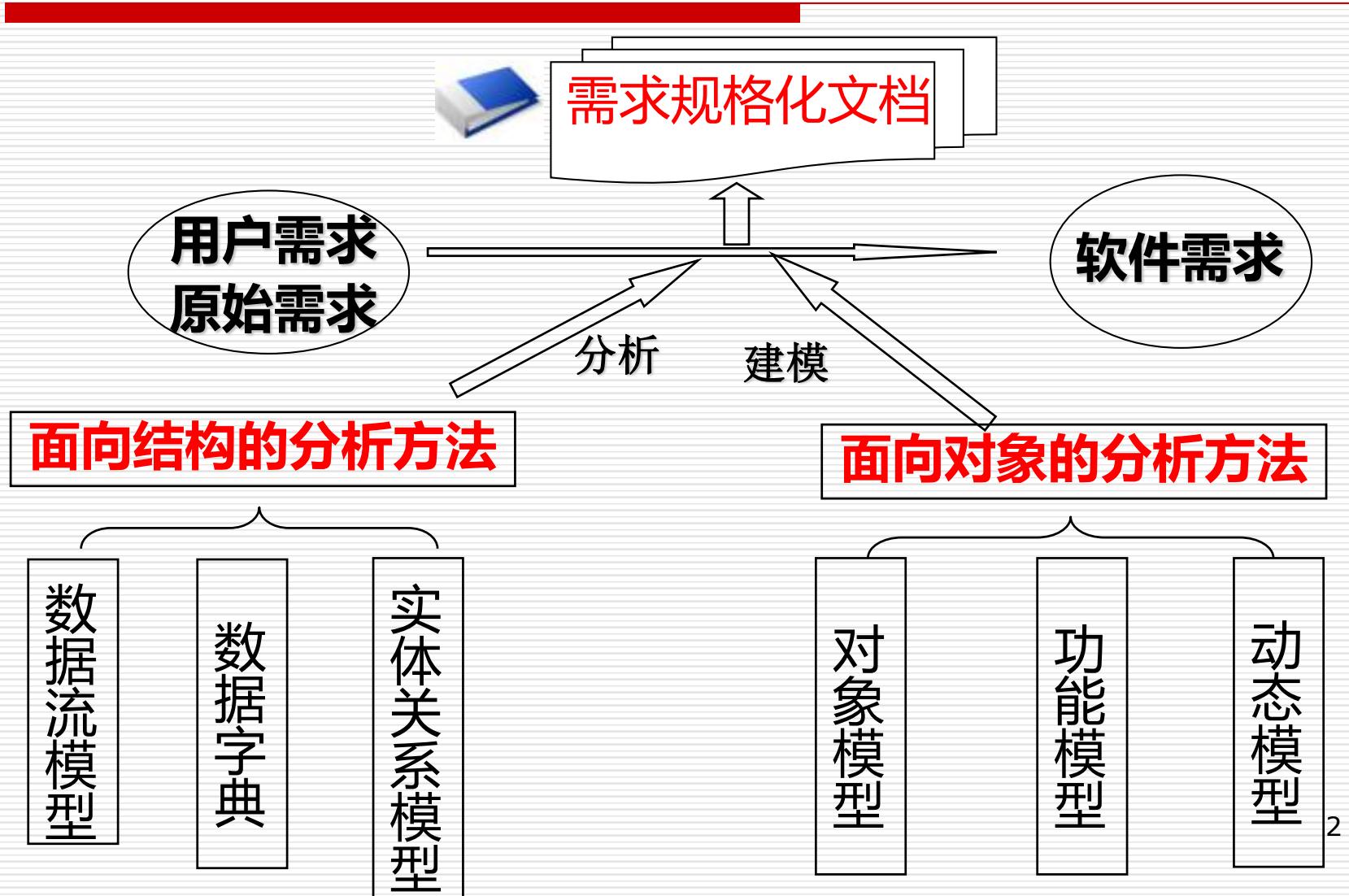


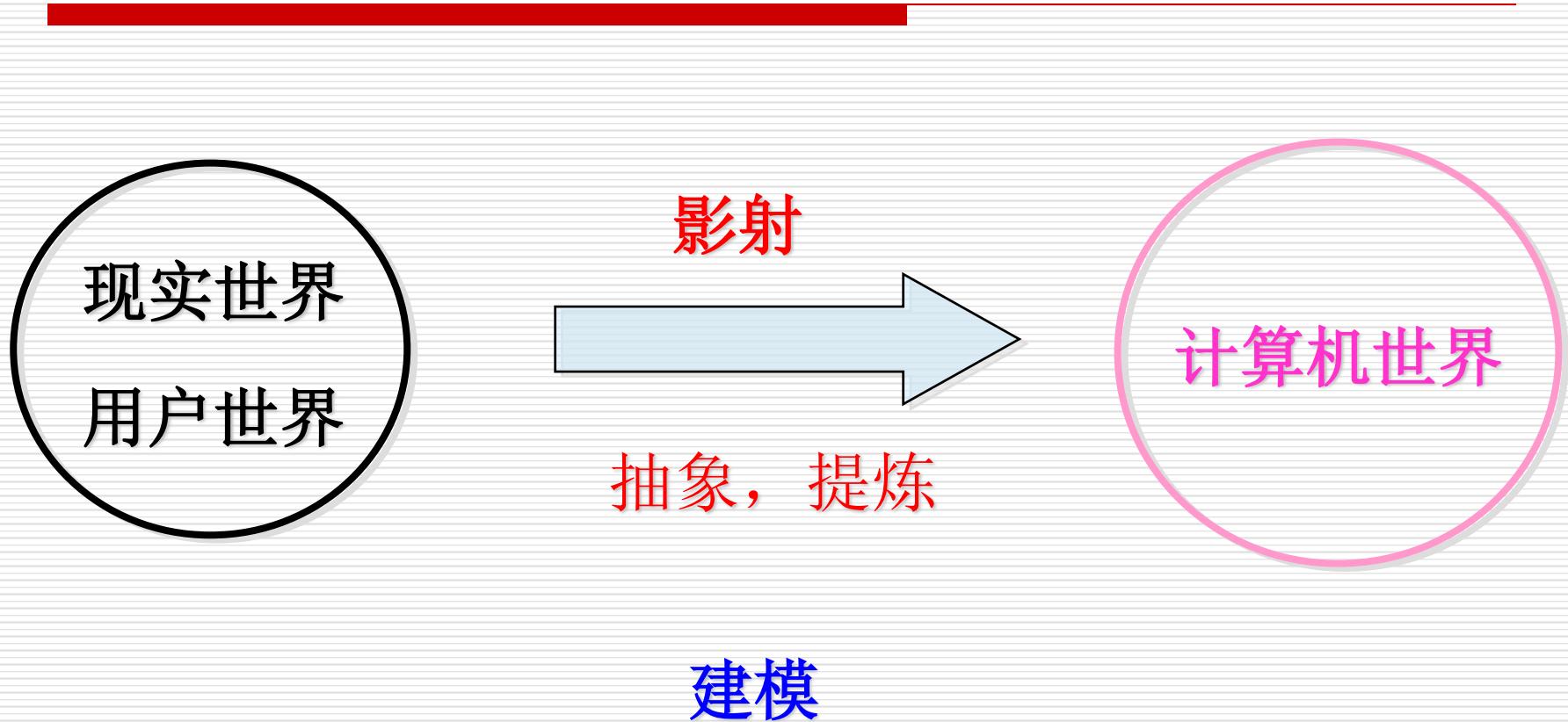
# **CHAPTER 3**

## **Software Requirement Analysis**

# Outline



## § 3.6 软件需求分析建模方法



# 模型的作用

---

- ✓ 在建模过程中了解系统
- ✓ 通过抽象降低复杂性
- ✓ 有助于回忆所有的细节
- ✓ 有助于开发小组间的交流
- ✓ 有助于与用户的交流（画图方法）
- ✓ 引导目标软件系统设计
- ✓ 为系统的维护提供文档

# 软件需求分析建模方法

---

- 结构化分析(传统建模方法)
  
- 面向对象分析

**掌握并能正确运用分析和开发方法，具有事半功倍的作用**

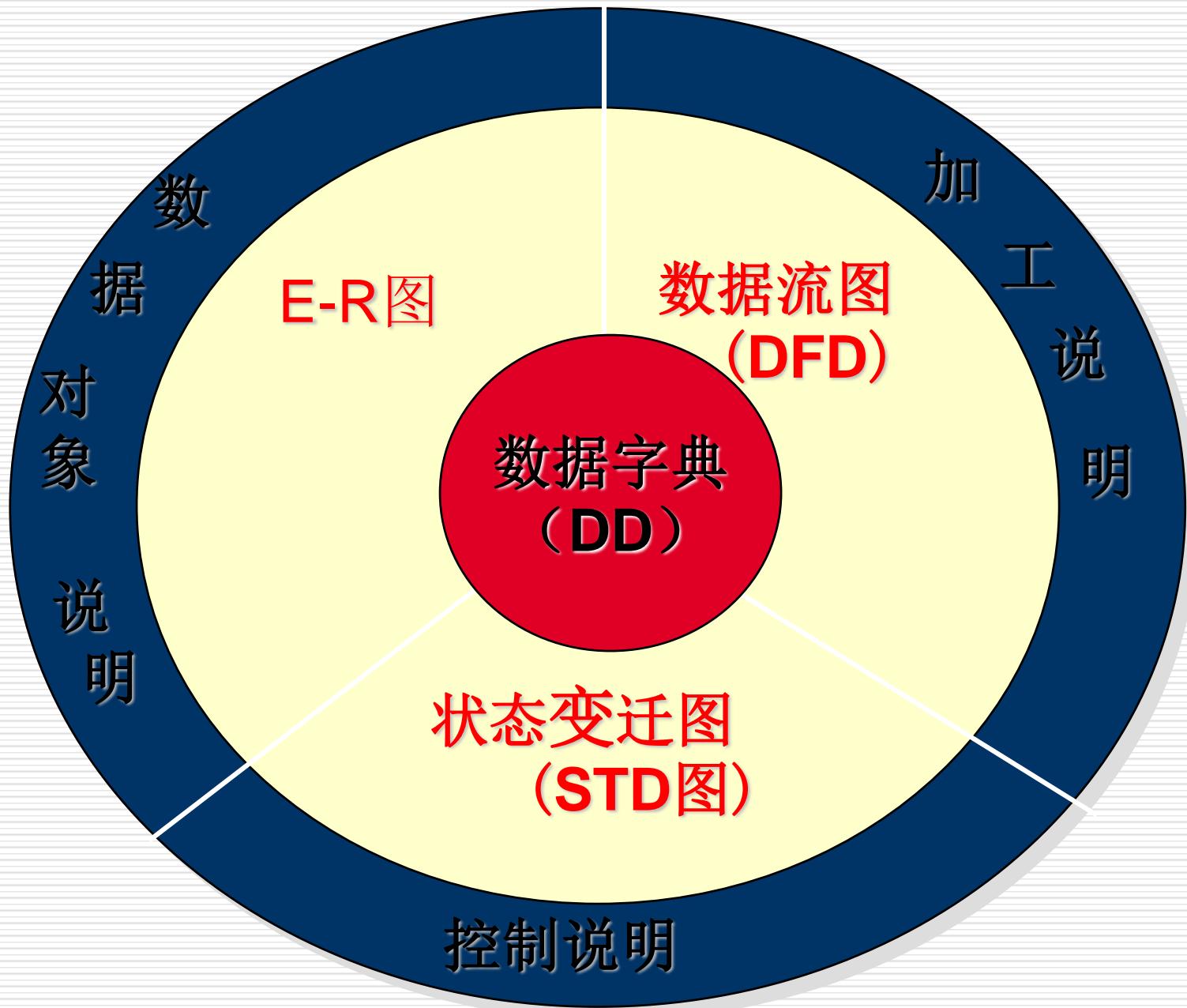
## § 3.6.1 Structured Analysis 主要内容

---

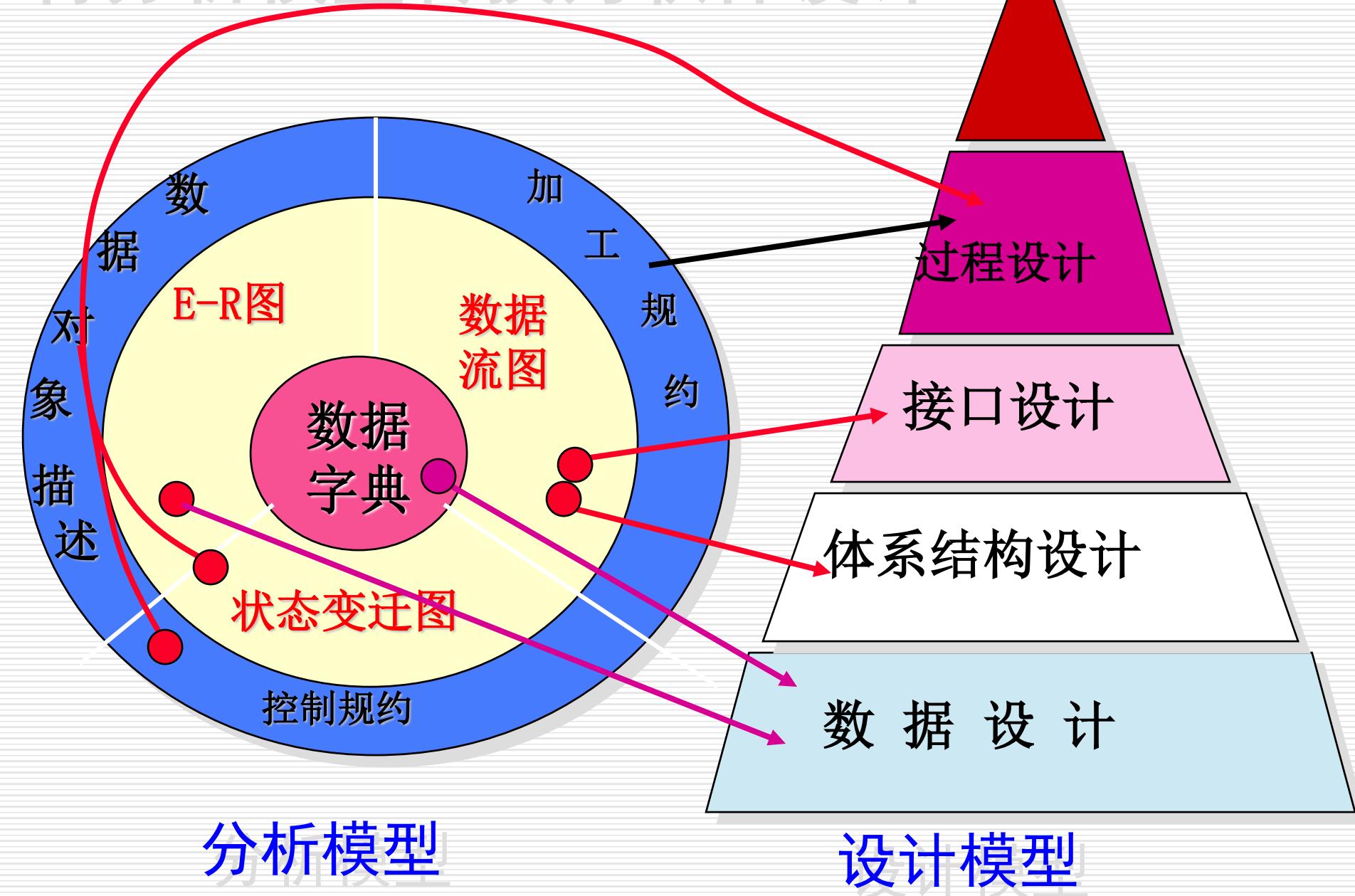
- Data Flow Diagrams
- Data Dictionary
- Entity Relationship Diagrams
- State-Transition Diagrams

- (1) 提供一组术语(符号), 指导需求抽象中需要关注的主要方面, 并用于表达分析中有用的信息。
- (2) 给出设计、开发指导, 以支持系统化地使用相关信息建造系统模型。

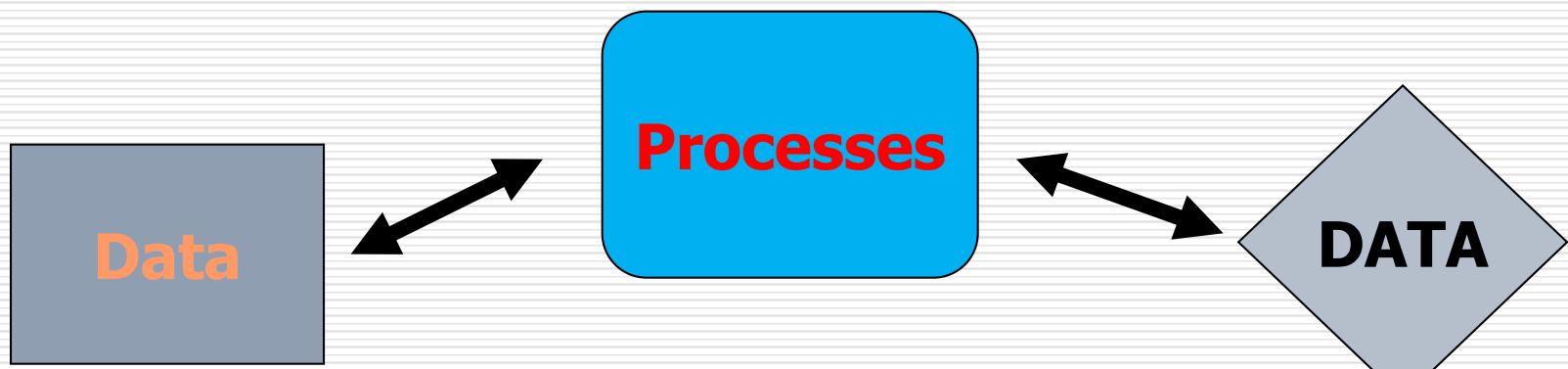
# 结构化分析模型的组成结构



# 将分析模型转换为软件设计



# 结构分析主要思想



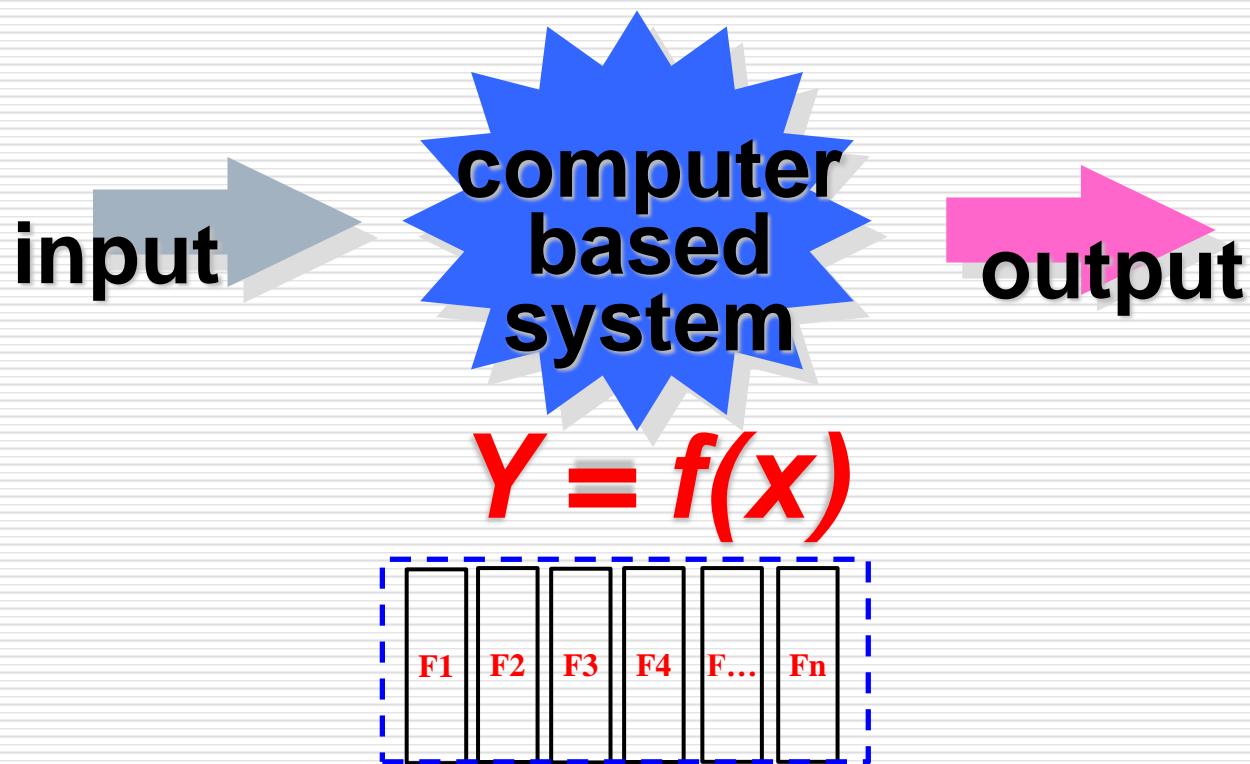
过程, 步骤, 处理, 加工, 操作, 计算, 功能, ...

- ✓ 面向结构的分析方法 SA
- ✓ 面向过程的分析方法
- ✓ 面向功能的分析方法

Maps and transform to structured program (顺序, 分支, 循环)

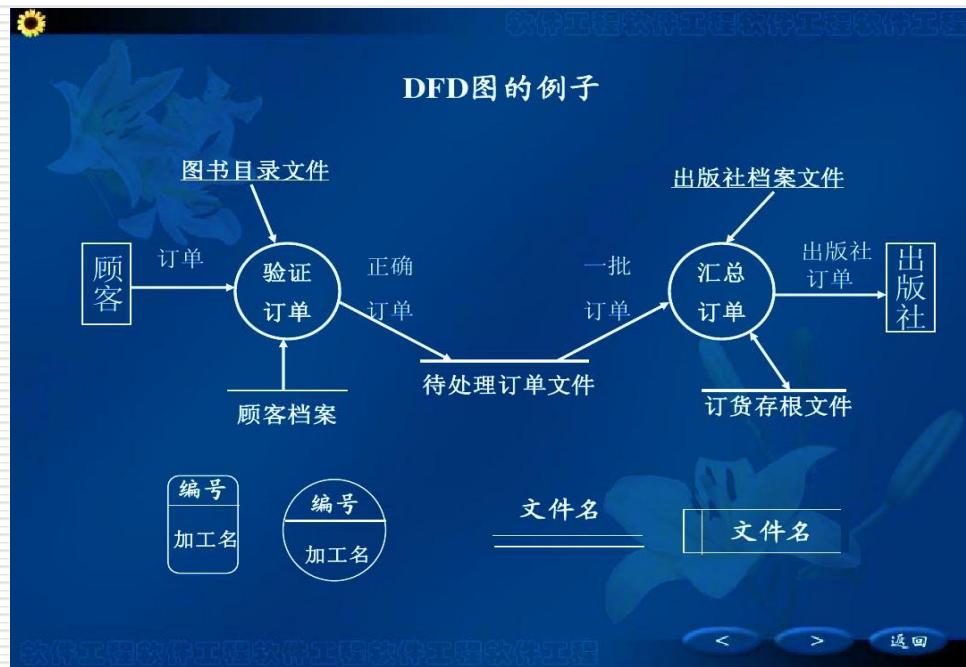
# Computer 模拟现实世界

Every computer-based system is an information transform .....



# (1) Data Flow Diagram 数据流图

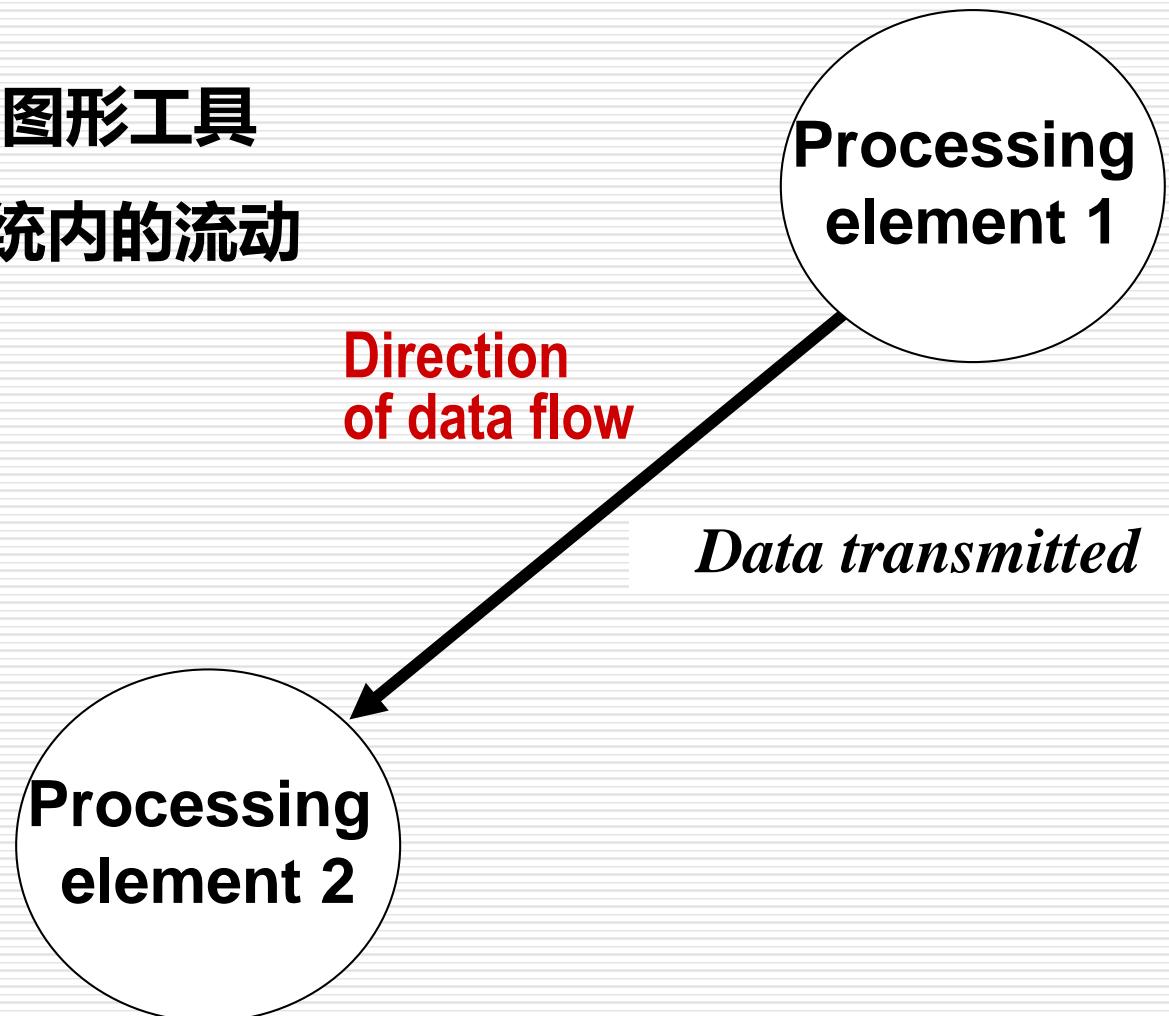
- ✓ 描述系统的逻辑模型，图中没有任何具体的物理元素，只是描绘**信息**在系统中的流动和处理情况。
- ✓ 就图本身而言，并不是只有程序员，或计算机专业技术人员能够读懂，特别是需求方（客户，用户）也能读懂。



画图，体现工程的思想

# (1) Data Flow Diagram 数据流图

描述逻辑模型的图形工具  
，表示数据在系统内的流动  
和变化。



# 数据流图的详解

---

- Data flow diagram is an intuitive way of showing how data is processing by a system.
- The DFD is a pictorial representation of all aspects of the logical data flow.
- DFD is used to show how data flows through a sequence of processing steps.
- It is a logic model which does not depend on system hardware, software, and so on. It has no physical meaning.
- It is a behavioral model.
- It is a functional model.
- **It is not a program flow diagram**

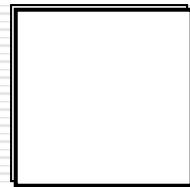
# Data Flow Diagram 技术

---

## ➤ Data flow diagrams (DFD)

- Symbols
- Rules for creating
- Decomposition
- Balancing ( level )

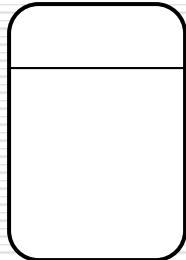
# DFD Symbols (符号, 术语)



**External Entity**



**Flow of Data**

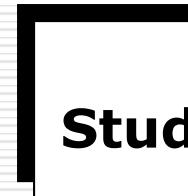


**Processing**



**Data Store**

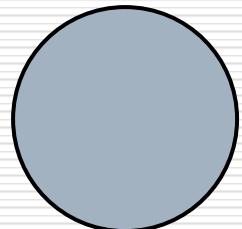
**abc**



**Student**



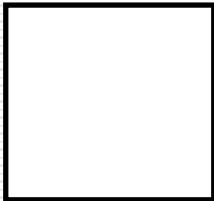
**Student Master**



**Describing letter**

# External Entity 外部实体

---



A producer or consumer of data

Examples: a person, a device, a sensor

Another example: computer-based system

Data must always originate somewhere  
and must always be sent to otherwhere

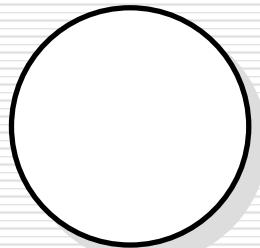
# External Entity 外部实体

---

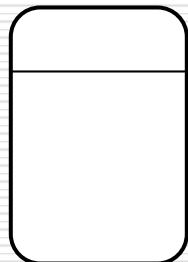
- **Source/Sink** (源/宿, 始/终)
  - Depicts the origin and/or destination of the data
  - Sometimes referred to as an external entity
  - Drawn as a square symbol
  - Name states what the external agent is
  - Because they are external, many characteristics are not of interest to us

# Processing 处理

---



A data transformer (changes input to output)

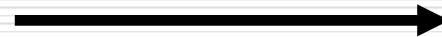


Data must always be processed in some way to achieve system function

至少有一个输入和一个输出

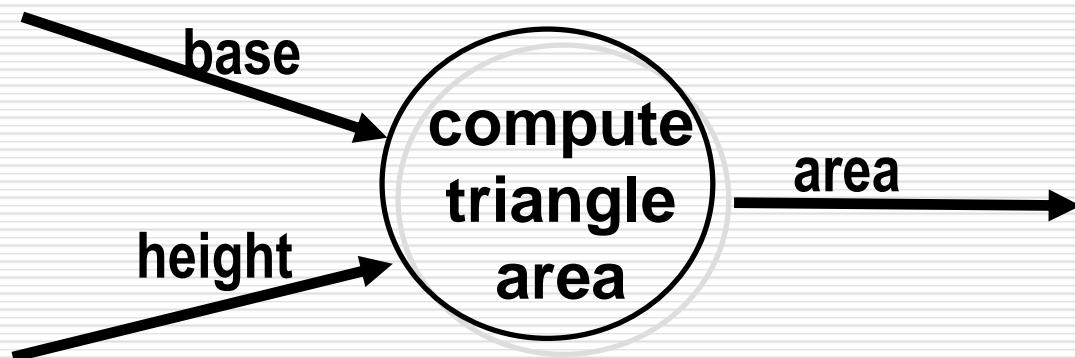
过程, 步骤, 行动, 处理, 加工, 操作, 变换, 计算, ...

# Data Flow



---

Data flows through a system, beginning as input and being transformed into output.

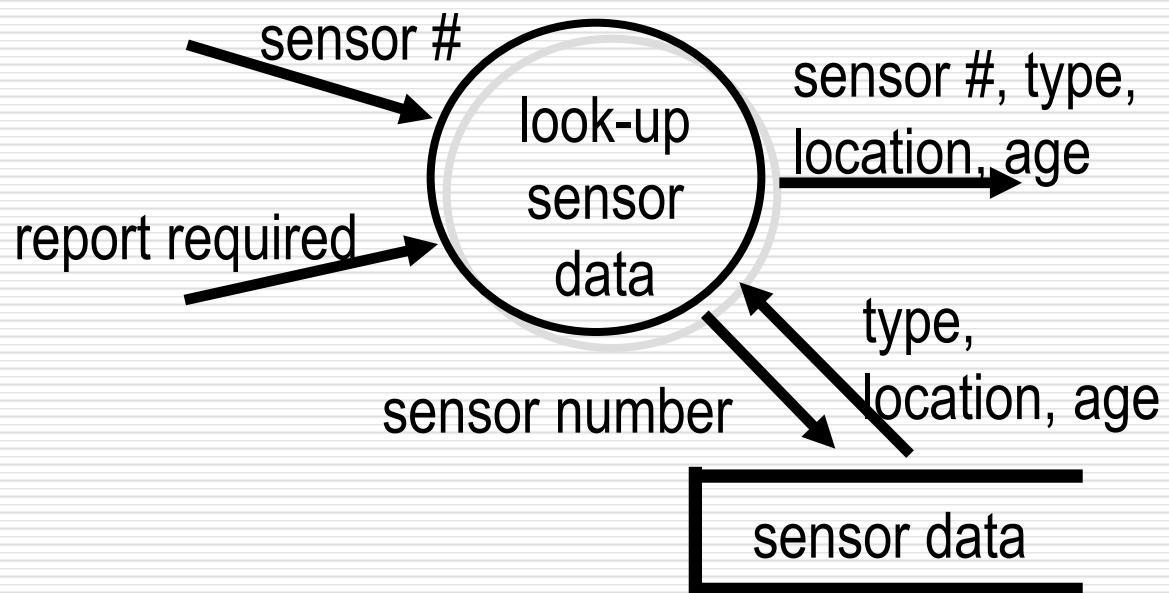


$$\text{Area} = \text{base} \times \text{height} / 2$$

# Data Stores



**Data is often stored for later use**



# 数据存储

---

## ➤ Data Store

- Depicts data at rest
- May represent data in
  - ✓ File folder
  - ✓ Computer-based file
  - ✓ Notebook
- The name of the store as well as the number are recorded in between lines

# **Describing letter**

---

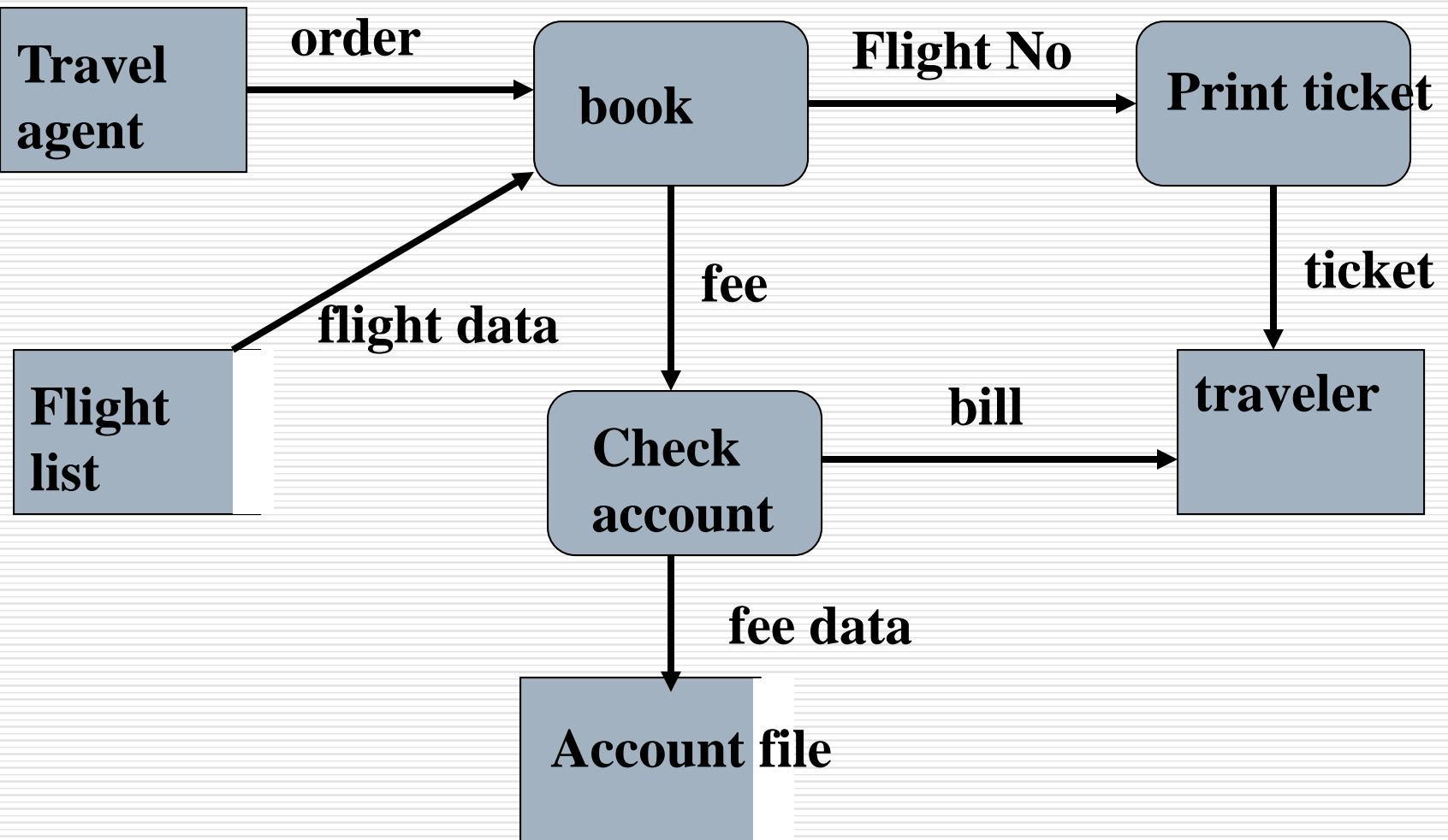
**Noun**

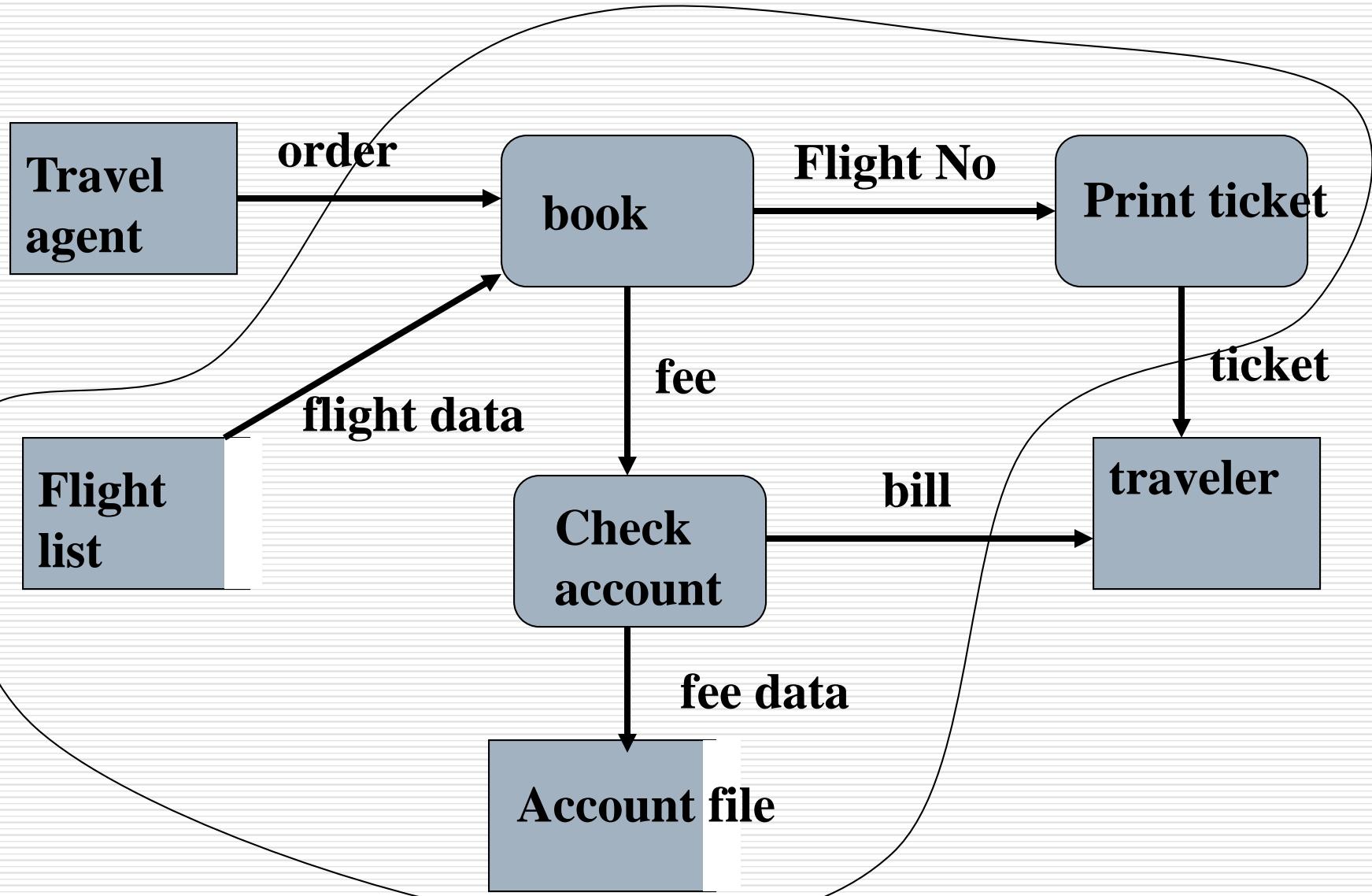
**verb, gerund**

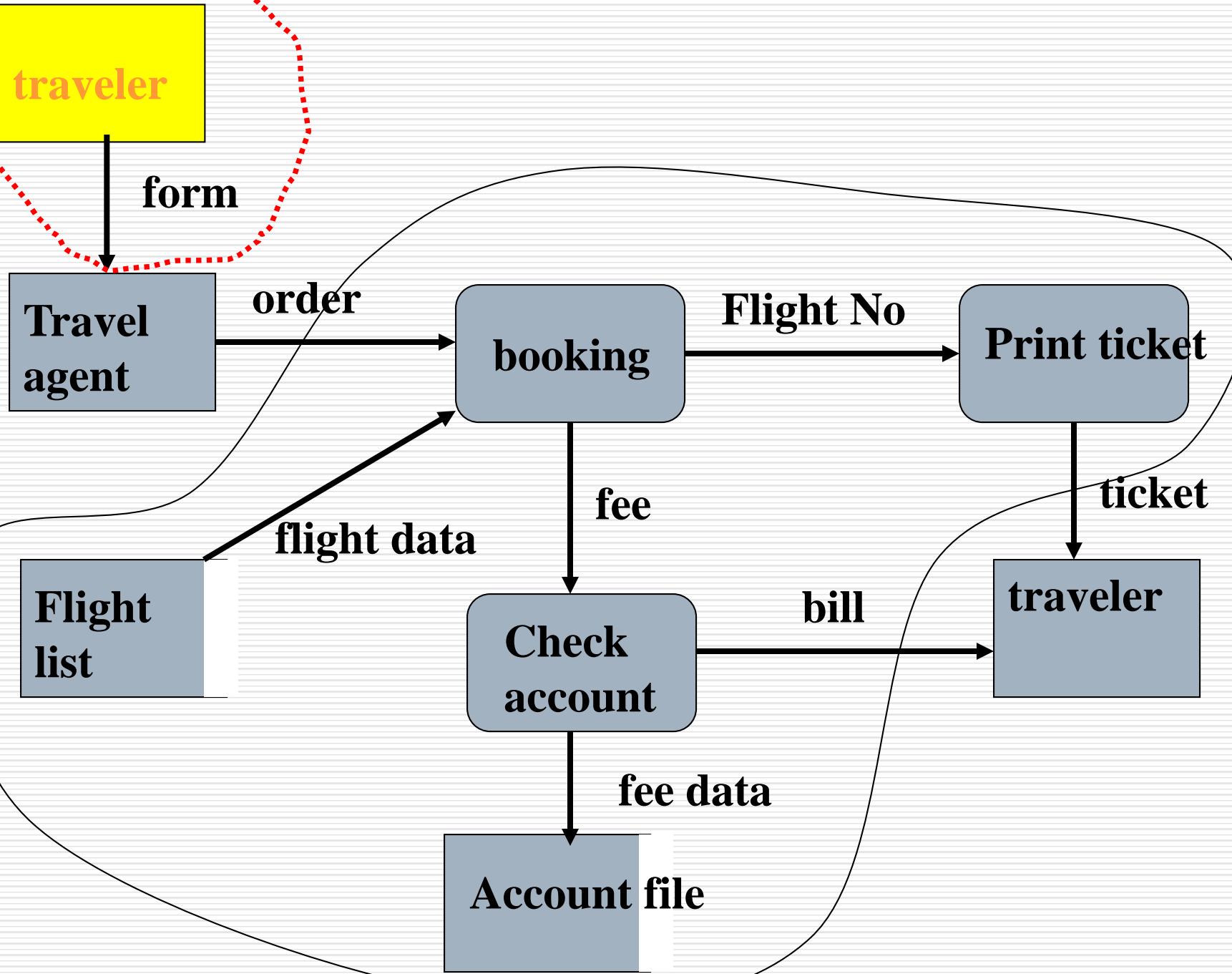
# 举例1： plane ticket booking system

---

- ✓ 旅客到旅行社预定机票
- ✓ 旅行社登记受理
- ✓ 根据目的地和航班情况制定机票
- ✓ 打印机票
- ✓ 计算帐单
- ✓ ...







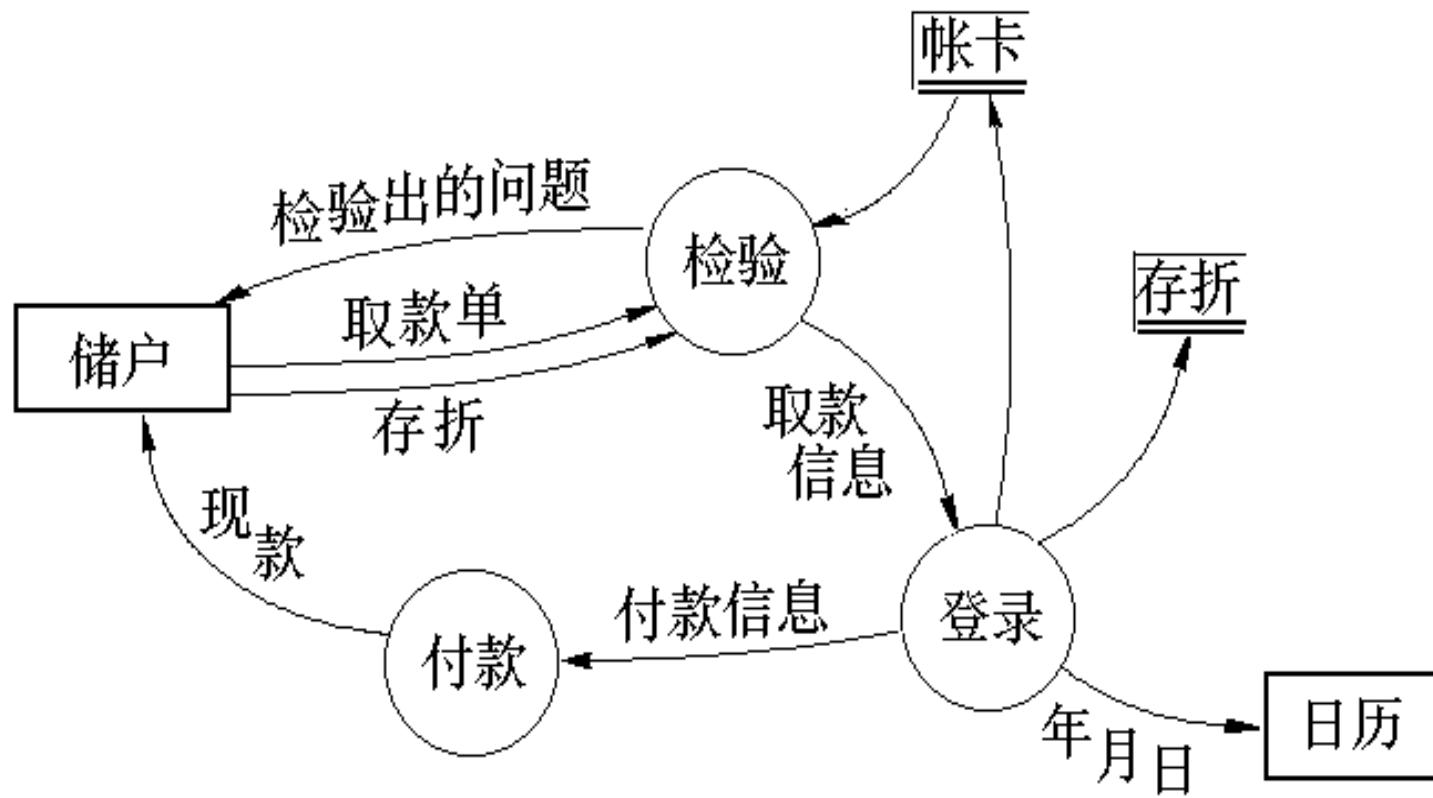
## 例2：去银行取款问题描述

---

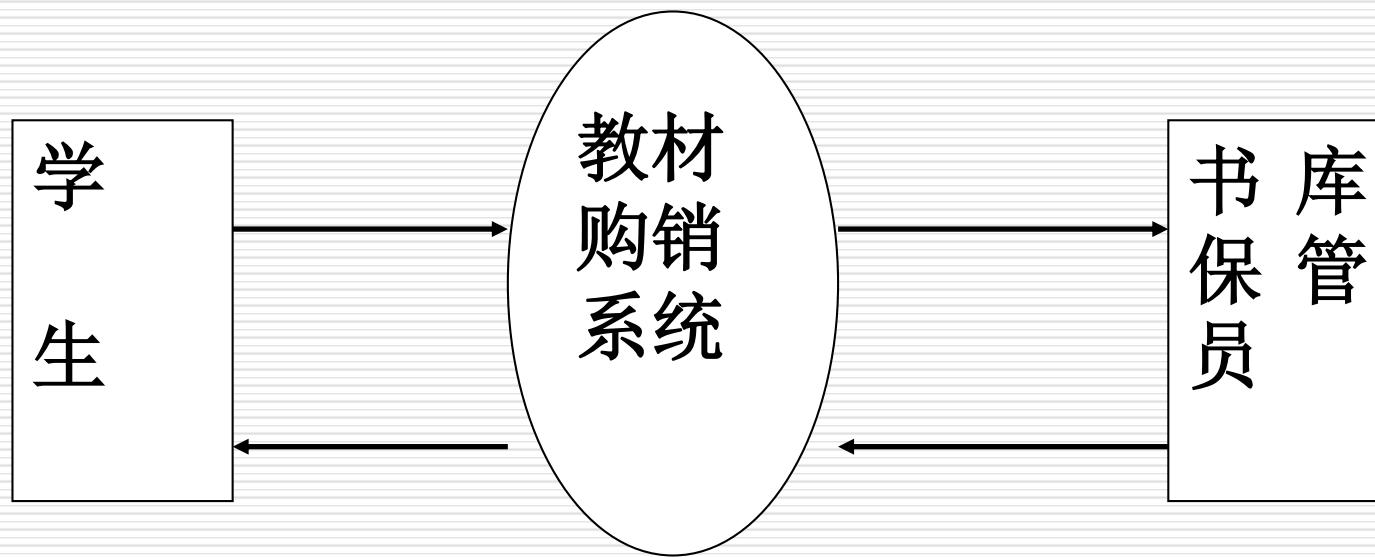
储户把存折和取款单一并交给银行出纳员检验。

出纳员核对账目，一旦发现存折有效性问题、取款单填写问题，或存折、帐卡与取款单不符等问题时，均应报告储户。在检查通过后，出纳员将取款信息登记在存折和帐卡上，并通知付款。根据付款同志给储户付款，从而完成这一简单的数据处理活动。

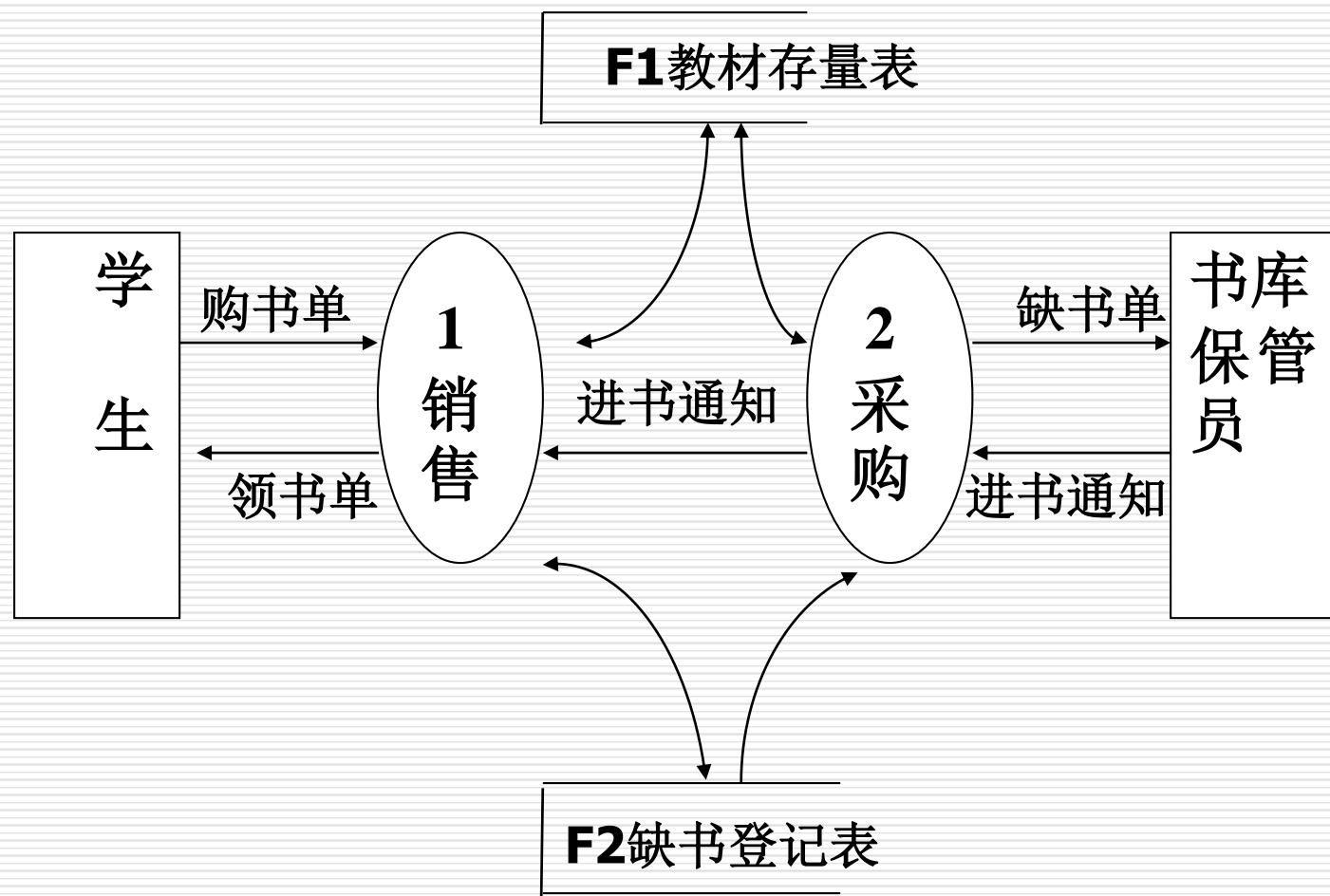
# 去银行取款数据流图



# 例3：售书系统

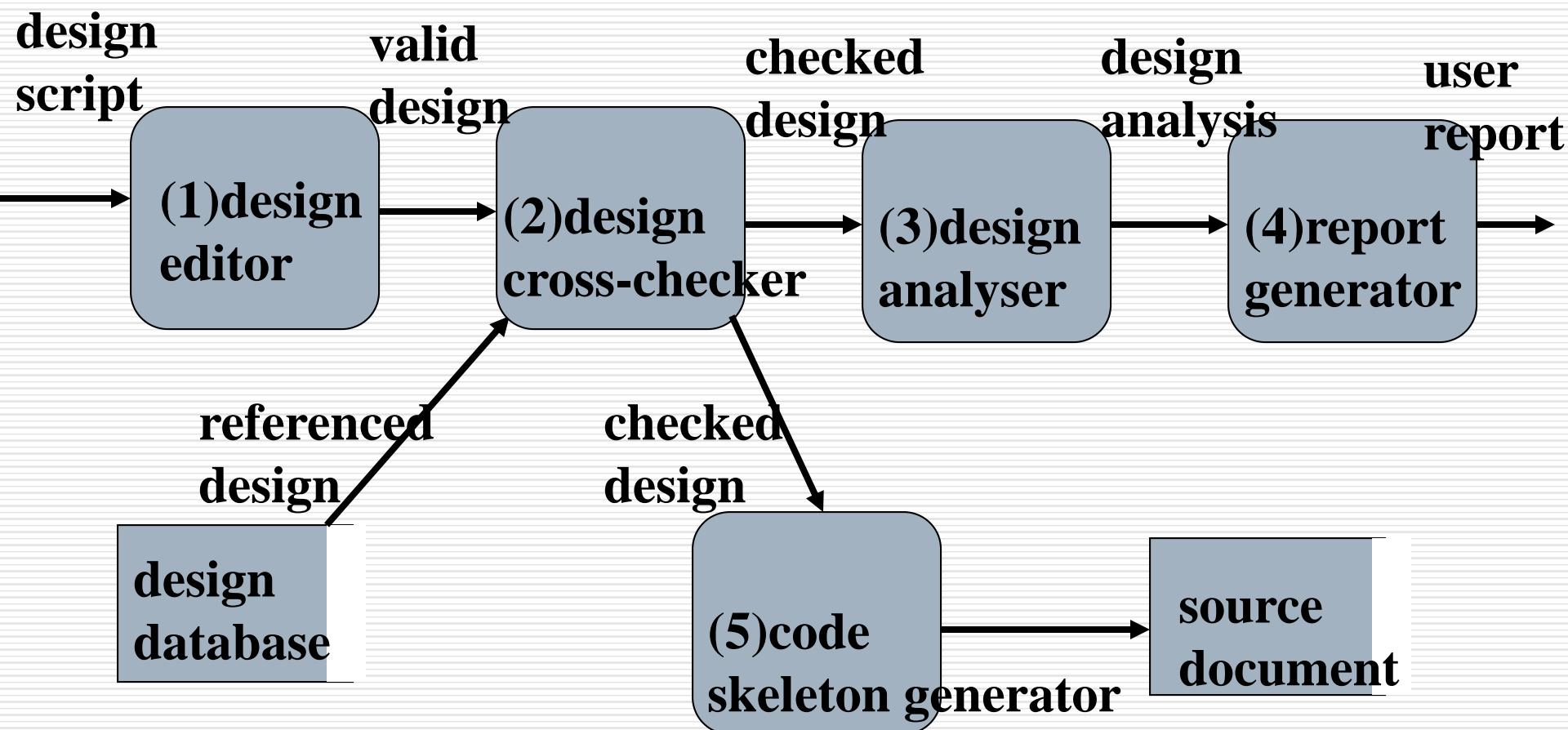


# 售书系统



# \*a example: DFD of a CASE toolset

CASE:computer aid software engineering, ROSE.



(1) the DFD of design editor .....

# Data Flow Diagramming Rules

---

- **Basic rules** that apply to all DFDs
  - Inputs to a process are always different than outputs
  - Objects always have a unique name
    - ✓ In order to keep the diagram uncluttered, you can repeat data stores and sources/sinks on a diagram

# Data Flow Diagramming Rules

## ➤ Process

- No process can have only outputs (a miracle)
- No process can have only inputs (black hole)
- A process has a verb phrase label

## ➤ Data Store

- Data cannot be moved directly from one store to another
- Data cannot move directly from an outside source to a data store
- Data cannot move directly from a data store to a data sink
- Data store has a noun phrase label

# Data Flow Diagramming Rules

---

## ➤ Source/Sink

- Data cannot move directly from a source to a sink
- A source/sink has a noun phrase label

## ➤ Data Flow

- A data flow has only one direction of flow between symbols
- A fork means that exactly the same data goes from a common location to two or more processes, data stores or sources/sinks

# Data Flow Diagramming Rules

---

## □ Data Flow (Continued)

- A join means that exactly the same data comes from any two or more different processes, data stores or sources/sinks to a common location
- A data flow cannot go directly back to the same process it leaves
- A data flow to a data store means update
- A data flow from a data store means retrieve or use
- A data flow has a noun phrase label

# How to draw DFD (宏观)

---

一般遵循“由外向里”的原则，即先确定系统的边界或范围，再考虑系统的内部，先画加工的输入和输出，再画加工的内部。即：

- (1)识别系统的输入和输出。
- (2)从输入端至输出端画数据流和加工，并同时加上文件
- (3)加工的分解“由外向里”进行分解。
- (4)数据流的命名，名字要确切，能反映相应的含义。
- (5)各种符号布置要合理，分布均匀，尽量避免交叉线。
- (6)先考虑稳定态，后考虑瞬间态。如系统启动后在正常工作状态，稍后再考虑系统的启动和终止状态。

# How to draw DFD (细则)

---

See: [ch3--Requirement analysis DFD model--数据流图画法实践.docx](#)

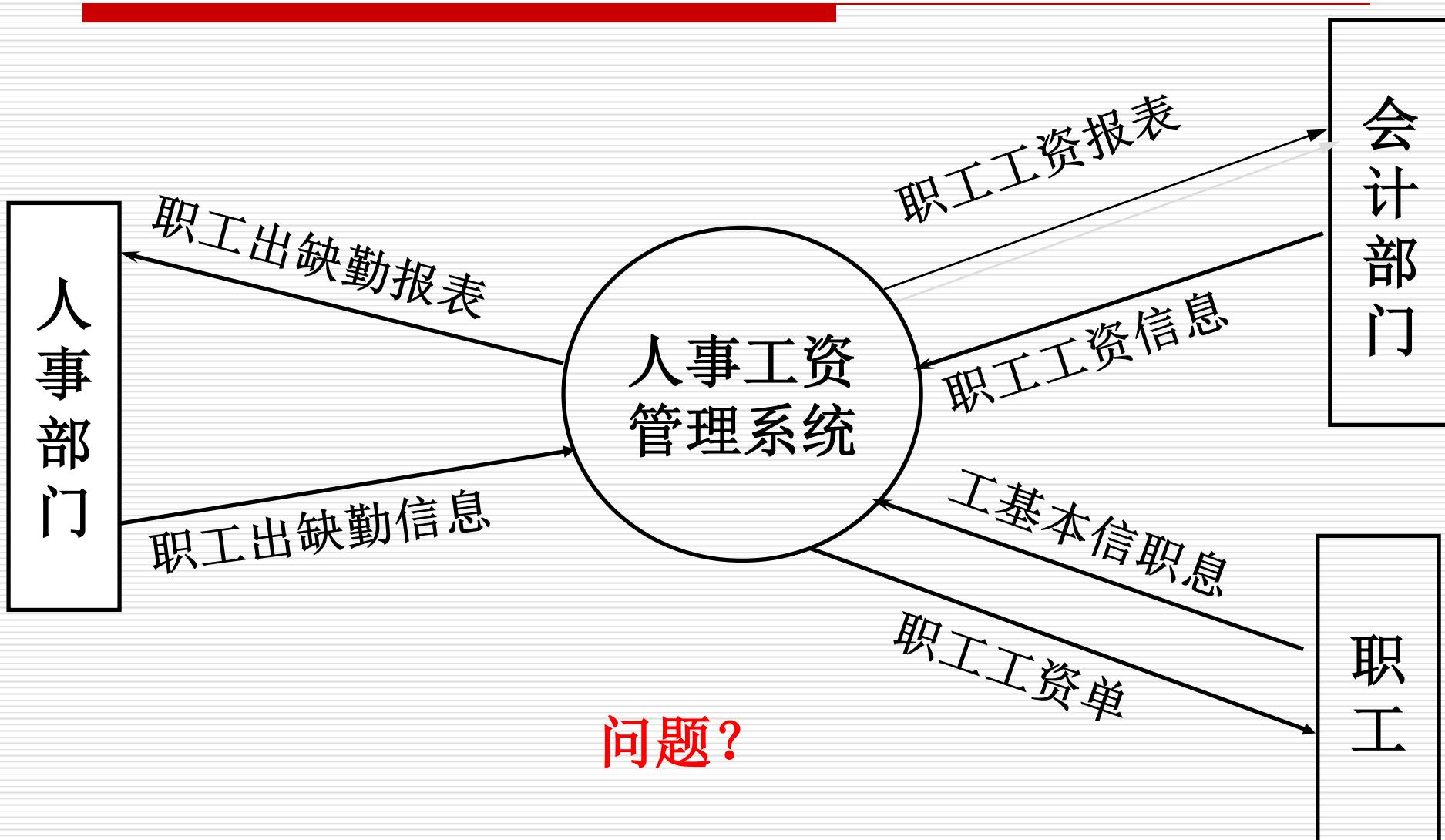
**Do some homework!**

# How to draw DFD

---

- ✓ **find data flow:** data, inf. message, number, value, quantity, metric, list, table, input, output...
- ✓ **find processing:** predicate sentence.
- ✓ **store of data:** file, directory, database
- ✓ **source and destination:** interface, boundary
- ✓ **do not interrupt data flow as possible.**
- ✓ **sometimes, sequence No.**
- ✓ contain  $7 \pm 2$  blocks ----> hierarchy---->top-down

# 人事工资管理系统的顶层DFD范例



# Data Flow Diagram 分层

# 数据流图大小、粒度

# 数据流图详细—简单度

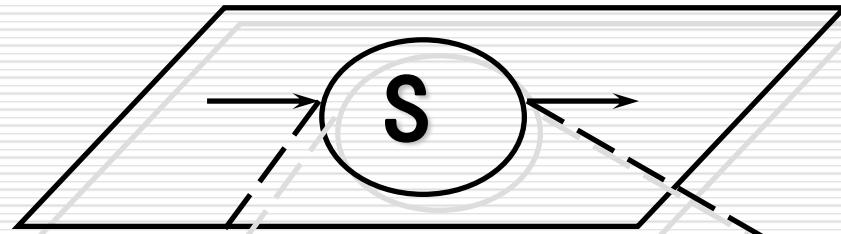
# 极端：最粗



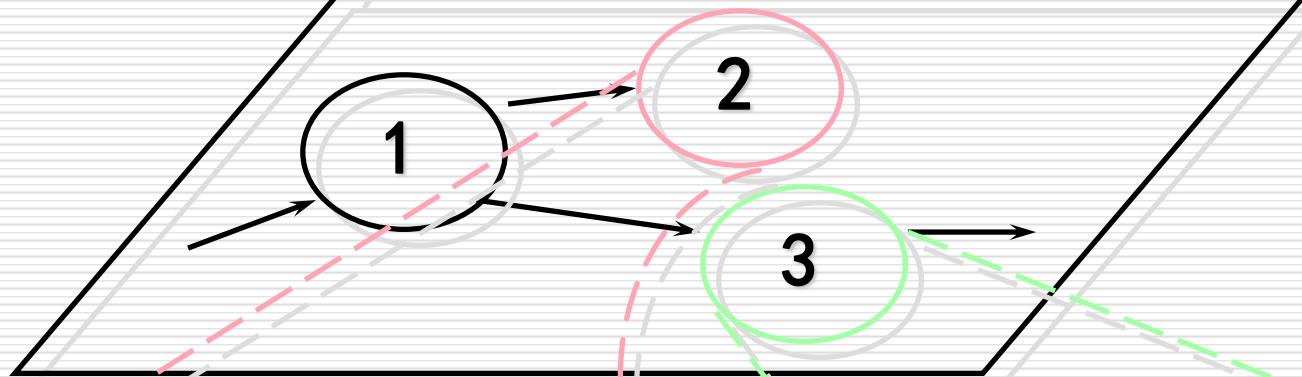
## 平衡

**最細**

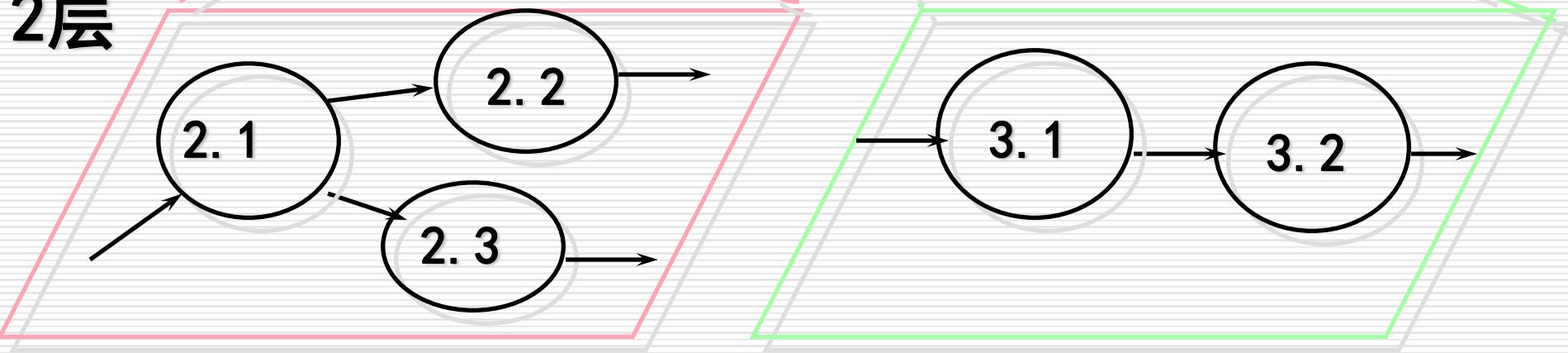
顶层  
(0 层)



1层



2层



## (1) 识别系统的输入和输出，画出顶层图(0级)

即确定系统的边界。在系统分析初期，系统的功能需求等还不是很明确，为了防止遗漏，不妨先将范围定得大一些。系统边界确定后，那么越过边界的的数据流就是系统的输入或输出，将输入与输出用加工符号连接起来，并加上输入数据来源和输出数据去向就形成了顶层图。

## (2) 画系统内部的数据流、加工与文件，画出一级细化图

从系统输入端到输出端(也可反之)，逐步用数据流和加工连接起来，当数据流的组成或值发生变化时，就在该处画一个“加工”符号。

画数据流图时还应同时画上文件，以反映各种数据的存贮处，并表明数据流是流入还是流出文件。最后，再回过头来检查系统的边界，补上遗漏但有用的输入输出数据流，删去那些没被系统使用的数据流。

### (3) 加工的进一步分解，画出二级细化图

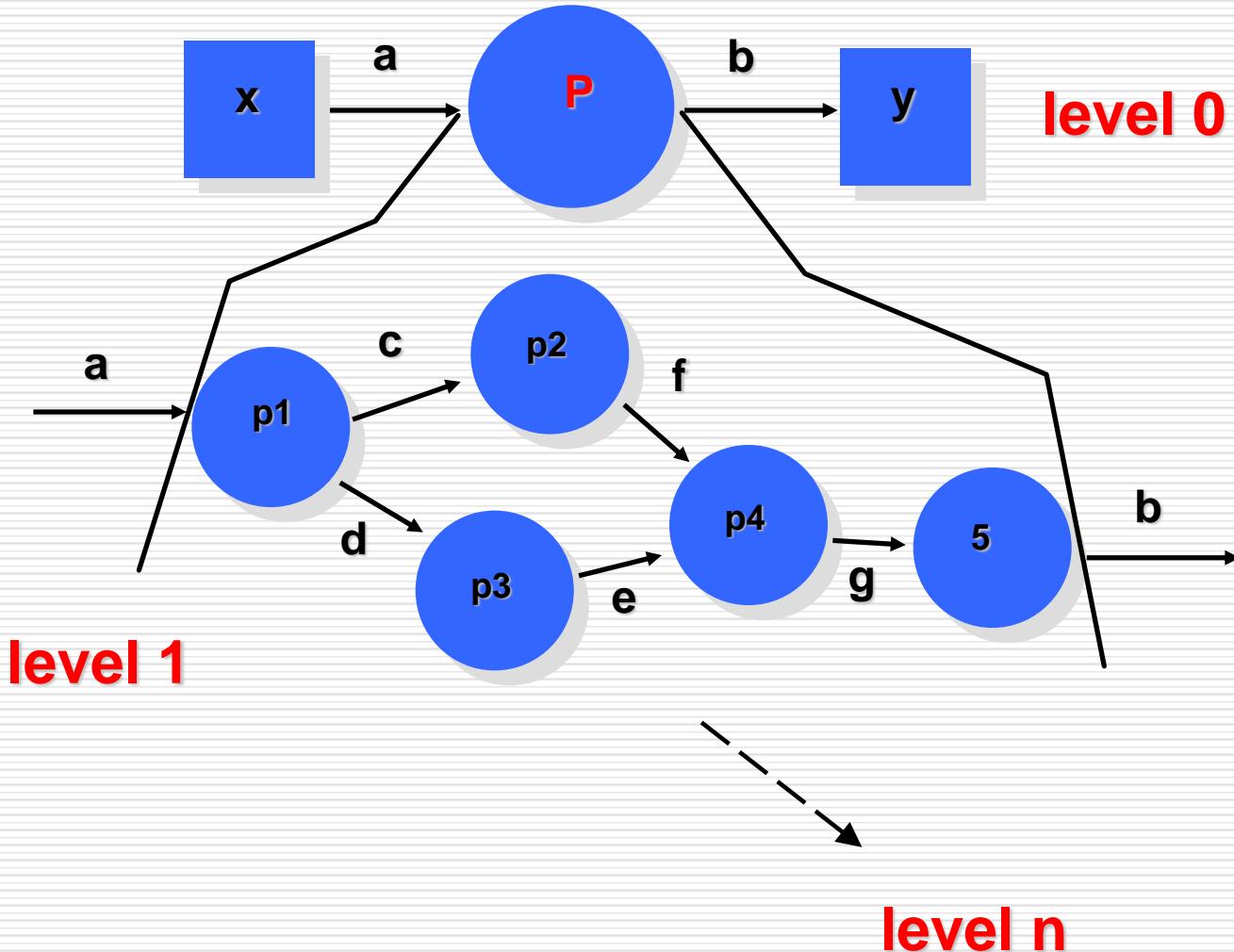
同样运用“由外向里”方式对每个加工进行分析，如果在该加工内部还有数据流，则可将该加工分成若干个子加工，并用一些数据流把子加工连接起来，即可画出二级细化图。二级细化图可在一级细化图的基础上画出，也可单独画出该加工的二级细化图，二级细化图也称为该加工的子图

### (4) 其它注意事项

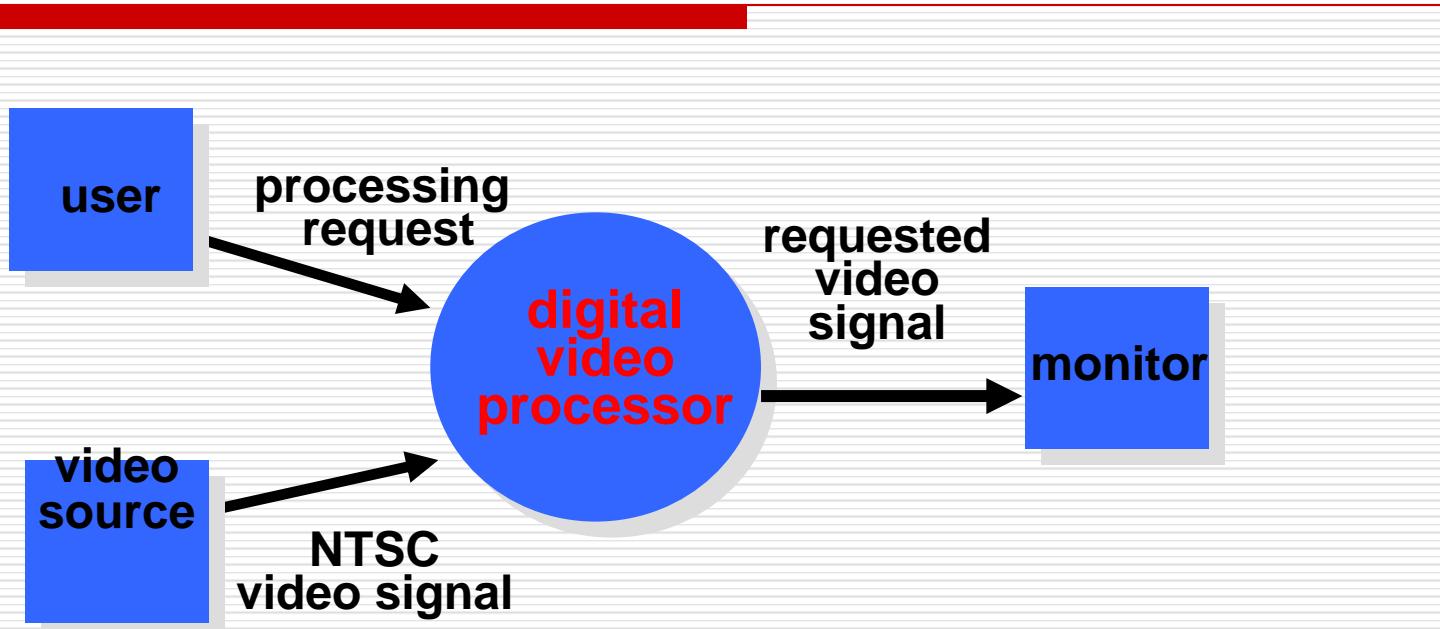
一般应先给数据流命名，再根据输入/输出数据流名的含义为加工命名。名字含义要确切，要能反映相应的整体。若碰到难以命名的情况，则很可能是分解不恰当造成的。应考虑重新分解。

从左至右画数据流图。通常左侧、右侧分别是数据源和终点，中间是一系列加工和文件。正式的数据流图应尽量避免线条交叉，必要时可用重复的数据源、终点和文件符号。此外，数据流图中各种符号布置要合理，分布应均匀。

# The Data Flow Hierarchy



# Level 0 DFD Example



# 分级、扩展、细化

---

- write a narrative describing the transform
- parse to determine next level transforms
- “balance” the flow to maintain data flow continuity
- use a 1:5 (approx.) expansion ratio ( 3 or 7)
- each bubble is refined until it does just one thing
- most systems require between 3 and 7 levels for an adequate data flow model

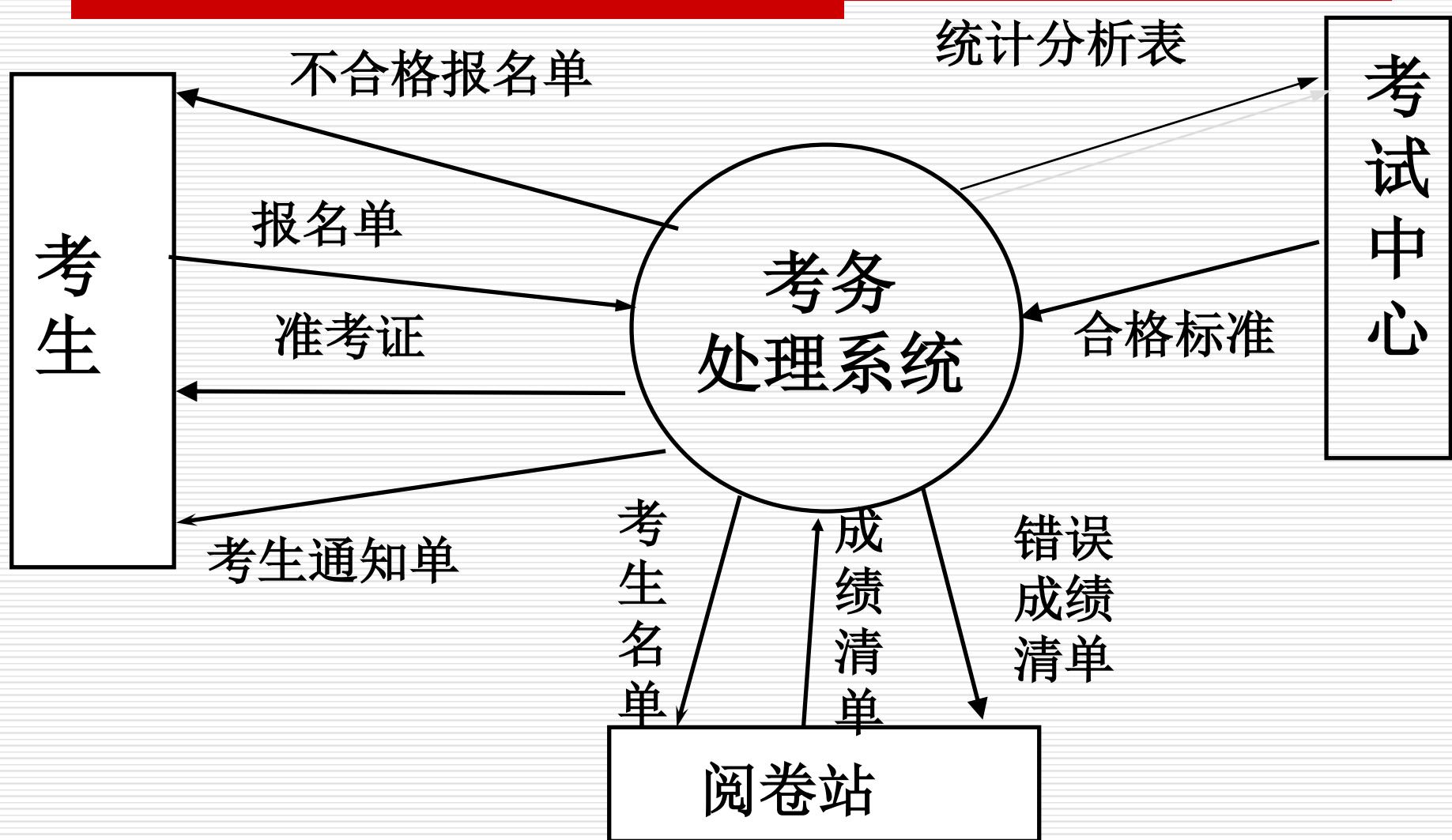
# 分层DFD举例

# 例1：考务处理系统功能描述

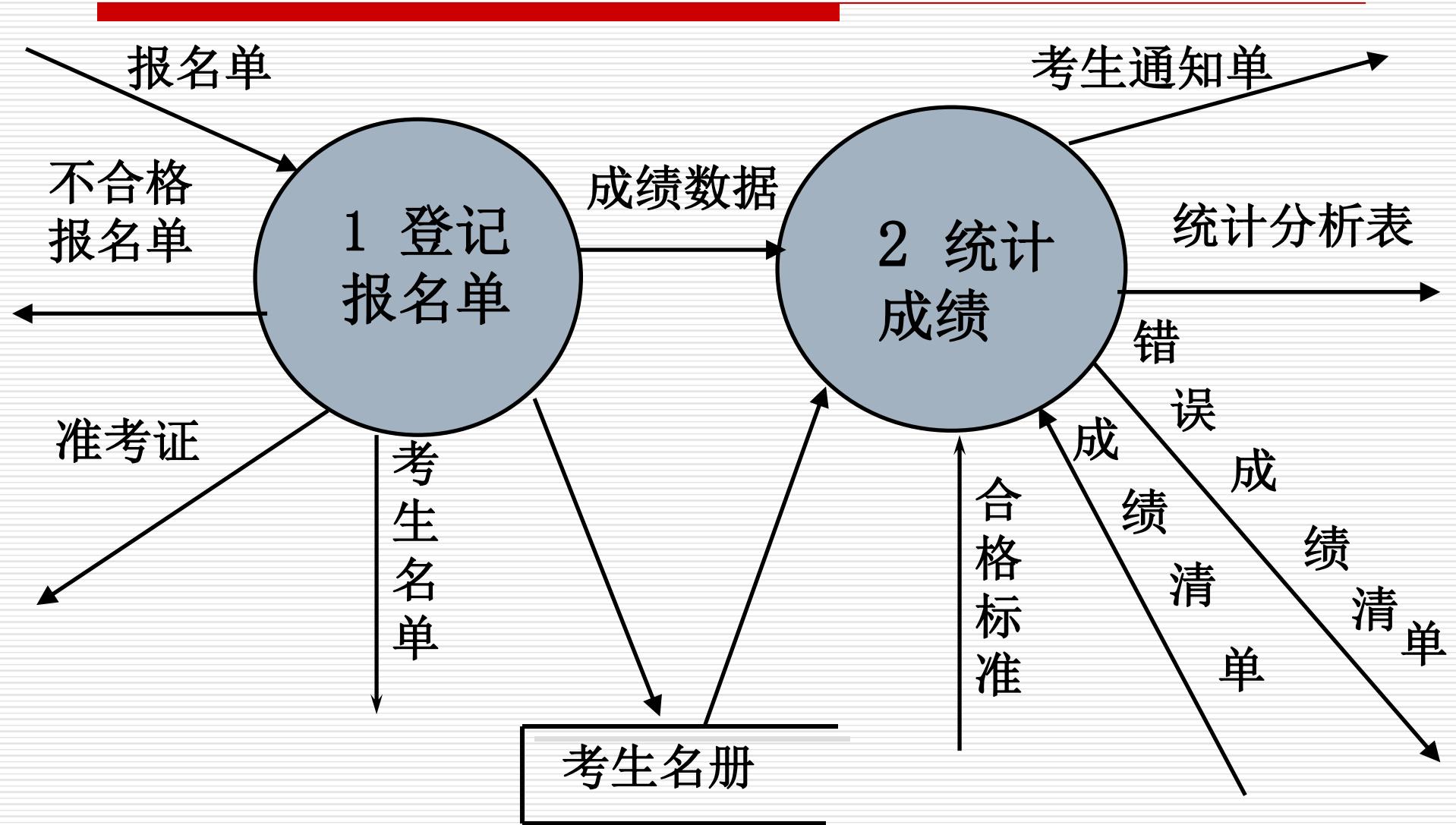
---

- (1)对考生送来的报名单进行检查;
- (2)对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站;
- (3)对阅卷站送来的成绩单进行检查，并根据考试中心制定的合格标准审定合格者;
- (4)制作考生通知单(含成绩及合格/不合格标志)送给考生;
- (5)按地区进行成绩分类统计和试题难度分析，产生统计分析表。

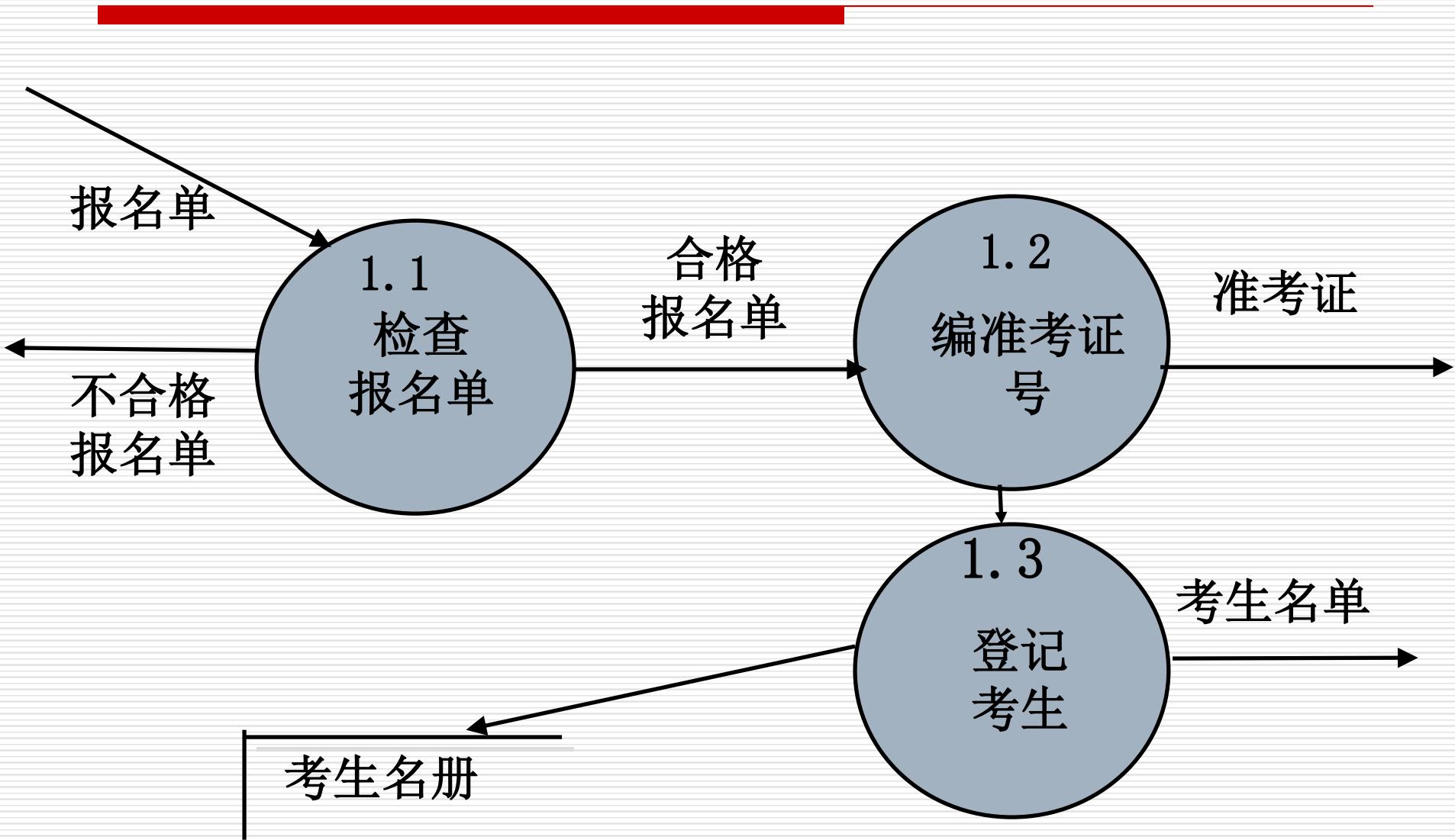
# 顶层数据流图



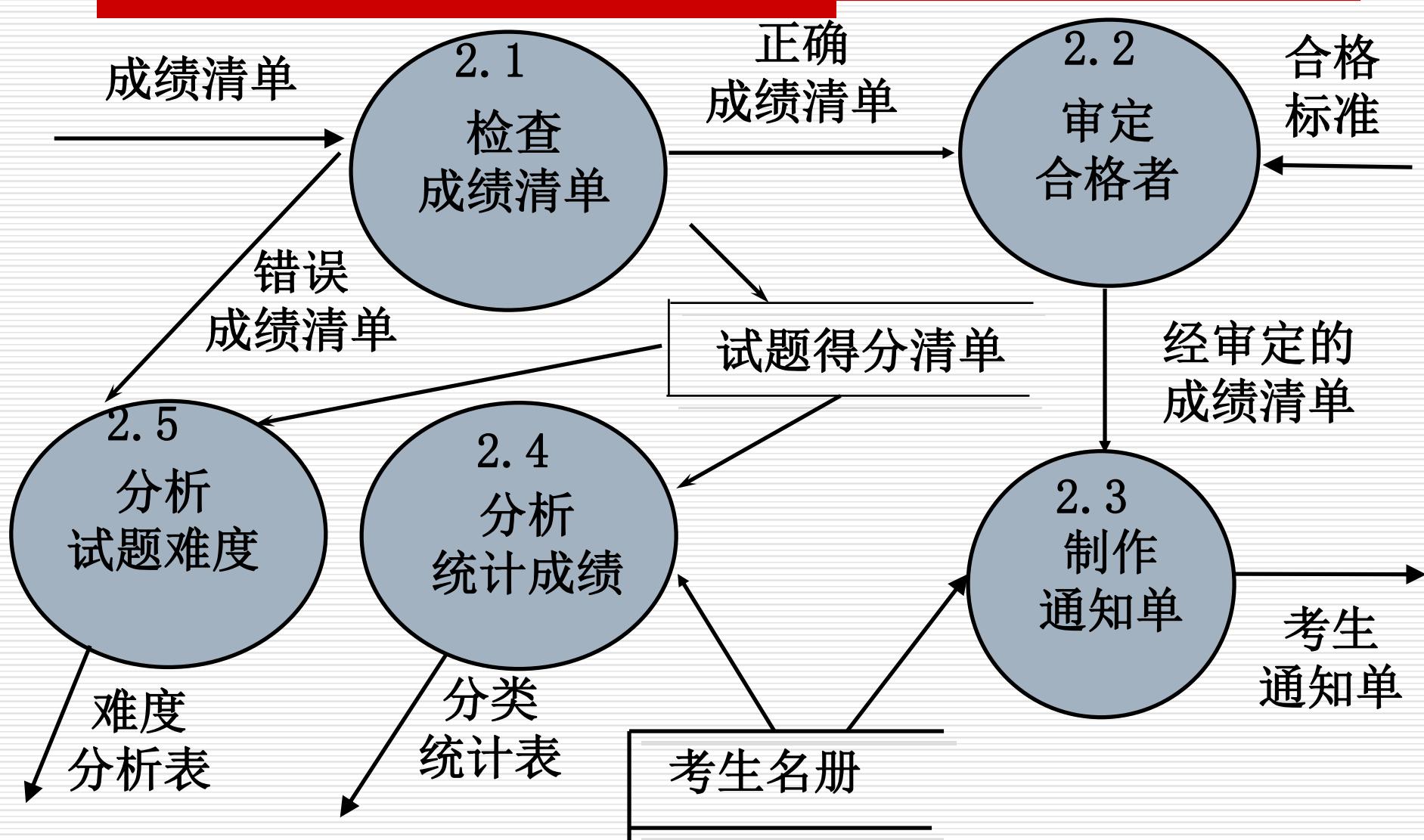
# 1 层数据流图



## 2 层数据流图 (a)



## 2 层数据流图 (b)

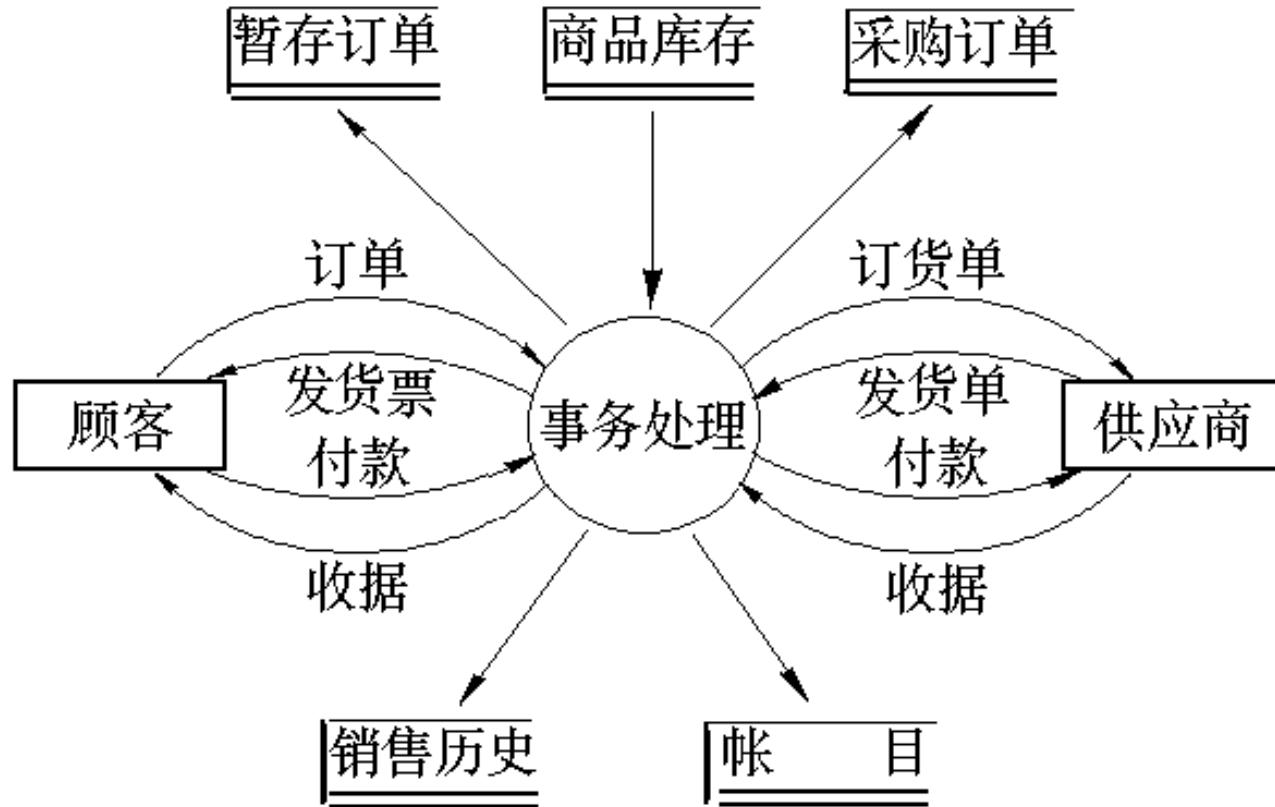


## 例2：商店业务处理系统问题描述

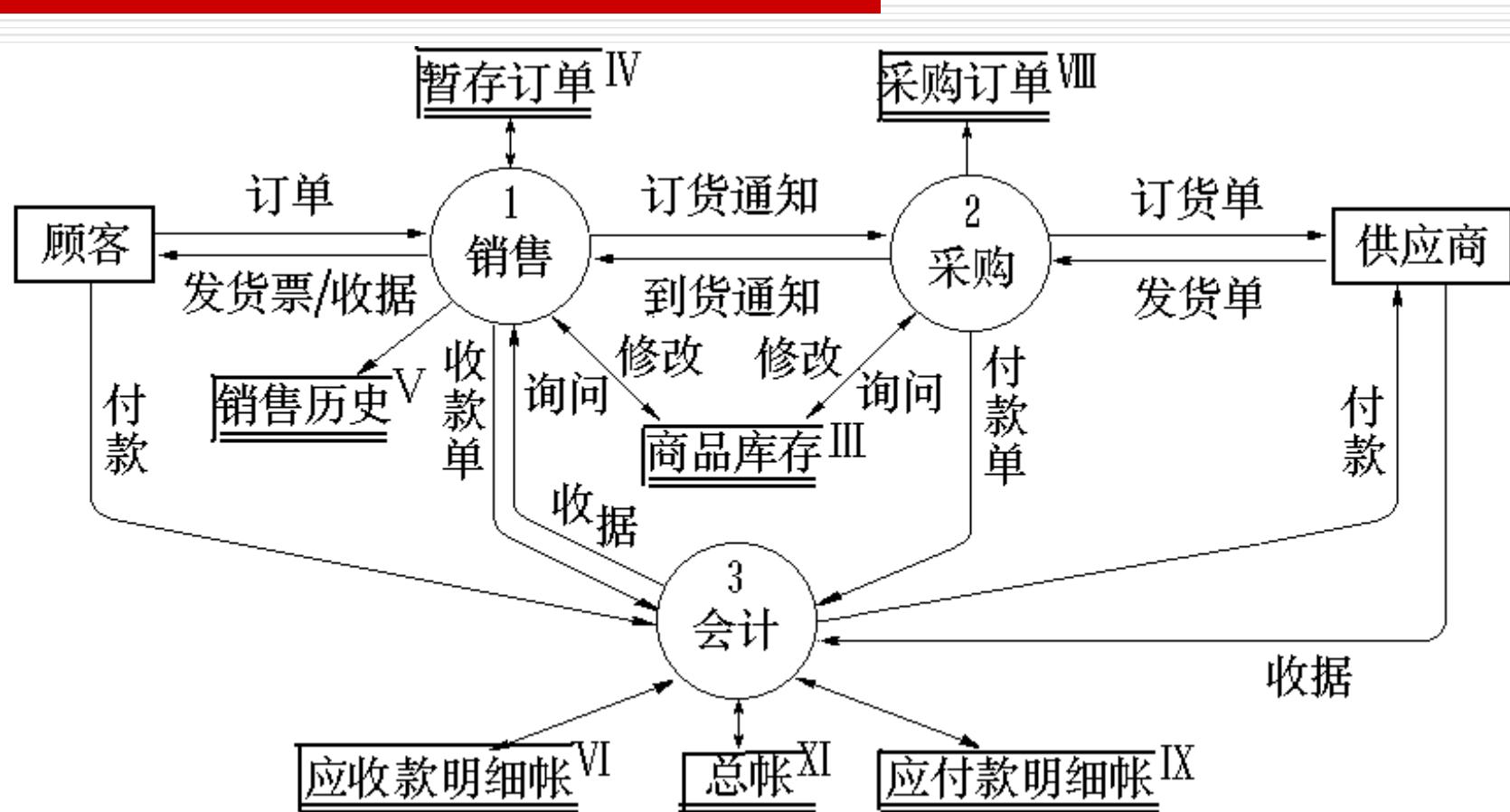
---

- 商店业务处理的主要功能应当有销售、采购、会计三大项。主要数据流输入的源点和输出终点是顾客和供应商。

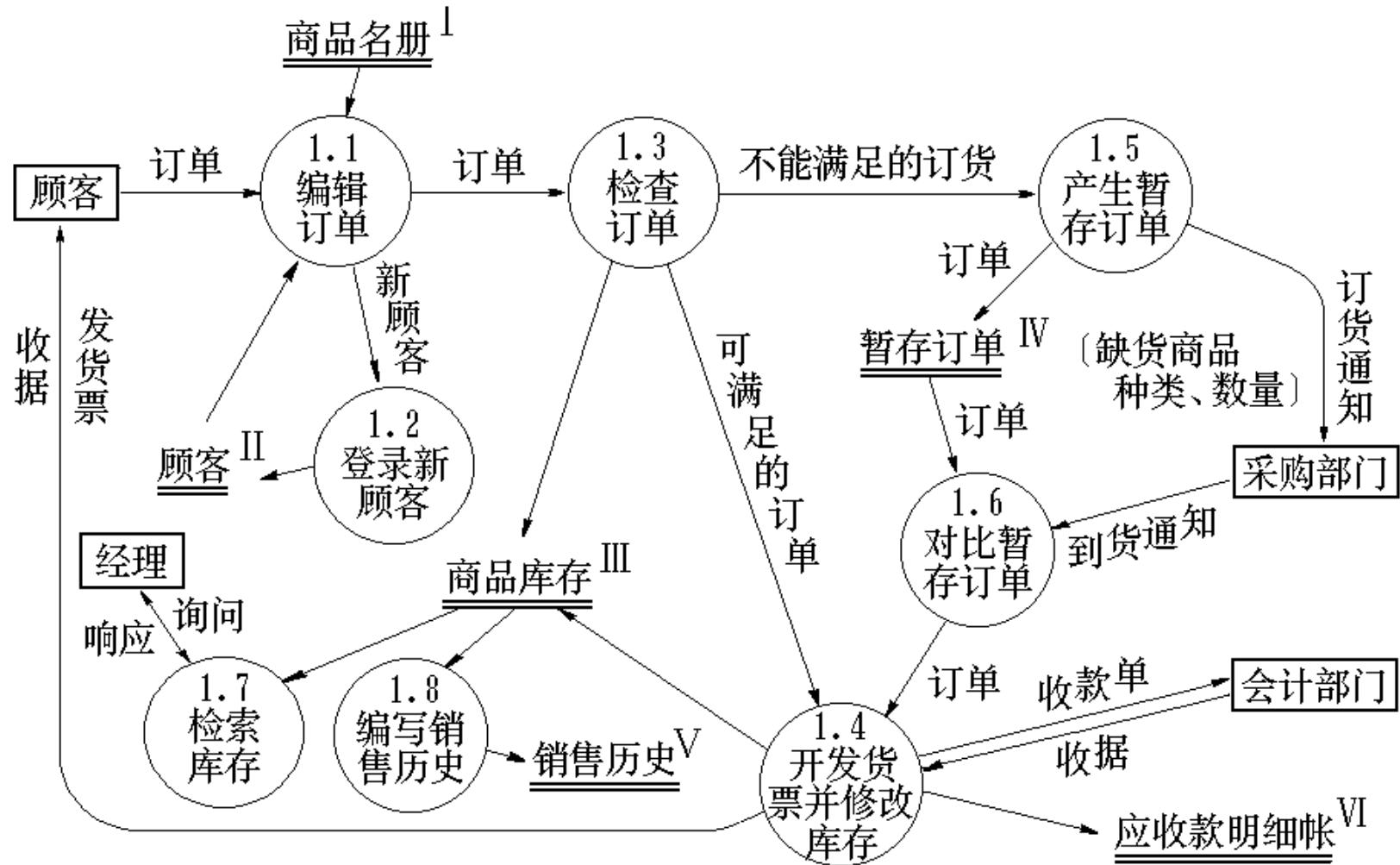
# 0层（顶层）数据流图



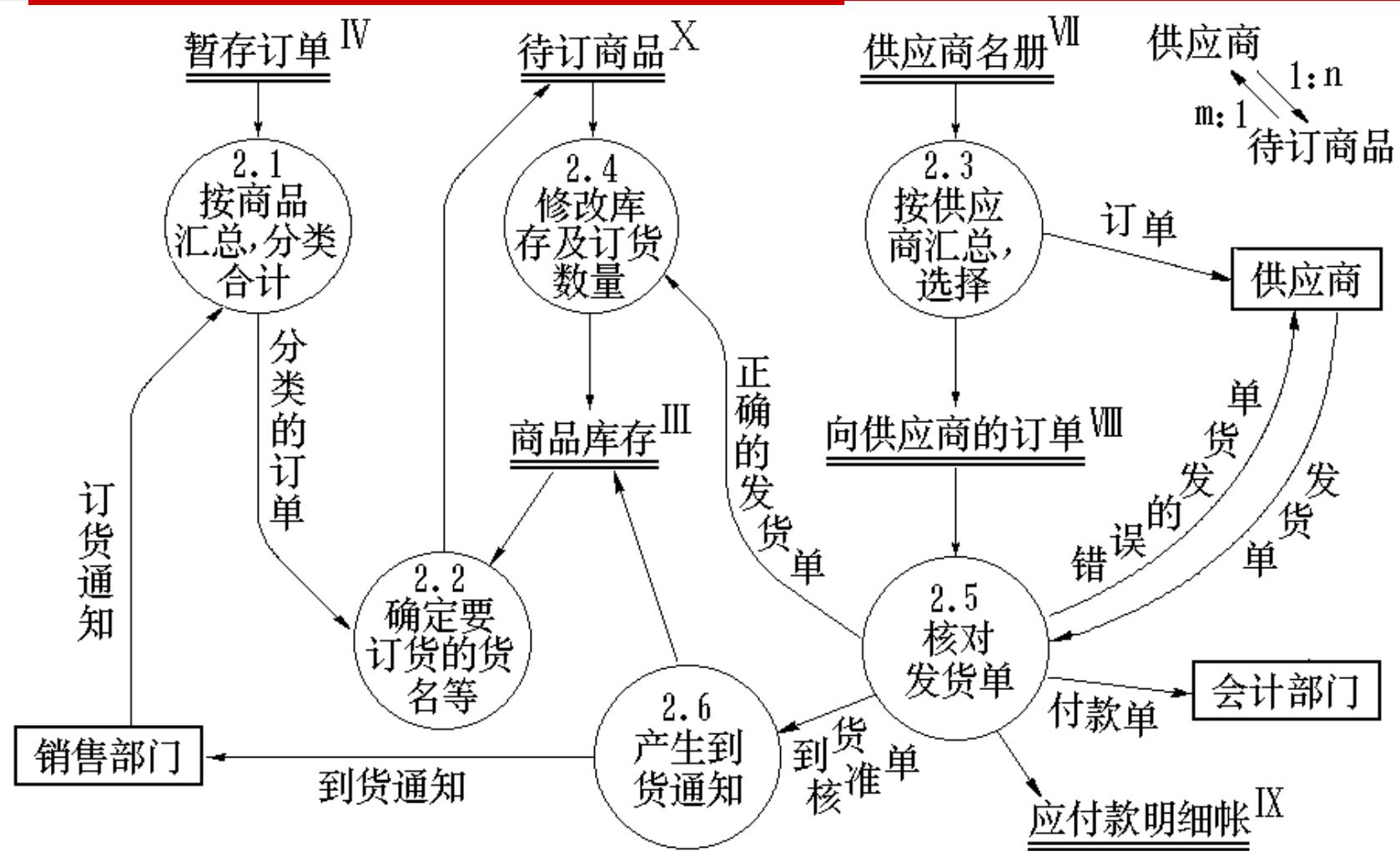
# 第1层数据流图



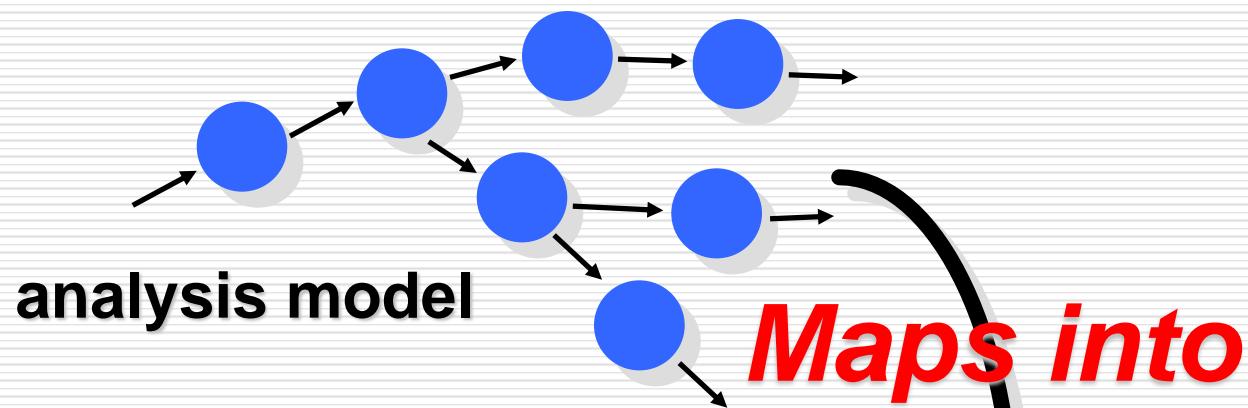
# 第2层数据流图——销售细化



# 第2层数据流图——采购细化



# DFDs: 对编程的指导作用



design model

# Drawback: Data Flow Diagram (缺点)

---

- ✓ Cannot give a precise and detailed definition.
- ✓ Cannot simulate the system with DFDs.
- ✓ Extensions:
  - Integrate with different description, i.e., STD
  - Augment DFD with control flow,
  - Revise DFD, to make it fully formal.

## (2) The Data Dictionary

---

- a quasi-formal grammar for describing the content of data that the software will process and create.
- a notation for describing control data and the values that control data can take, e.g., “on” or “off”
- a repository that also contains “where-used” “how used” information
- a notation that can be represented manually, but is best developed using CASE tools

# 数据字典(DD, Data Dictionary)

---

- DD是对所有与系统相关的**数据**元素的一个有组织的列表, 以及**精确的、严格的定义**, 使得用户和系统分析员对于输入、输出、存储成分和中间计算结果等有共同的理解。
  
- 数据, 信息, 知识 -----> 近义词  
概念, 名称, 术语 -----> 广义上理解

# 数据字典(DD)

- DD是数据信息的**集合**
- 是对**数据流图**中包含的所有所有元素的定义的集合。  
即它是数据流条目、数据存储条目、数据项条目、基本加工条目的汇集。
- 用来定义数据流图中各成分的具体含义，它以一种准确的、无二义性的说明方式为系统分析、设计、维护提供了有关元素的一致的定义和详细的描述。
- 它和数据流图共同构成系统的逻辑模型，是“需求说明书”的主要组成部分。

# “数据流”词条描述的要素

---

- 数据流名:
- 说明: 简要介绍作用即它产生的原因和结果
- 数据流来源: 来自何方
- 数据流去向: 去向何处
- 数据流组成: 数据结构
- 数据量流通量: 数据量, 流通量

# “数据元素”词条描述

---

- 数据元素名：
- 类型：数字（离散值，连续值），文字（编码类型）
- 长度：
- 取值范围：
- 相关的数据元素及数据结构：

# “数据文件”词条描述

---

- 数据文件名：
- 简述：存放的是什么数据
- 输入数据：
- 输出数据：
- 数据文件组成：数据结构
- 存储方式：顺序，直接，关键码
- 存取频率：

# “加工逻辑”词条描述

---

- 加工名:
- 加工编号: 反映该加工的层次
- 简要描述: 加工逻辑及功能简述
- 输入数据流:
- 输出数据流:
- 加工逻辑: 简述加工程序, 加工顺序

# 基本加工逻辑说明

---

- 对数据流图的每一个基本加工，必须有一个基本加工逻辑说明
- 基本加工逻辑说明必须描述基本加工如何把输入数据流变换为输出数据流的加工规则
- 加工逻辑说明必须描述实现加工的策略而不是实现加工的细节
- 加工逻辑说明中包含的信息应是充足的，完备的，有用的，没有重复的多余信息

# 用于写加工逻辑说明的工具

---

- 结构化英语（伪代码PDL）
- 判定表
- 判定树
- 其他

# 源点及汇(终)点词条描述

---

- 名称：外部实体名
- 简要描述：什么外部实体
- 有关数据流：
- 数目：

# 定义式中使用的符号

操作符	含义描述
=	定义为
+	与(顺序结构)
{...}	重复(循环结构)
(..   ..)	或(选择结构)
(.. , ..)	
( ... )	任选
m..n	界域
*..., *	注释符

# 数据结构的描述

符 号	含 义	举 例
=	被定义为	
+	与	$x = a + b$
[..., ...]		
[...   ...]	或	$x = [a, b], \quad x = [a   b]$
{ ... } 或 $m\{...\}n$	重复	$x = \{a\}, \quad x = 3\{a\}8$
(...)	可选	$x = (a)$
“...”	基本数据元素	$x = "a"$
..	连结符	$x = 1..9$

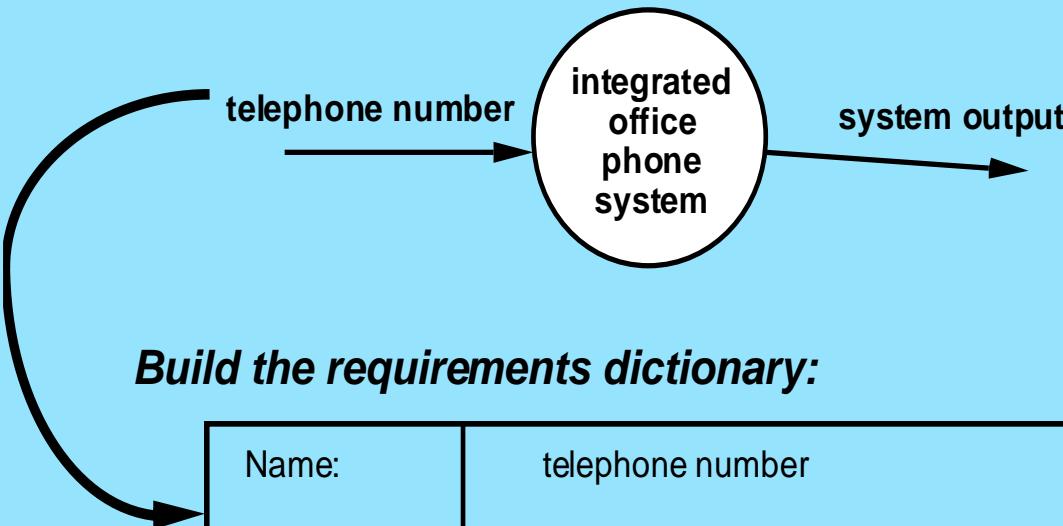
# Building a Data Dictionary

- Name: the primary name of the composite data item
- Aliases: other names for the data item
- Where used: data transforms (processes) that use the composite data item
- How used: the role of the data item (input, output, temporary storage, etc.)
- Description: a notation for representing content (presented on next slide)
- Format: specific information about data types, pre-set values (if known)

# Data Dictionary Notation

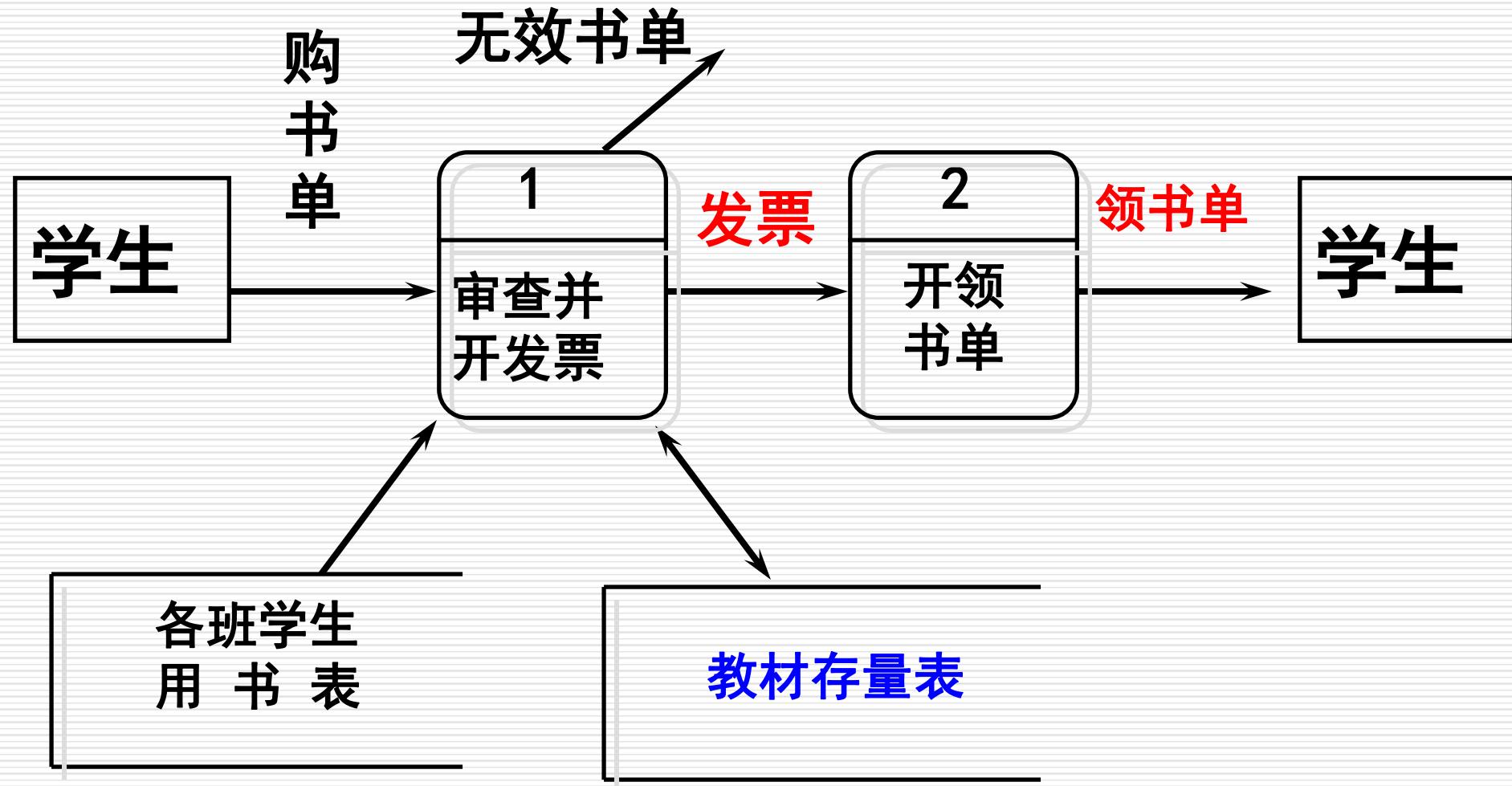
<u>Notation</u>	<u>Meaning</u>
=	is composed of
+	and
[   ]	either-or
{ } <sup>n</sup>	n repetitions of
( ... )	optional data
* ... text ... *	delimits a comment

# Data Dictionary Example



Name:	telephone number
Aliases:	phone number, number
Where/How used:	read-phone-number (input) display-phone-number (output) analyze-long-distance-calls (input)
Description:	telephone no. = [ local extension   outside no.   0 ] outside no. = 9 + [ service code   domestic no. ] service code = [ 211   411   611   911 ] domestic no. = ( ( 0 ) + area code ) + local number area code = *three numeral designator*
Format:	alphanumeric data

# 举例：



# 数据流条目说明举例

数据流名: **发票**

别名: 无

简述: 学生购书时填写的项目

来源: 学生

去向: 加工1 “审查并开发票”

组成: (学号)+姓名+ {书号+数量}

数据流量: 1000次/周

高峰值: 开学期间 1000次/天

# 数据流条目说明举例

数据流名：**领书单**

别名：无

简述：学生购书时依据和凭条

来源：开领书单处

去向：学生

组成：（学号）+姓名+{书号+数量}

数据流量：1000次/周

高峰值：开学期间 1000次/天

# 数据流条目说明举例

存储文档名：**教材存量表**

别名：无

简述：

来源：

组成：

存储方式：

存取频率：

# 数据流条目说明举例

加工名: **开领书单**

加工编号: 2

简要描述: 根据发票的内容, 开出书库认可的领书单

输入数据流: 发票

输出数据流: 领书单

加工逻辑: ...

# 数据流条目说明举例

加工名: ...

加工编号:

简要描述:

输入数据流:

输出数据流:

加工逻辑:

# 银行存折 举例

户名

所号

帐号

开户日

性质

印密

- 存折=户名+所号+帐号+开户日+性质+（印密）+1{存取行}50
- 户名=2{字母}24
- 所号=“001”..“999”
- 帐号=“00000001”..“99999999”
- 开户日=年+月+日
- 性质=“1”..“6”    注：“1”表示普通户，“5”表示工资户等
- 印密=“0”    注：印密在存折上不显示
- 存取行=日期+（摘要）+支出+存入+余额+操作+复核

F1:航班信息文件 = {航空公司名称 + 航班号  
+ 起点 + 终点 + 日期 + 起飞时间 + 降落时间}

航空公司名称 = 2{字母}4

航班号 = 3{十进制数字}3

字母 = “A”...“Z”

十进制数字 = “0”...“9”

起点 = 终点 = 1{汉字}10

起飞时间 = 降落时间 = 时 + 分

时 = “00”...“23”

分 = “00”...“59”

日期 = 年 + 月 + 日

年 = “00”...“99”

月 = “01”...“12”

日 = “01”...“31”

# DD的实现

---

- (1) 人工方法
- (2) 自动方法(利用字典管理程序)

## DD应具特点

- (1) 通过名字可方便查阅数据定义
- (2) 无冗余
- (3) 易更新修改

# DD的实现

➤ IDE，软件工具的  
在线帮助，F1

➤ 举例：Excel  
application，应用  
workbook，工作簿  
worksheet，工作表  
range，区域

row

column

cell

.....

## 浏览 Excel 帮助

- Excel 入门
- 激活 Excel
- 自定义
- 文件管理
- 图表
- 设置条件格式
- 窗体和控件
- 函数参考
- 导入和导出数据
- 安全和隐私
- 宏
- 使用图形
- 拼写和语法



类

- <全局>
- ActiveCell
- ActiveChart
- ActivePrinter
- ActiveSheet
- ActiveWindow
- ActiveWorkbook
- AddIns
- Application
- Calculate
- Cells
- Charts
- Columns
- CommandBars
- Creator
- DDEAppReturnCode
- DDEExecute
- DDEInitiate
- DDEPoke
- DDERequest
- DDETerninate
- Evaluate
- Excel4IntlMacroSheets
- Excel4MacroSheets
- ExecuteExcel4Macro
- Intersect
- Names
- Parent
- Range
- rgbAliceBlue
- rgbAntiqueWhite
- rgbAqua
- rgbAquamarine
- rgbAzure
- rgbBeige
- rgbBisque
- rgbBlack
- rgbBlanchedAlmond
- rgbBlue
- rgbBlueViolet
- rgbBrown
- rgbBurlyWood

# (3) Entity-Relationship (E-R) Modeling

---

## 实体关系图建模分析

ERD is pictorial representation to **entity** and its'  
**relationship** in real-life, but does not care how to  
implement this entity in system.

# 实体关系图

---

- Entity-Relationship Diagram
- 简称ER图
- 可用于描述数据流图中“**数据存贮**”及其之间的关系，它是数据库概念设计的最常用的工具。
- The relational database schemes can be derived from ERD. CASE can do it automatically( eg. Power-Design)

# Entity-Relationship (E-R)

---

## Entity (object)

- ✓ A person, place, object, event or concept in the user environment about which the organization wishes to maintain data
- ✓ Represented by a rectangle in E-R diagrams

## Entity Type

- ✓ A collection of entities that share common properties or characteristics

## Attribute

- ✓ A named property or characteristic of an entity that is of interest to an organization

# Entity-Relationship (E-R)

---

## □ Relationship

- ✓ An association between the instances of one or more entity types that is of interest to the organization
- ✓ Association indicates that an event has occurred or that there is a natural link between entity types
- ✓ Relationships are always labeled with **verb phrases**

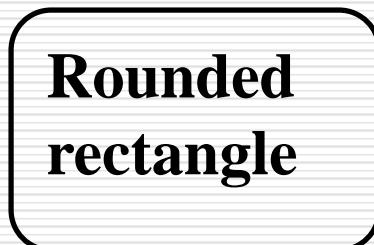
# Symbols for ERD



entity



relationship



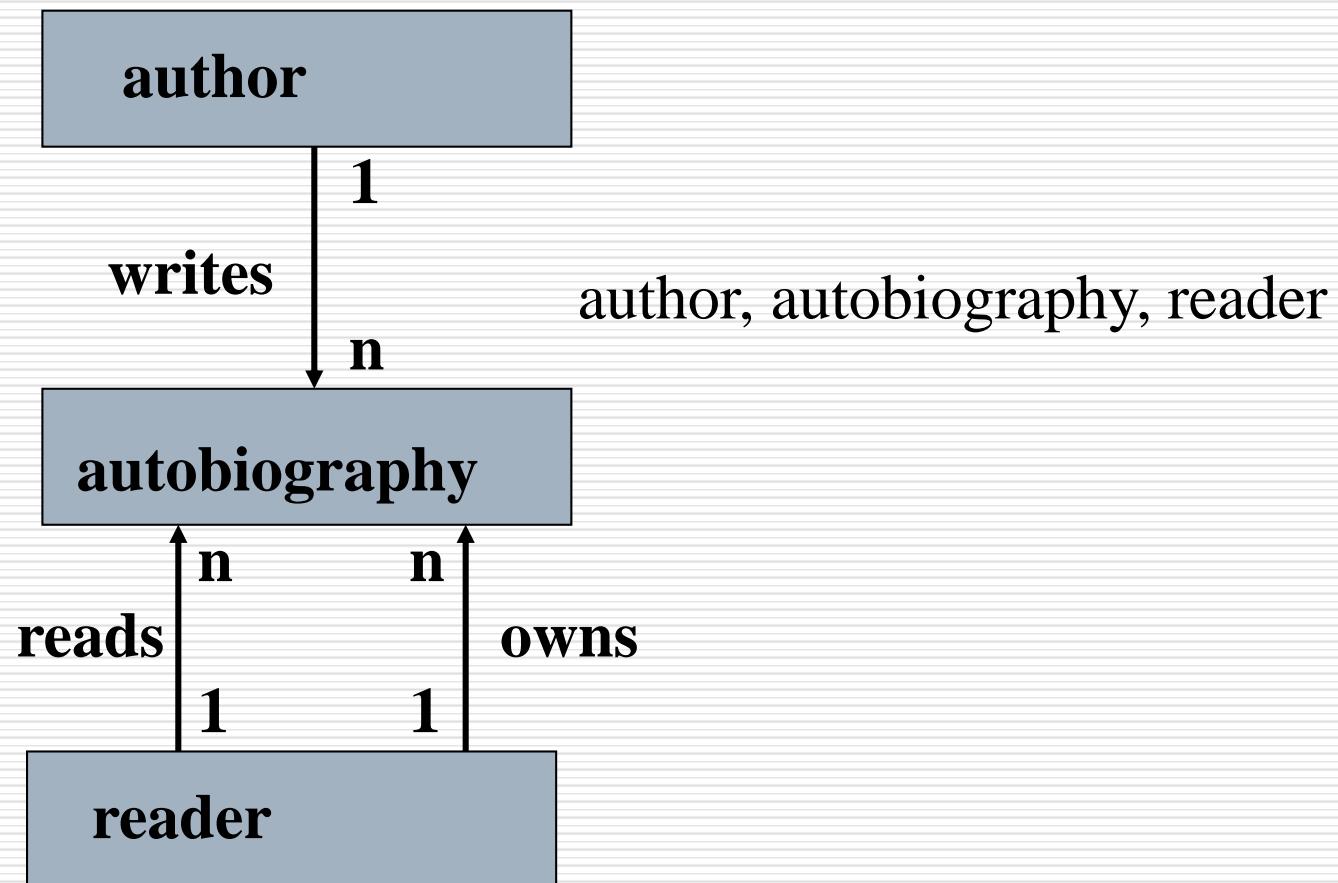
attribute

# 实体联系图的基本成分

---

- 实体用长方形表示
- 实体的属性用椭圆形表示
- 联系用菱形框表示
- 用无向边把实体与其属性连接起来

# A simple example for ERA



# **Another example for ERA**

---

**we wish to build school information system.**

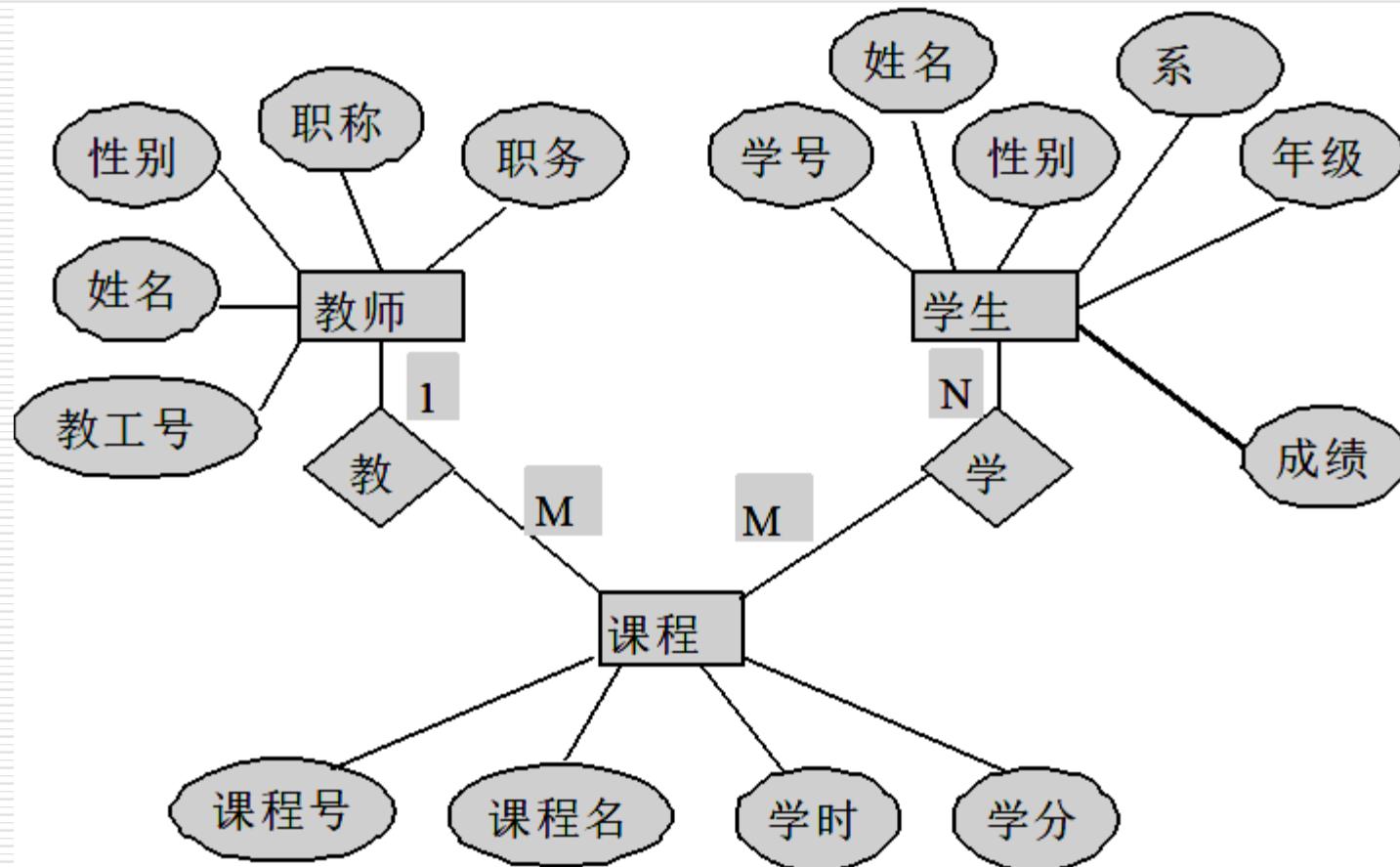
**it concerns teacher, pupil, course, room, .....**

**Teacher:** 姓名, 工号, 性别, 职称, 职务, .....

**Pupil:**

**Course:**

# 实体关系图



# Homework, 9月26日

---

Page 54

T1

T5

Page 73

T2

T3

T5

Send to email: se2003tj@sina.com