Gane-Sarson

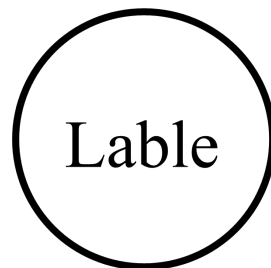


## 2.1 DFD的基本元素

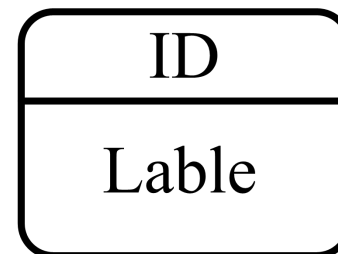
### ■ 过程

- 过程是指施加于数据的动作或者行为，它们使得数据发生变化，包括被转换（**transformed**）、被存储（**stored**）或者被分布（**distributed**）
- 可能是由软件系统控制的，也可能是由人工执行的，它重在数据发生变化的效果而不是其执行者
- 可能会表现为不同的抽象层次
  - 内容足够细节和具体，能够对其直接进行“编码”处理的过程被称为原始过程（**Primitive Process**，又称为基本过程**Elementary Process**）

DeMarco-Yourdon



Gane-Sarson





## 2.1 DFD的基本元素

### ■ 数据流

- 数据流是指数据的运动，它是系统与其环境之间或者系统内两个过程之间的通信形式

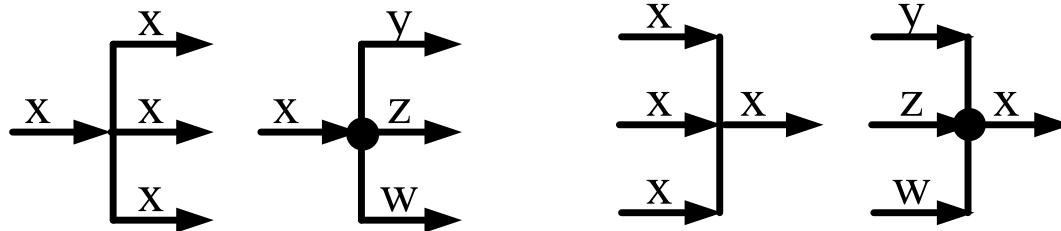
DeMarco-Yourdon

——Lable——→

Gane-Sarson

——Lable——→

- 数据流可以分割和组合



- 数据字典和ERD通常被用来描述DFD数据流的详细内容

## 2.1 DFD的基本元素

### ■ 数据存储

- 数据存储是软件系统需要在内部收集、保存，以供日后使用的数据集
- 数据存储的详细内容通常也是用数据字典和ERD来进行描述的

DeMarco-Yourdon

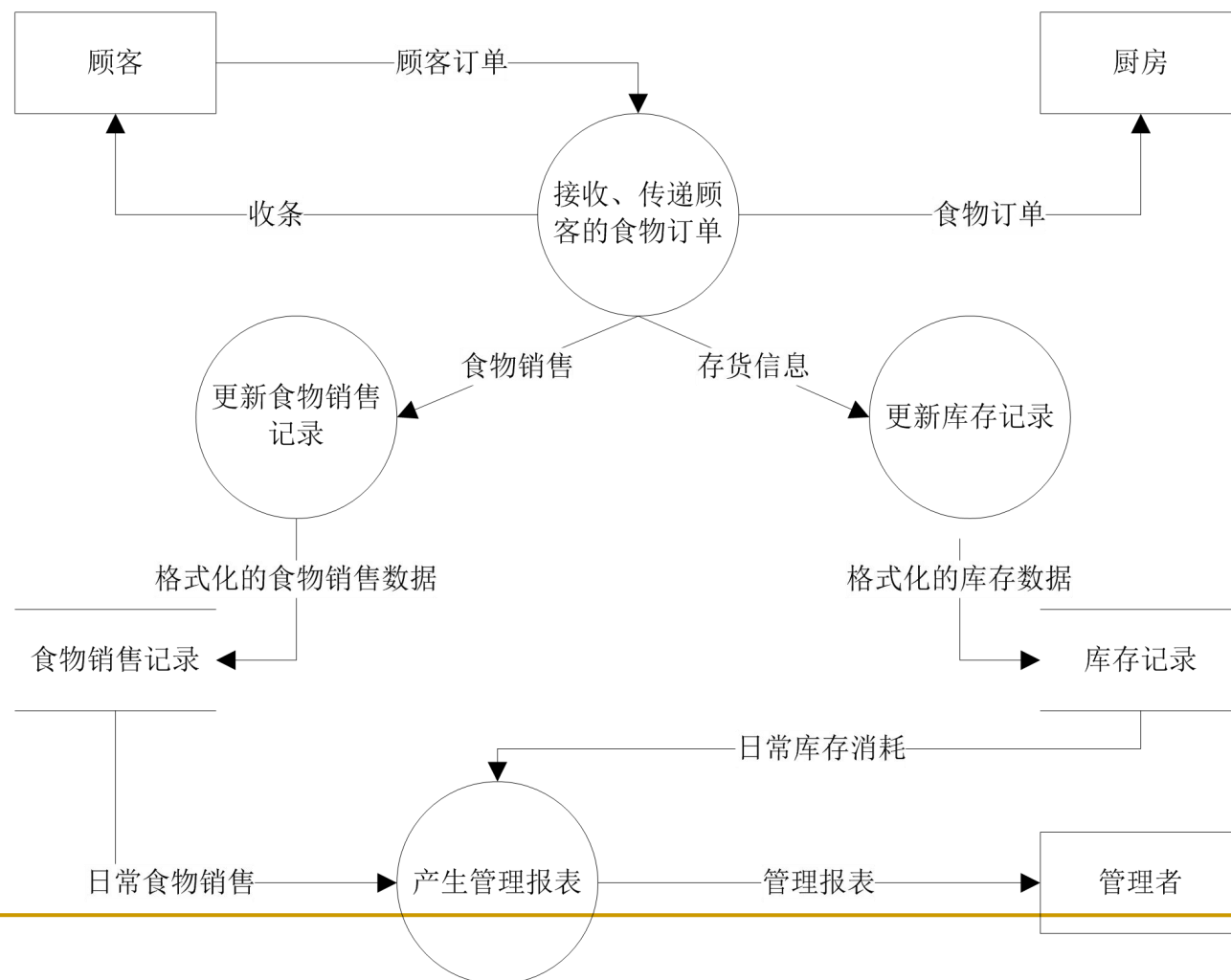


Gane-Sarson



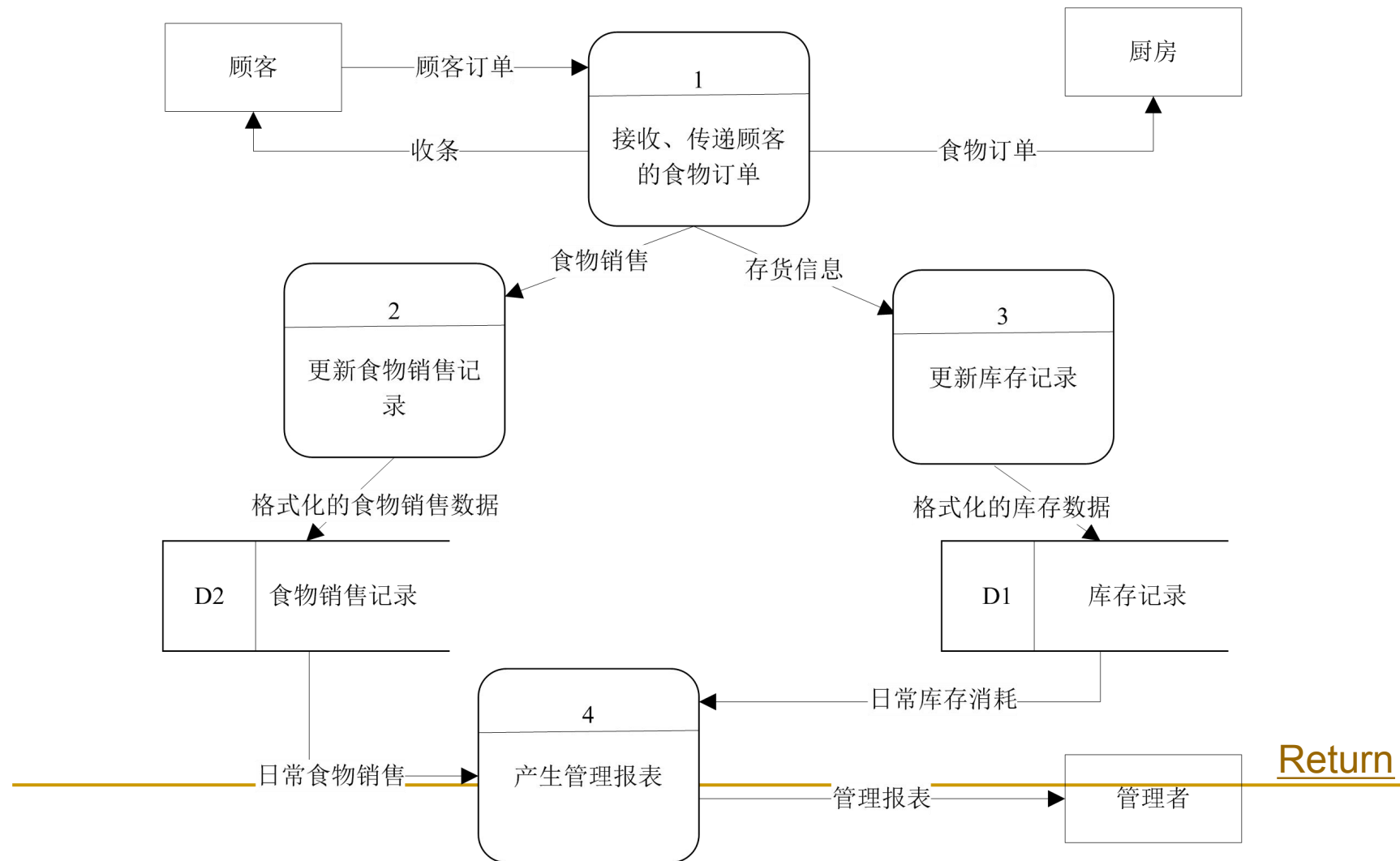
## 2.1 DFD的基本元素

### ——示例： DeMarco



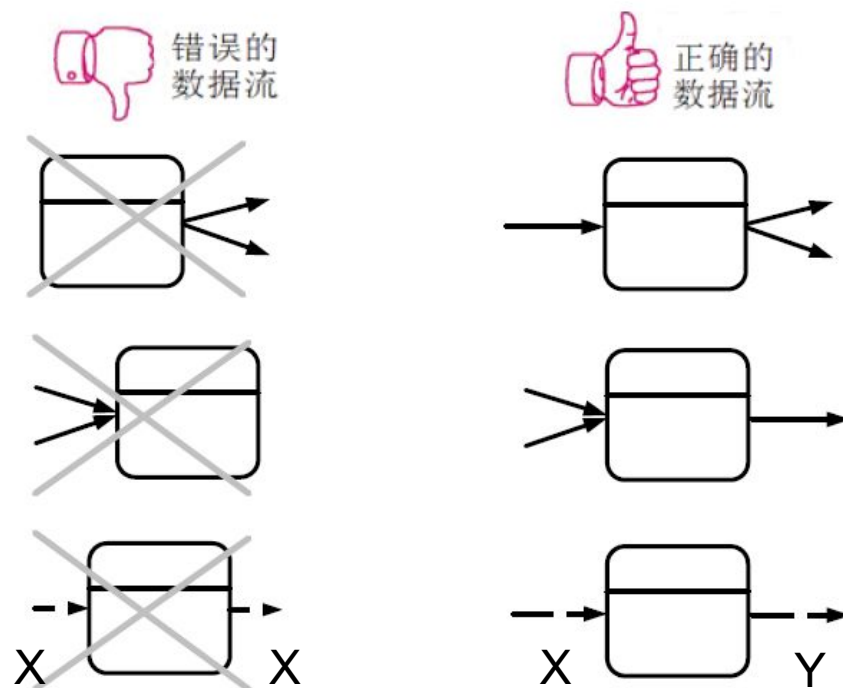
## 2.1 DFD的基本元素

### ——示例：Gane-Sarson



## 2.2 规则

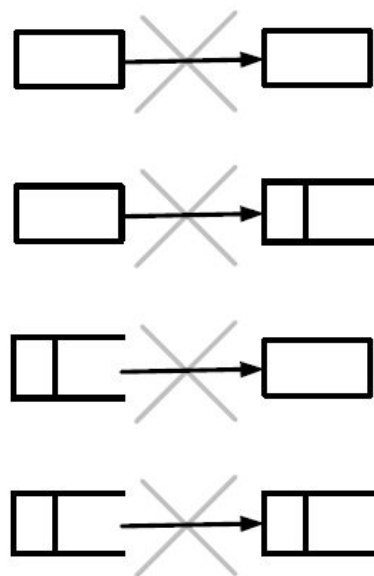
- 过程是对数据的处理，必须有输入，也必须有输出，而且输入数据集和输出数据集应该存在差异



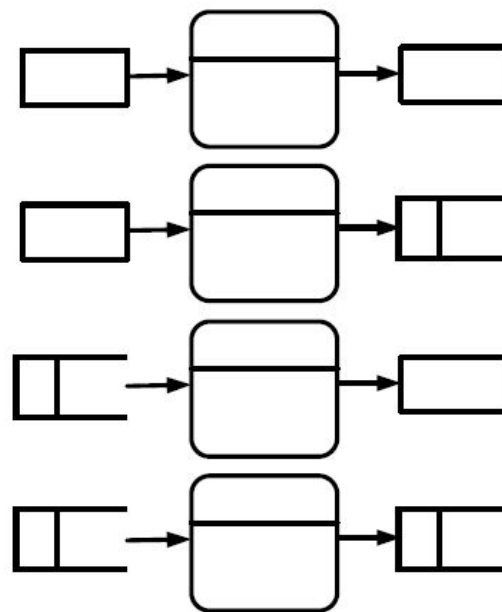
## 2.2 规则

- 数据流是必须和过程产生关联的，它要么是过程的数据输入，要么是过程的数据输出

 错误的  
数据流



 正确的  
数据流



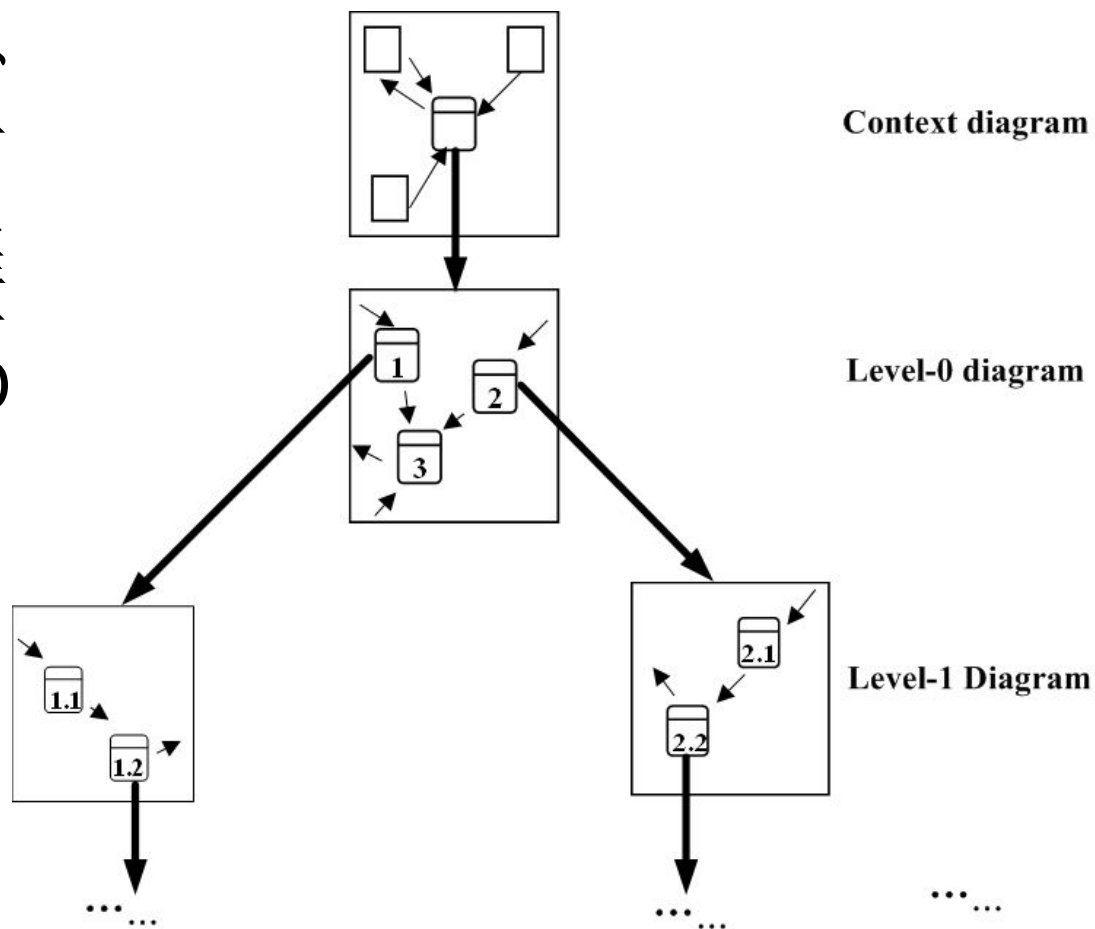
---

## 2.2 规则

- DFD当中所有的对象都应该有一个可以唯一标识自己的名称。
    - 过程使用动词
    - 外部实体、数据流和数据存储使用名词
-

## 2.3 层次结构

- 依据所含过程的不同抽象程度，**DFD**可以在不同的抽象层次上进行系统的描述
- 一个比较抽象的过程可以被展开为一个子过程更加具体的**DFD**图
- **DFD**的层次结构
  - 上下文图
  - 0层图
  - N层图 ( $N>0$ )

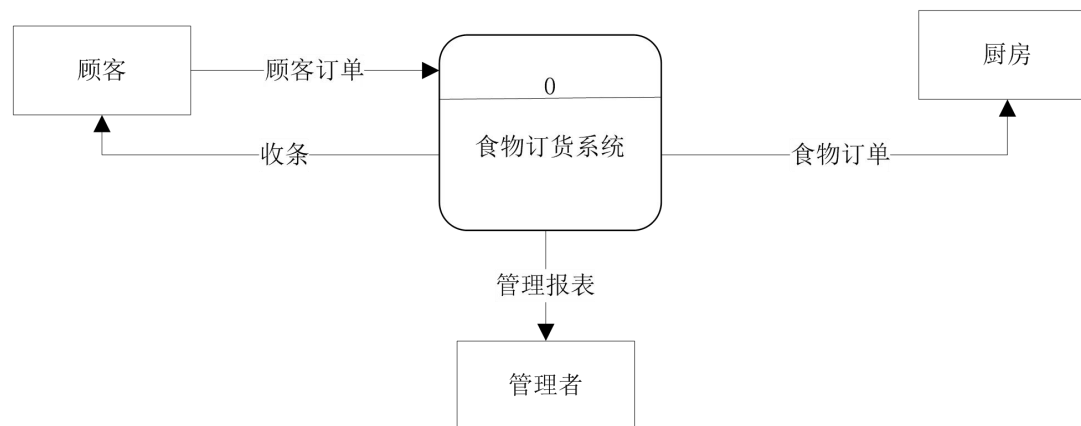




## 2.3 层次结构

### ■ 上下文图

- 将整个系统看做是一个过程，这个过程实现系统的所有功能，是系统功能的最高抽象
  - 上下文图中存在且仅存在一个过程，表示整个系统。这个单一的过程通常编号为0
  - 上下文图中需要表示出所有和系统交互的外部实体，并描述交互的数据流，包括系统输入和系统输出
  - 上下文图中不会出现数据存储实例
- 它非常适合于描述系统的应用环境、定义系统的边界



---

## 2.3 层次结构

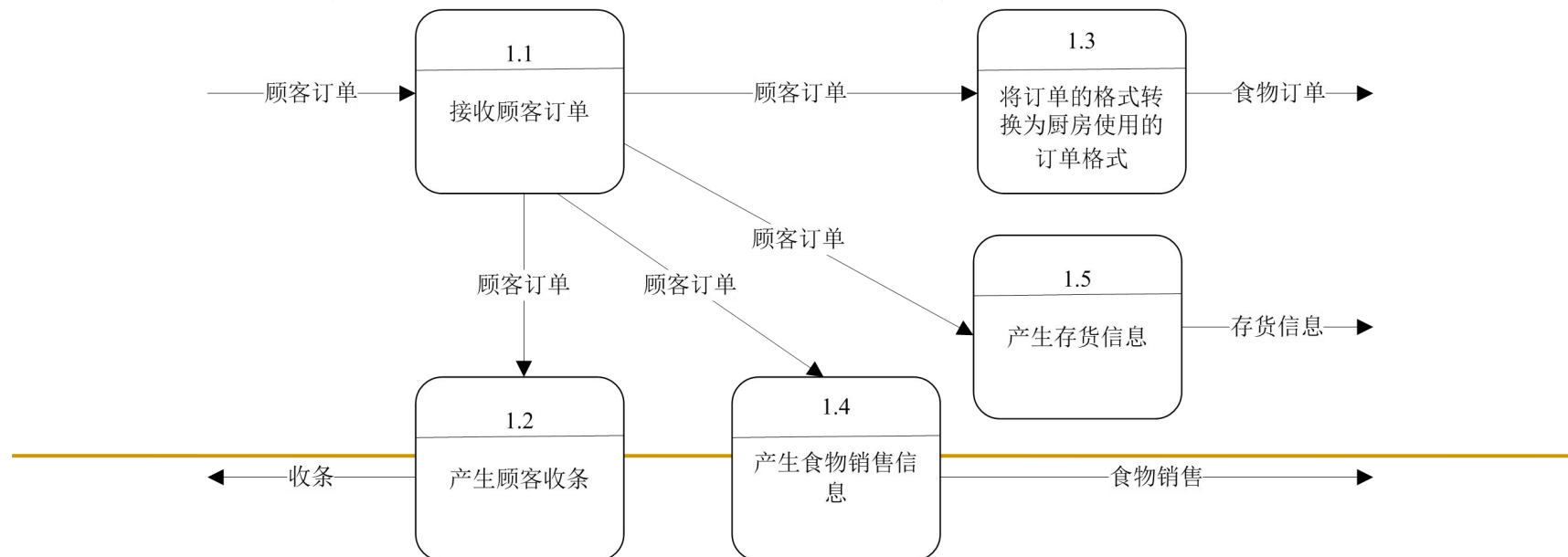
### ■ 0层图

- 位于上下文图下面一层，是上下文图中单一过程的细节描述，是对该单一过程的第一次功能分解
  - 是整个系统的功能概图
  - 0层图应该被描述的简洁、清晰，需求工程师要根据系统的复杂度掌握0层图中过程的抽象程度
  - 示例
-

## 2.3 层次结构

### ■ N层图

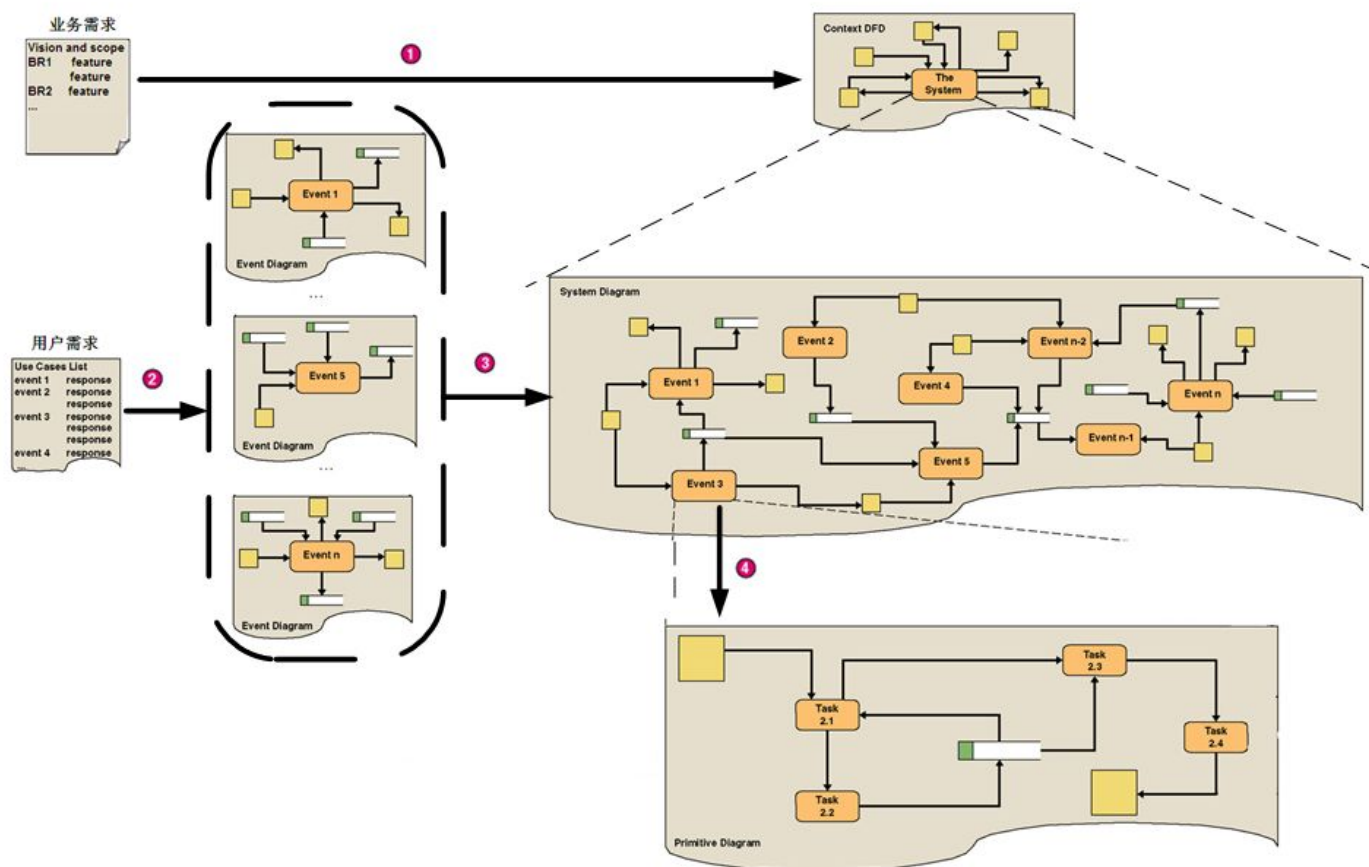
- 对0层图的过程分解产生的子图称为1层图，对N层图的过程分解后产生的子图称为N+1层图（ $N>0$ ），过程分解是可以持续进行的，直至最终产生的子图都是原始DFD图
- 原始DFD图可以进一步展开为
  - 微规格说明
  - 数据字典
- 在低于0层图的子图上通常不显示外部实体



## 2.4 层次结构的建立

### 建立步骤

1. 创建上下文图
2. 发现并建立DFD片段
3. 根据DFD片段组合产生0层图；
4. 对0层图的过程进行分解，产生N层图



---

## 2.4 层次结构的建立

### ——1. 创建上下文图

- 在需求获取阶段获得的业务需求以及业务需求所决定的项目前景与范围可以用来帮助建立系统的上下文图

---

## 2.4 层次结构的建立

### ——2. 发现并建立DFD片段

- **DFD**片断是系统对某个事件的响应过程的**DFD**描述，它是为系统中发生的重要事件创建的。
- 它将系统对事件的处理看做是一个单一的过程，重点描述这个单一过程与事件外界（包括系统内其他部分和系统外的外部实体）的数据流交互

---

## 2.4 层次结构的建立

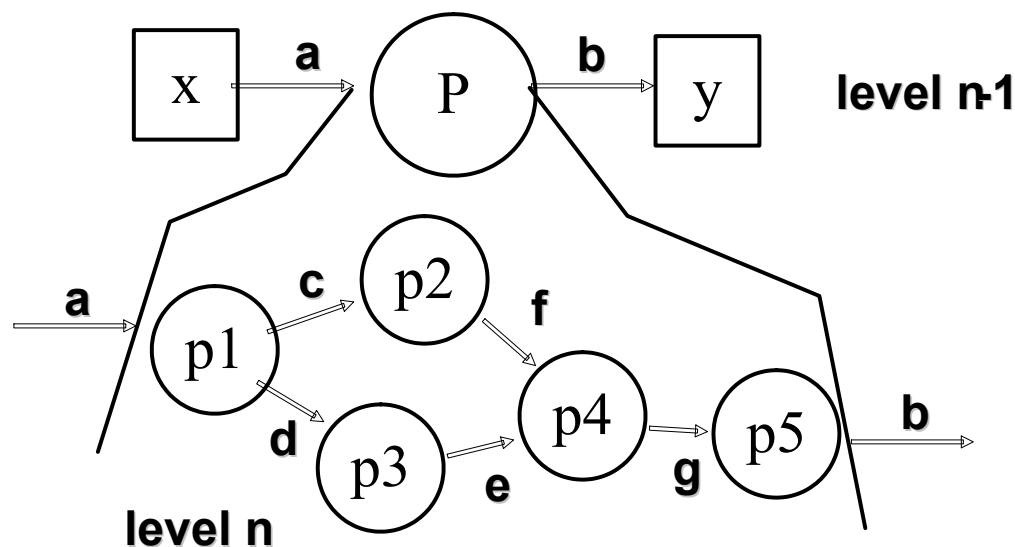
### ——3. 产生0层图

- 往往需要多次调整DFD片段的整合结果才能得出
  - 对DFD图（尤其是0层图）质量的判定有下面几个准则：
    - 1、没有语法错误，遵守12.2.2所述的各项规则。
    - 2、具有良好的语义，过程的功能设置要高内聚、低耦合。
    - 3、保持数据一致性，过程的输入流要足以产生数据输出。同时过程的输出流是在充分利用输入数据的基础上产生的，不存在输入数据的浪费。
    - 4、控制复杂度，不要一次在图中显示太多的信息。一般情况下，一个图中的过程数量最好控制在5~9（人脑的最佳信息处理量）个。而且图中的数据流数量越少越好，越简洁越好（接口最小化）。
-

## 2.4 层次结构的建立

### ——4. 功能分解产生N层图

- 功能分解是一个拆分功能的描述，将单个复杂的过程变为多个更加具体、更加精确和更加细节的过程
- 在功能分解过程当中，最重要的是要保证分解过程的平衡性（**Balance**），它要求DFD子图的输入流、输出流必须和父过程的输入流、输出流保持一致





---

## 2.4 层次结构的建立

### ——4. 功能分解产生N层图

- 在分解产生的子图为下述情景之一时，可以判定其为原始DFD图，此时应该停止持续的功能分解活动：
    - 所有过程都已经被简化为一个选择、计算或者数据库操作；
    - 所有数据存储都仅仅表示了一个单独的数据实体；
    - 用户已经不关心比子图更为细节的内容，或者子图的描述已经详细的足以支持后续的开发活动；
    - 每一个数据流都已经不需要进行更详细的切分，以展示对不同数据的不同处理方式；
    - 每一个业务表单、事务、计算机的屏幕显示（**computer on-line display**）和业务报表都已经被表示为一个单独的数据流；
    - 系统的每一个最低层菜单选项都能在子图中找到独立的过程。
-

---

## 2.4 层次结构的建立

### —— 示例

- 使用**DFD**描述常见的电梯控制系统。
    - 一个控制系统控制多个电梯。每个电梯被置于一个相应甬道之中，在卷扬电机的作用下在甬道内做上下运动。甬道内安装有多个传感器，通常每个电梯停靠点一个，用来感应电梯的实时位置。电梯内部和建筑的每个电梯停靠层都设置有指示器，用来告知用户的电梯实时位置和运动状况。电梯内和建筑的每个电梯停靠层都设有按钮，用户可以通过这些按钮提出服务申请并进出电梯。控制系统调度用户的申请，让电梯以最有效的方式满足用户的服务要求
-

## 2.4 层次结构的建立

### —— 示例：1. 建立上下文图

业务需求	实现业务需求需要的系统特性	局部解决方案的对外交互
BR1: 让电梯运转起来	SF1.1: 能够获知电梯位置感应, 并转交给指示器	外部输入: 感应器感知信号 外部输出: 指示器要求信号
	SF1.2: 能够控制卷扬电机, 实现服务请求的电梯运动	内部输入: 调度要求 外部输出: 卷扬电机控制信号
	SF1.3: 用户可以利用按钮发出服务请求	外部输入: 按钮信号 内部输出: 服务请求
BR2: 实现电梯的有效调度, 最大限度的服务用户	SF2.1: 系统从服务请求队列中建立高效率的调度	内部输入: 服务请求 内部输出: 调度要求
BR3: 保证安全, 尤其是用户出入电梯的安全	SF3.1: 系统要根据电梯的运动状况和服务申请控制电梯门的开关	内部输入: 服务请求、电梯状况 外部输出: 门控命令

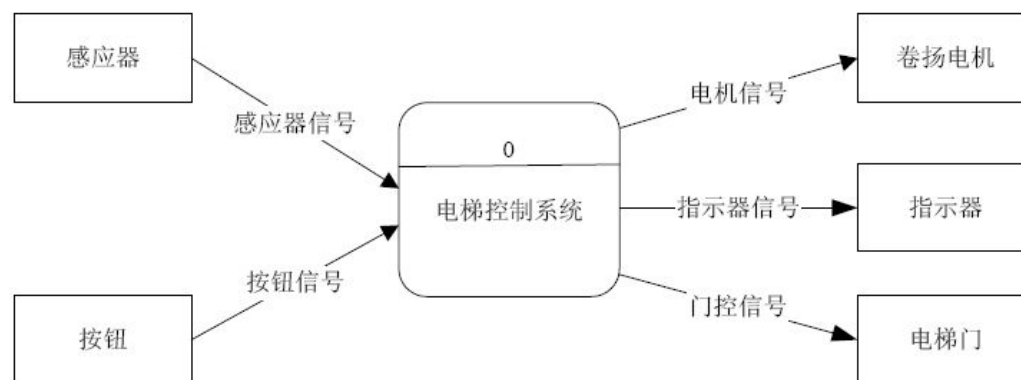


图 12-25、电梯控制系统的上下文图

## 2.4 层次结构的建立

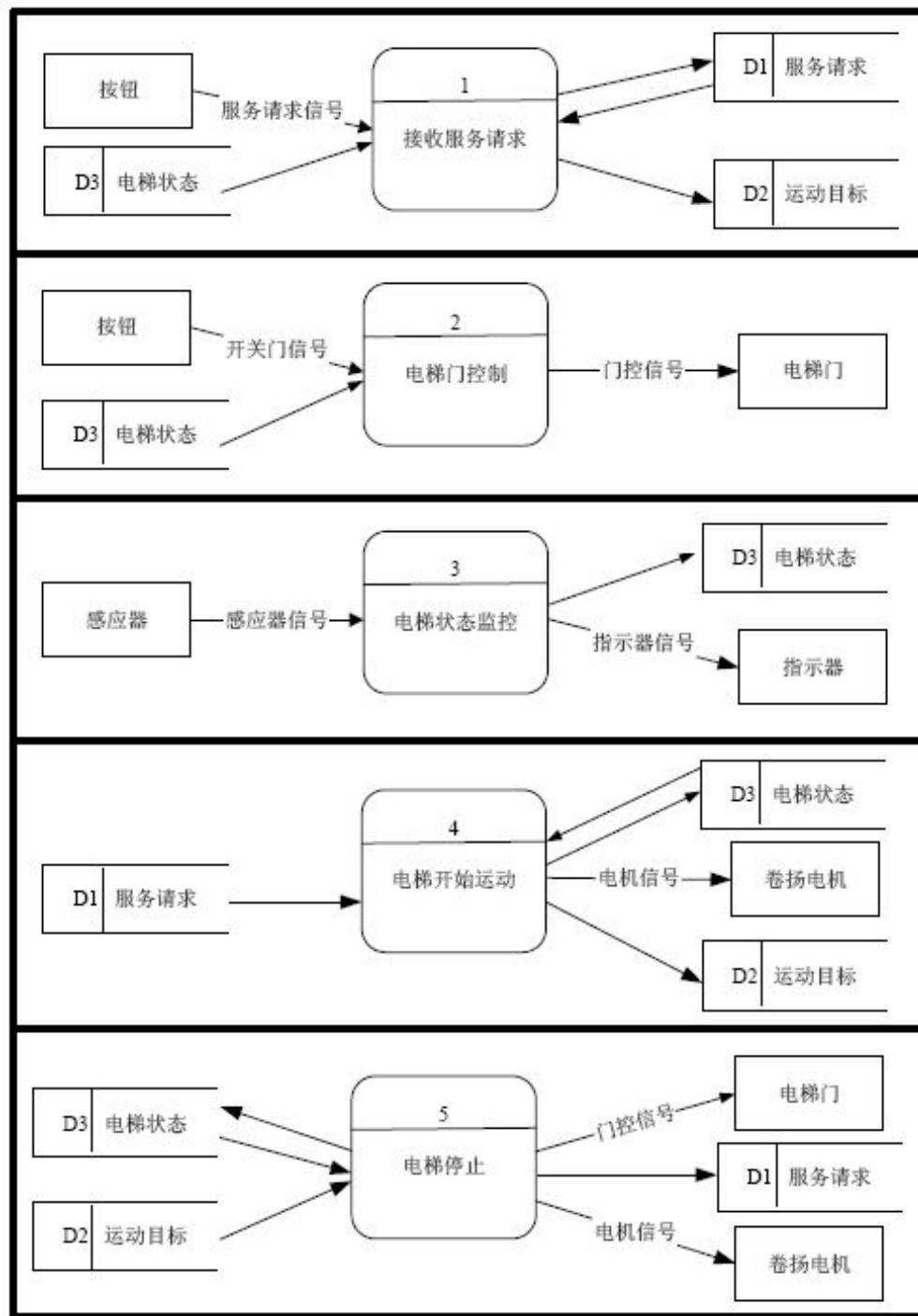
### —— 示例：2.建立DFD片段

事件	系统的响应
用户利用按钮发出服务请求	系统首先要记录请求，以备调度。如果请求时电梯处于运动状态，则系统需要重新执行请求调度，并在需要的情况下更改运动目标。
用户利用按钮发出开关门请求	系统察看电梯状态，如果处于静止状态且处于目前楼层，则发出门控命令，否则不予处理。
感应器信号发生变化	系统要根据新的信号更新电梯状态，并通知指示器改变显示。
电梯开始运动，即门已关闭，开始运动	系统要改变电梯状态为运动状态，然后根据等待的服务请求调度确定电梯的运动目标，结合电梯目前位置，控制卷扬电机开始工作。
电梯停止，即电梯已经到达目标位置	系统更新电梯状态为静止状态以停止对新增请求的处理，去除已完成的请求，然后控制卷扬电机停止运动，并在停止后，开启电梯门

表 12 - 8、电梯控制系统的外部事件及其响应

## 2.4 层次结构的建立

### —— 示例：2.建立DFD片段



## 2.4 层次结构的建立

### —— 示例：3.建立0层图

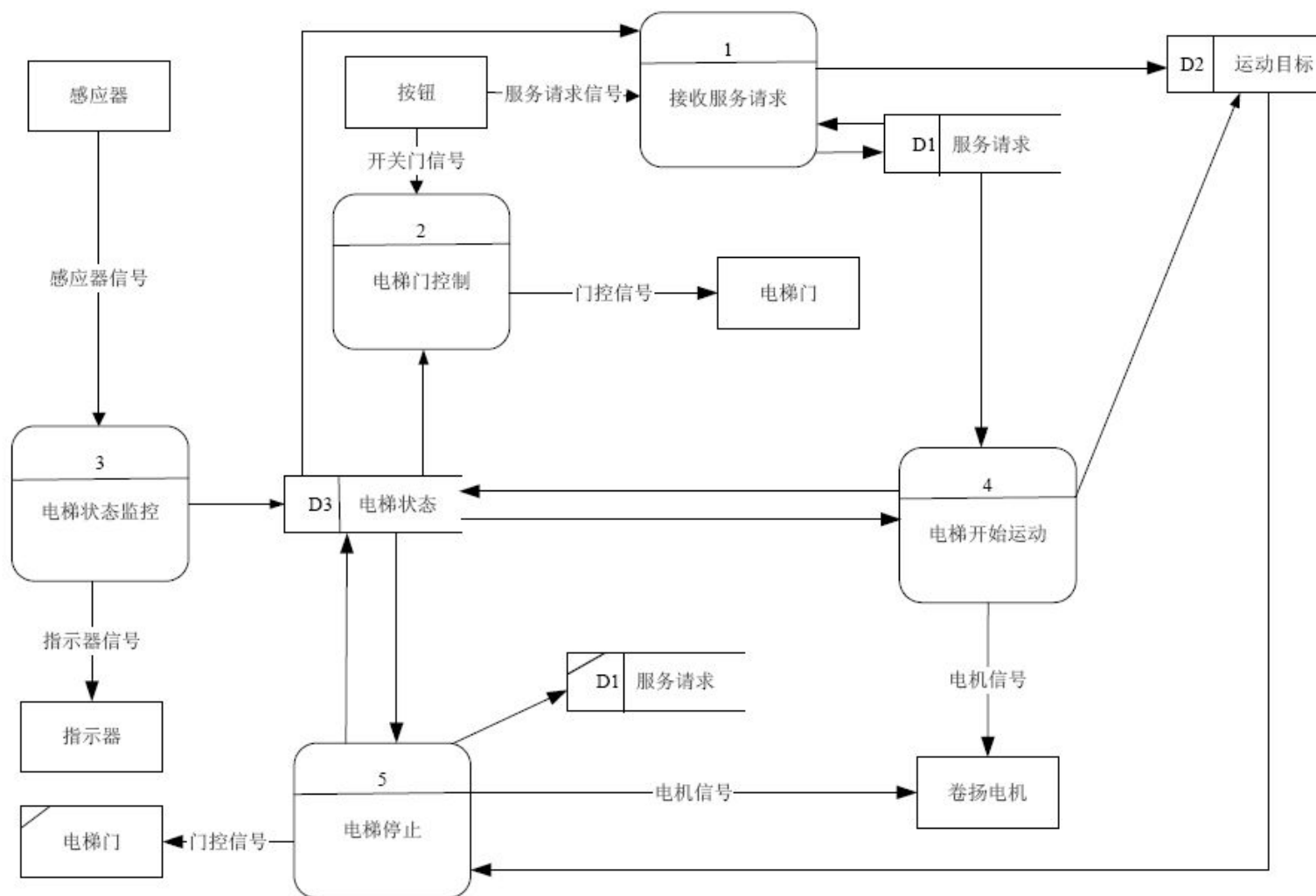


图 12-27、电梯控制系统的初始 0 层图

## 2.4 层次结构的建立

### —— 示例：3.建立0层图

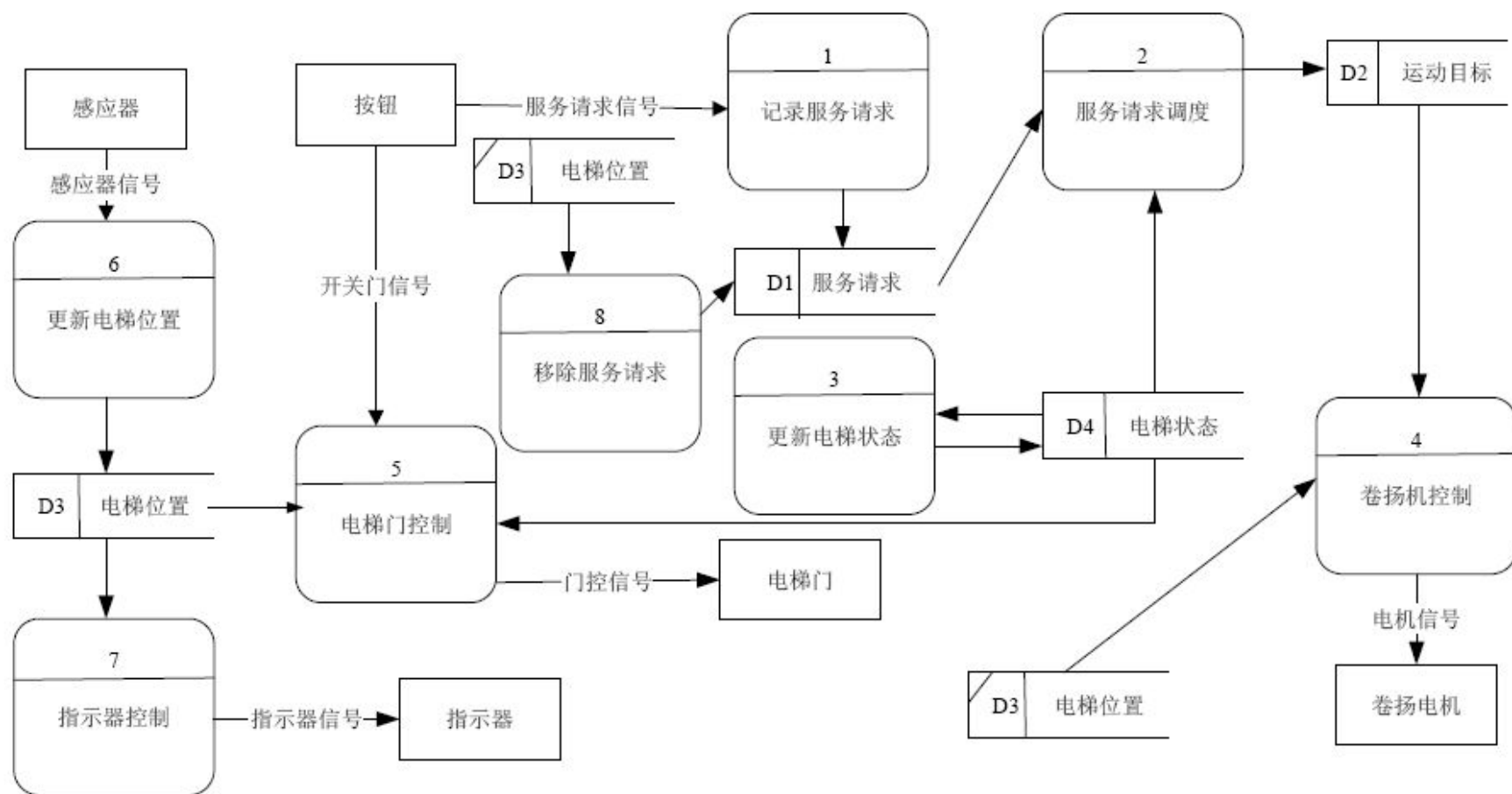


图 12-28、电梯控制系统的最终 0 层图

---

## 2.4 层次结构的建立

—— 示例：4.功能分解建立N层图

- 无



## 2.5 DFD的验证

- 验证DFD的语法
  - 确保DFD中不会发生语法错误
- 验证DFD的结构
  - 验证DFD层次结构之间的一致性
  - 验证DFD层次结构说明的完备性
- 验证DFD的语义
  - 确保DFD所说明内容的正确性和准确性

---

# 主要内容

1. 过程建模
  2. 数据流图DFD
  3. 微规格说明
    1. 结构化英语/伪码
    2. 行为图
    3. 决策表
    4. 决策树
  4. 数据字典
  5. 模块结构图
-

## 3.1 结构化英语/伪码

- 叙述上采用了结构化程序语言的三种控制结构：顺序、条件决策和循环；
- 使用了一些类似于结构化程序语言关键字的词语来表明叙述的逻辑，例如IF、THEN、ELSE、DO、DO WHILE、DO UNTIL等等；
- 在格式上，使用 and 结构化程序语言相同的缩进方式来表明叙述的结构。
- 简短语句，使用名词和动词，避免使用容易产生歧义的形容词和副词



## 3.1 结构化英语/伪码

### ■ 示例

READ customer account-type and summary of customer monthly balances

SELECT CASE

    CASE 1 (account-type is NOW)

        BEGIN IF

            IF daily-balance < 300 for any given day

                THEN set service-charge to \$5

                ELSE set service-charge to \$0

        END IF

    CASE 2 (account-type is REGULAR)

        BEGIN IF

            IF daily-balance < 100 for any given day

                THEN set service-charge to greater of \$3 or \$.20 times  
                number-checks

                ELSE set service-charge to \$0

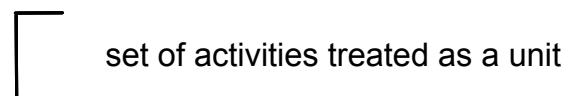
        END IF

END CASE

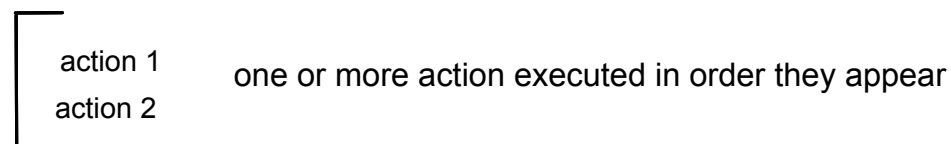
## 3.2 行为图

- 结构化英语的一种特殊表达方式，用特定的图示来表示过程的逻辑结构

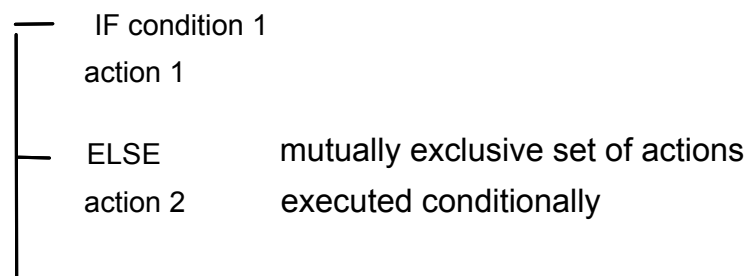
复合语句



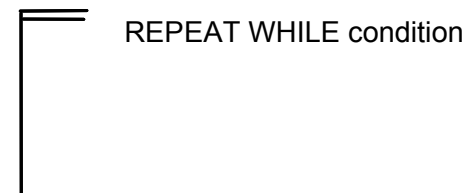
顺序



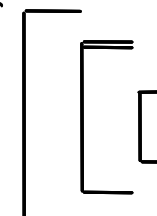
条件选择



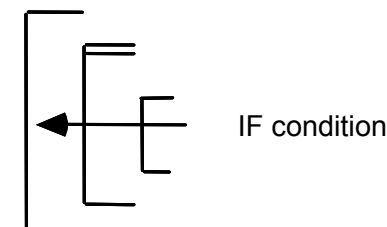
循环



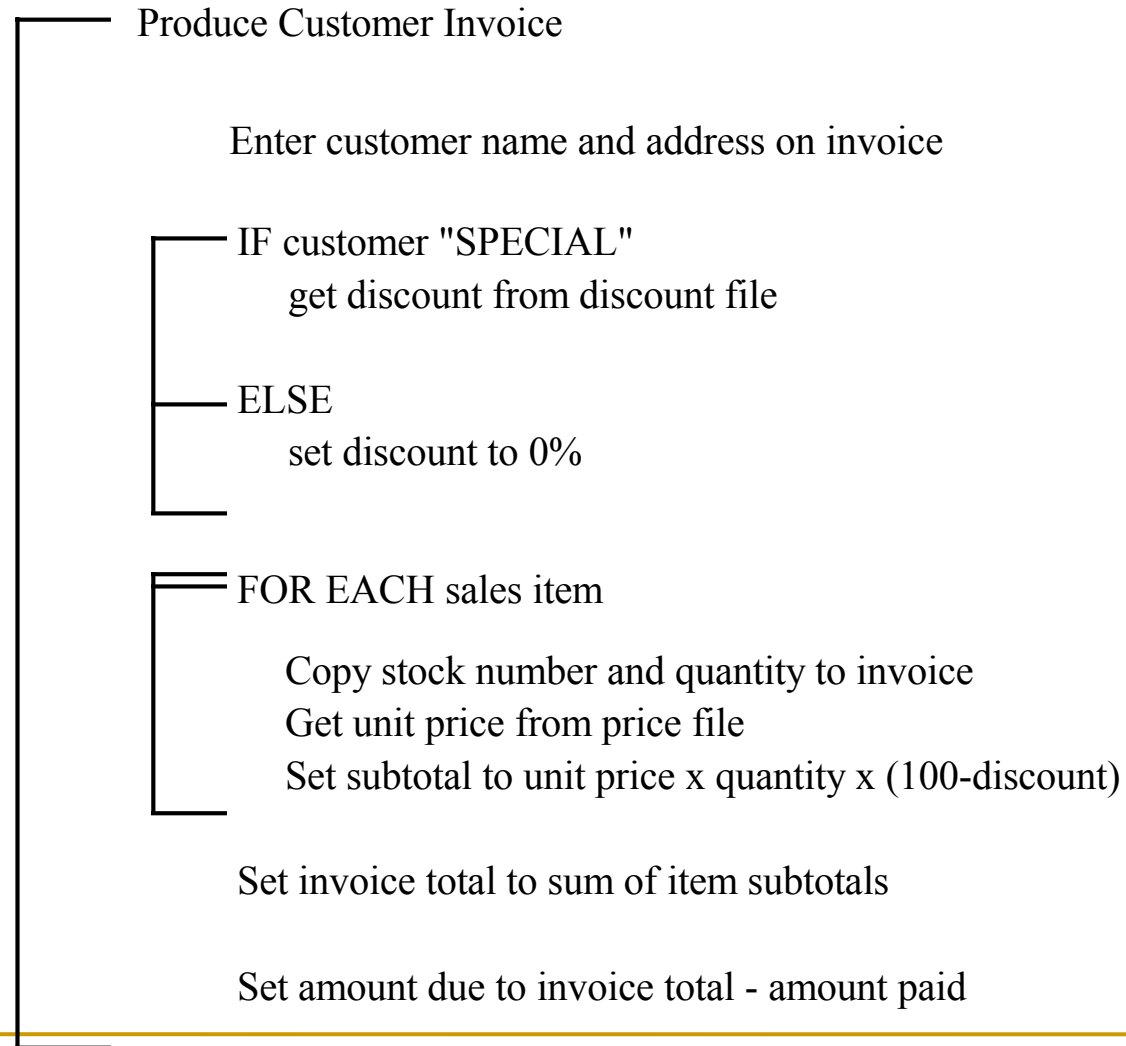
嵌套



从嵌套中退出



## 3.2 行为图示例



## 3.3 决策表

条件和行动	规则
条件声明（Condition Statement）	条件选项（Condition Entry）
行动声明（Action Statement）	行动选项（Action Entry）

- 条件声明是进行决策时需要参考的变量列表
- 条件选项是那些变量可能的取值
- 动作声明是决策后可能采取的动作
- 动作选项表明那些动作会在怎样的条件下发生

## 3.3 决策表

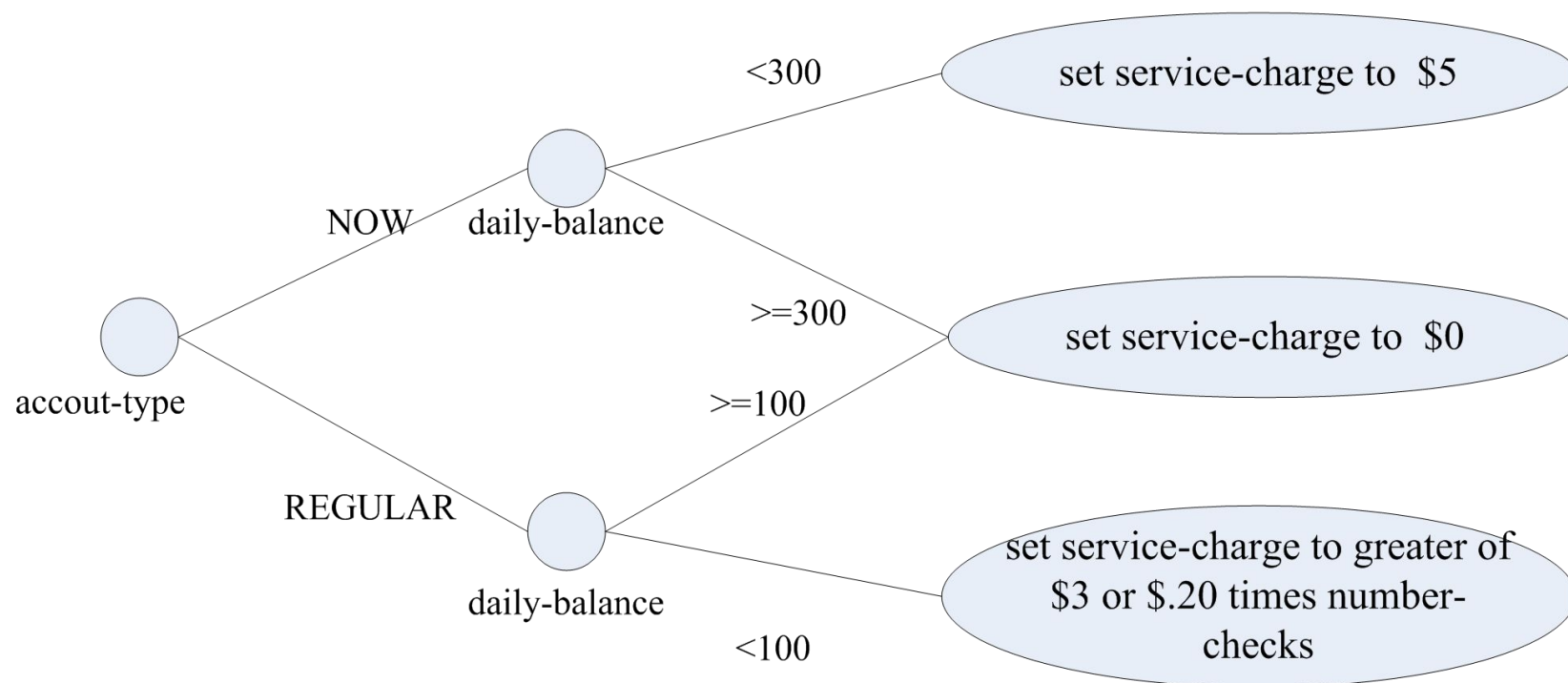
### ——示例

条件和行动	规则			
account-type	NOW	NOW	REGULAR	REGULAR
daily-balance	< 300	>=300	< 100	>=100
set service-charge to \$5	X			
set service-charge to \$0		X		X
set service-charge to greater of \$3 or \$.20 times			X	



## 3.4 决策树

- 通常是一颗平放的树，树根在左边，树枝从左向右展开。树枝上是有关条件和行动的描述



---

# 主要内容

1. 过程建模
  2. 数据流图DFD
  3. 微规格说明
  4. 数据字典
  5. 模块结构图
-

## 4. 数据字典

- 数据字典是一个储存库，包含软件使用和产生的所有数据对象的描述，其中也包括**DFD**当中数据流和数据存储的定义
- 有组织地列出**DFD**中的涉及的所有数据元素（数据流、数据存储），并定义每个数据元素的
  - 名称
  - 表示方法
  - 单位/格式
  - 范围
  - 使用地点
  - 使用方法
  - 其他描述信息

## 4. 数据字典

### ——数据结构描述

- 数据字典要求对数据元素（尤其是其结构）的描述要精确、严格和明确

符号	含义	示例
=	包含，由...构成	Name=first_name+last_name
+	指明序列结构	
()	内容可选	Phone_No.=(Area_No.)+Local_No.
[]	内容多选一	Number=[0 1 2 3 4 5 6 7 8 9]
	分割[]内部的多个选项	
n{}m	循环,最少n次,最多m次	Area_No=3{Number}4
@	数据存储的标识符(关键字)	Student=@ID+Name+...
**	注释	Area_No=3{Number}4**区号为3到4位数字

## 4. 数据字典

### ——数据结构描述示例

定义	说明
<p>telephone no. = [ local extension   outside no.   0 ]</p> <p>local extension = 3 {0-9} 3</p> <p>outside no. = 9 + [ service code   domestic no. ]</p> <p>service code = [ 110   120   ... ]</p> <p>domestic no. = (area code) + local number</p> <p>area code = 3 {0-9} 4</p> <p>local number = 8 {0-9} 8</p>	<p>电话号码可能是内线、外线或者转接主机（拨0）</p> <p>内线号码是3位数字</p> <p>外线要先拨9，然后再拨特服号码或普通电话号码</p> <p>特服号码有110、120、...</p> <p>普通电话号码为可选的区号加本地号</p> <p>区号是3到4位数字</p> <p>本地号是8位数字</p>

## 4. 数据字典

- 数据字典为每个数据元素组织描述信息

名称	数据元素的原始名称
别名	数据元素的其他名称
使用地点	会使用该数据元素的过程
使用方法	该数据元素扮演的角色（输入流、输出流或者数据存储等）
使用范围	该数据元素存在的范围
描述	对数据元素内容的描述
单位/格式	数据元素的数据类型，可能事先设置的取值

## 4. 数据字典

### ——示例

名称	telephone number
别名	phone number, number
使用的地点和方法	read-phone-number (input) display-phone-number (output) analyze-long-distance-calls (input)
描述	telephone no. = [ local extension   outside no.   0 ] local extension = 3 {0-9} 3 outside no. = 9 + [ service code   domestic no. ] service code = [ 110   120   ... ] domestic no. = (area code ) + local number area code = 3 {0-9} 4 local number = 8 {0-9} 8
格式	alphanumeric data

---

# 主要内容

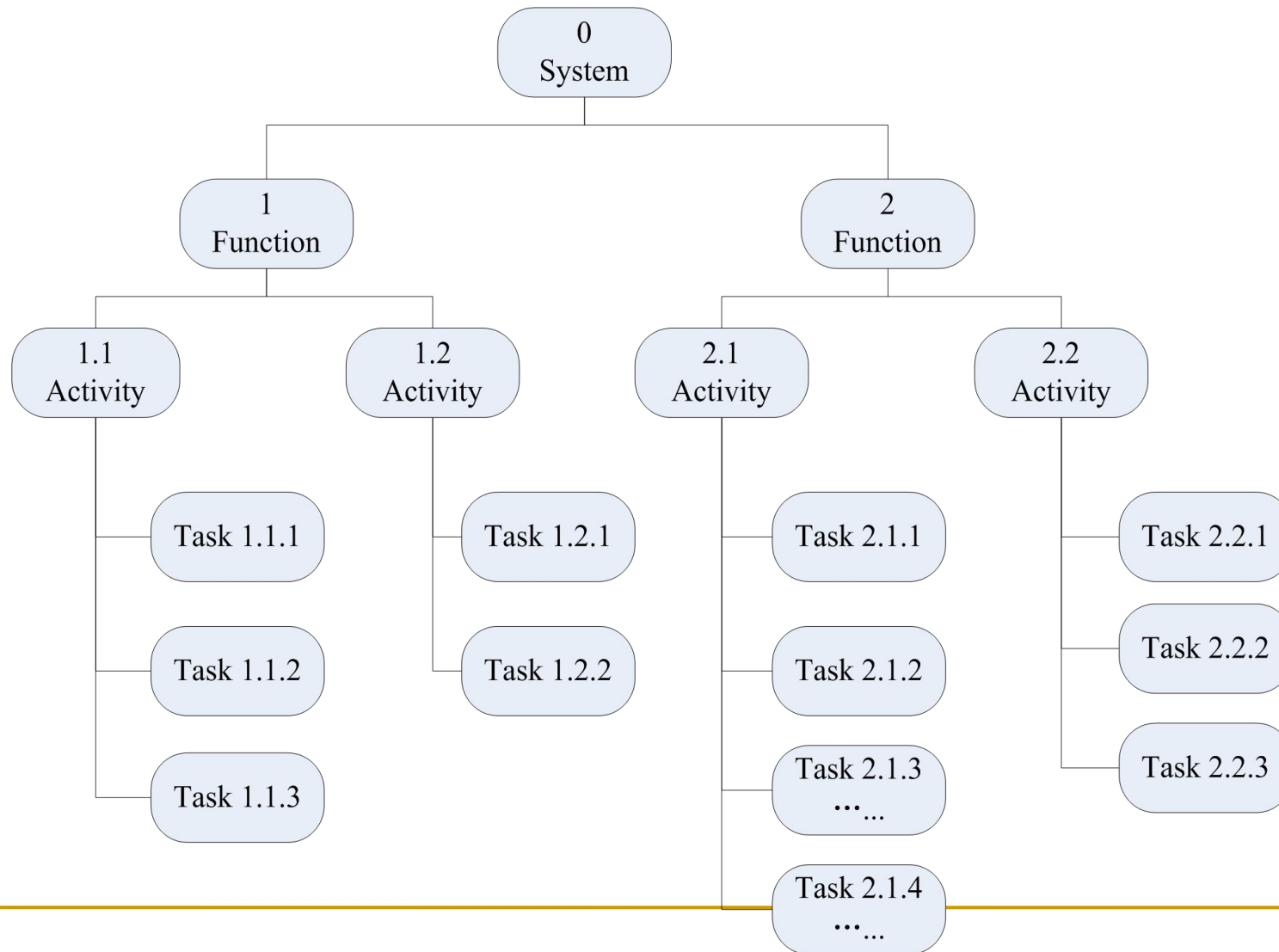
1. 过程建模
  2. 数据流图DFD
  3. 微规格说明
  4. 数据字典
  5. 模块结构图
    1. 功能分解图
    2. 过程依赖图
  6. DFD的类型
-



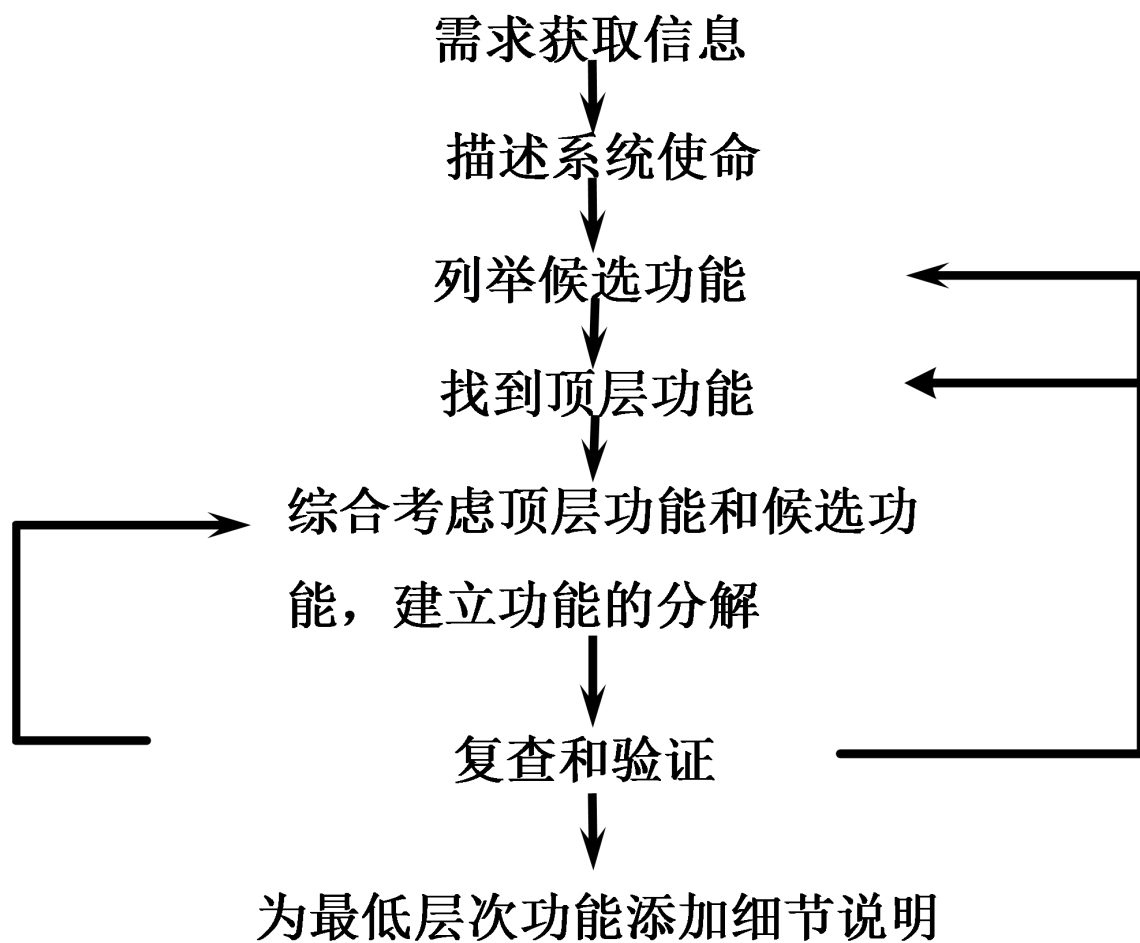
## 5.1 功能分解图

- 在一个图内自上至下的集中显示系统的功能分解结构
  - 最顶层的单独功能通常是对整个系统的使命描述，是对系统业务需求的概括
  - 系统使命说明的下一层被称为功能的最顶层，描述了系统应该具备的一些重要功能，它们支撑着系统使命的实现
  - 功能最顶层下面的分支是对最顶层功能执行分解后形成的层次关系
  - 最底层的是基本的业务功能。这些基本的业务功能是人们所能找到的最基本的、不可再细分的功能或处理
- 能够更加集中、更加直观的展示大量过程之间的层次关系

## 5.1 功能分解图



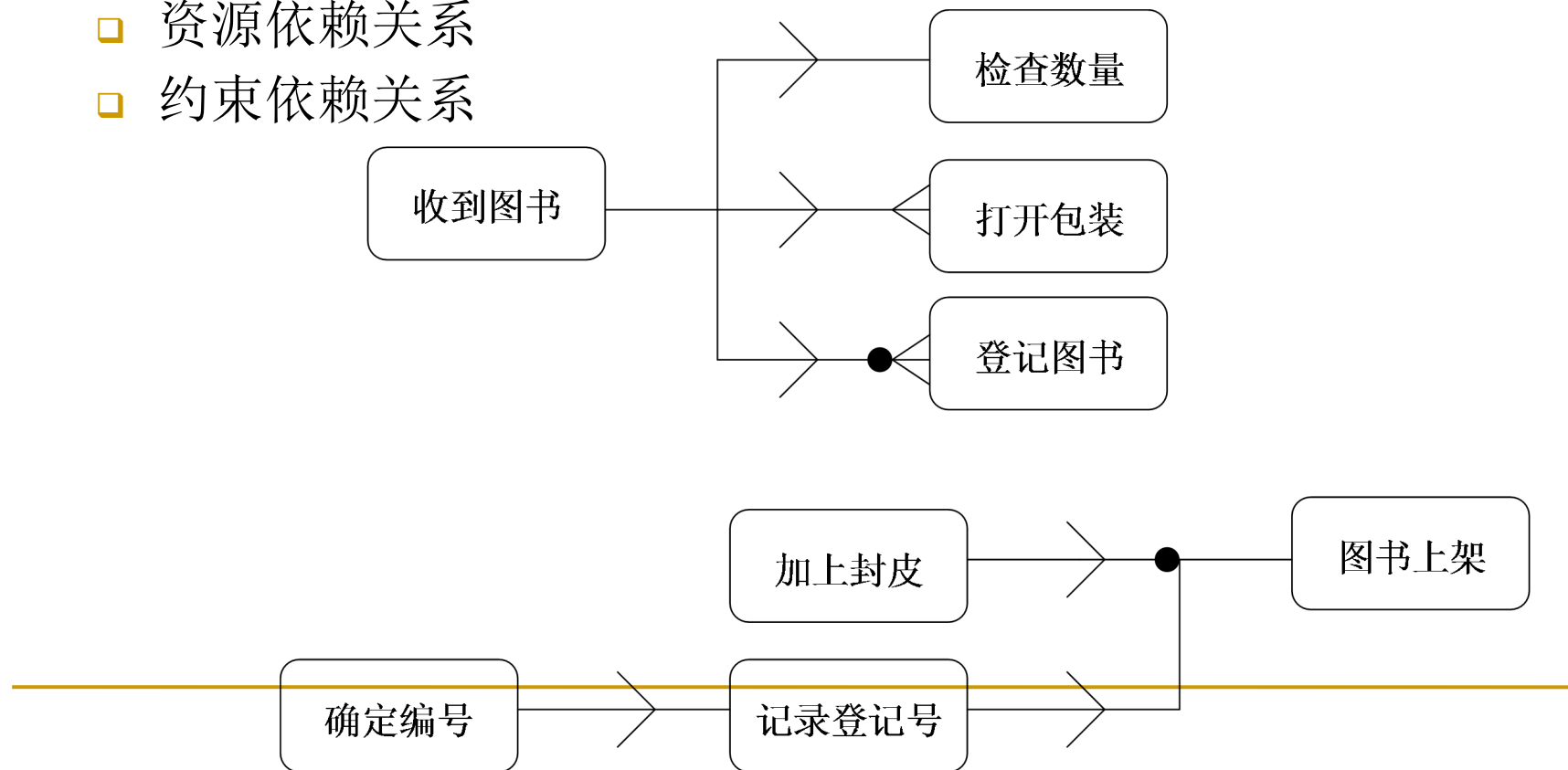
## 5.1 功能分解图



## 5.2 过程依赖图

### ■ 描述功能和过程之间的依赖关系

- 数据依赖关系
- 资源依赖关系
- 约束依赖关系



---

# 主要内容

1. 过程建模
  2. 数据流图DFD
  3. 微规格说明
  4. 数据字典
  5. 模块结构图
-

## 本章小结

- 结构化分析的主要方法是过程建模和数据建模，其中过程建模是其核心
- 过程建模以**DFD**为中心，结合使用微规格说明、数据字典、**ERD**、**FDD**、**PDD**等技术一起完成结构化分析的建模任务
- **DFD**建立在四种简单的元素之上，使用层次结构表达了功能分解关系，它是一种简洁、易于理解和使用技术，但是它较用于建模复杂系统