

Chapter 7

- **Requirements Modeling: Flow Oriented Modeling**

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 7/e
by Roger S. Pressman

两种需求分析方法

- One view of requirements modeling, called *structured analysis*, considers data and the processes that transform the data as separate entities.

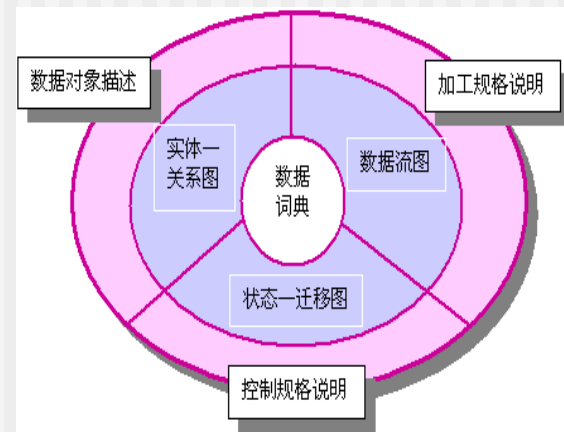
面向功能 ,把系统看成一组功能

- The second approach to analysis modeled, called *object-oriented analysis*,

类是系统的基础, 系统由类组成

结构化分析方法

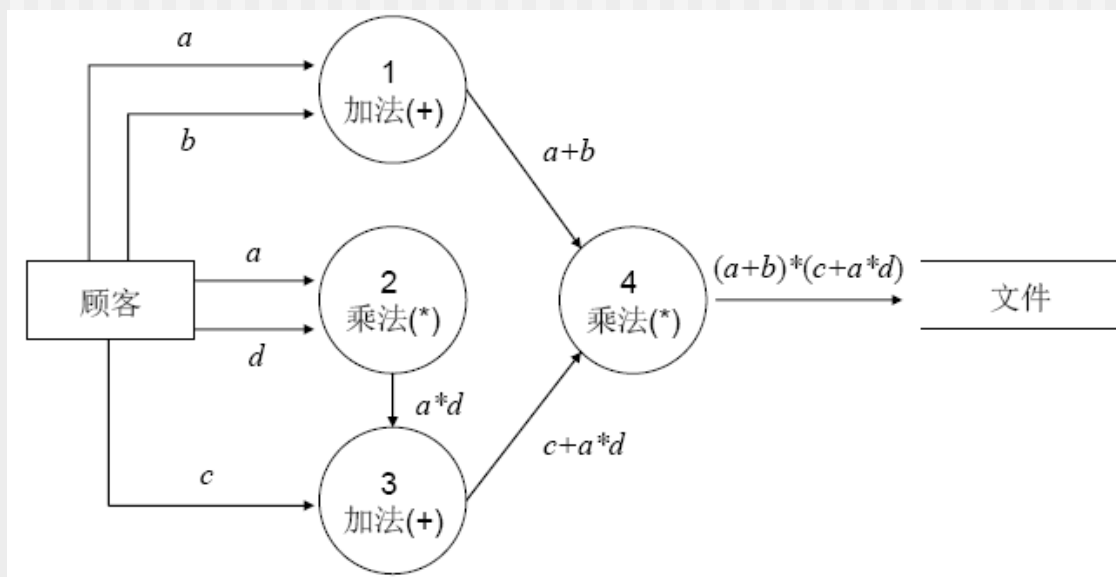
- 结构化分析方法（SA）又称面向数据流的分析方法。
- 主要是创建基于数据流的模型，通过分析数据流发现系统功能和行为，并对其进行划分。
- SA主要构建四种分析模型
 - 数据流图(Data Flow Diagram)——功能视角
 - 数据字典
 - 实体关系图——数据视角
 - 状态迁移图——行为视角



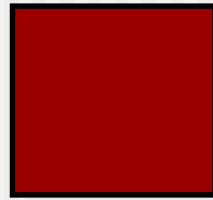
Data Flow Diagram

- 数据流图将系统看作由数据流联系起来的各种功能的组合。
- 通过DFD建模，分析问题域中数据如何流动，以及在各个流动过程中的加工、变化，从而发现数据流和功能。

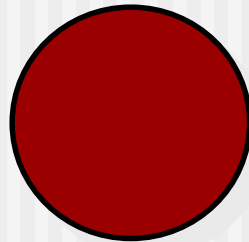
例如：针对需要分析实现的业务 $e=(a+b)*(c+a*d)$ ，可以利用DFD发现若干功能：加法，乘法。



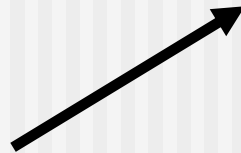
Flow Modeling Notation



external entity



process



data flow



data store

External Entity



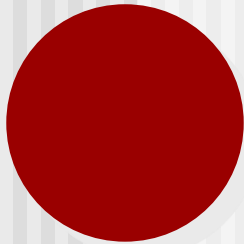
A producer or consumer of data

Examples: a person, a device, a sensor

存在于系统之外的人员、组织或其它系统

表示系统中输入数据流的源点或输出数据流的终点

Process



A data transformer (changes input to output)

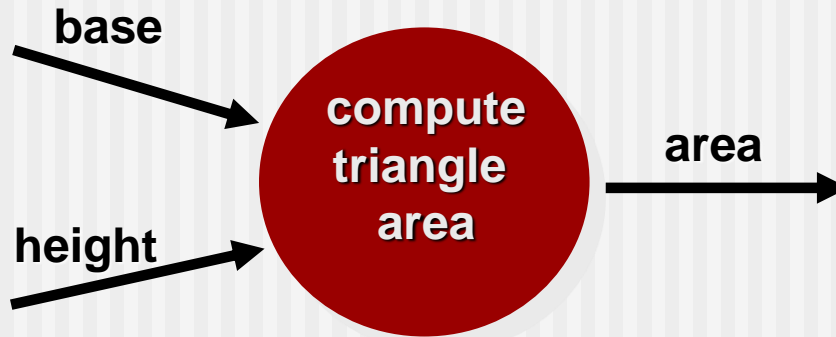
Examples: *compute taxes, determine area, format report, display graph*

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

Data Flow



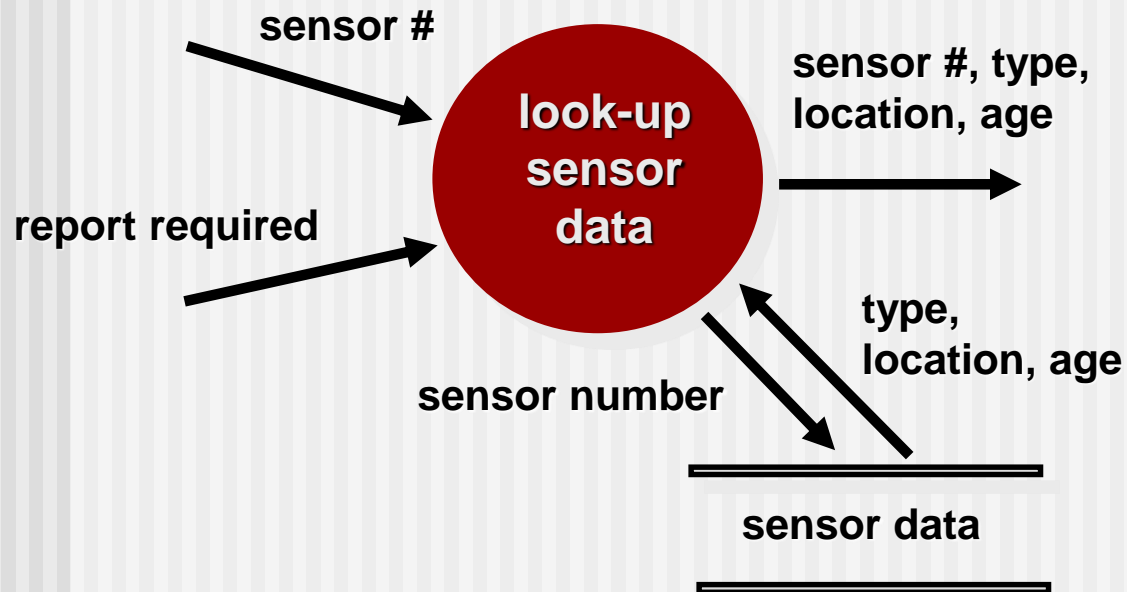
Data flows through a system, beginning as input and transformed into output.



数据流可能存在于：
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

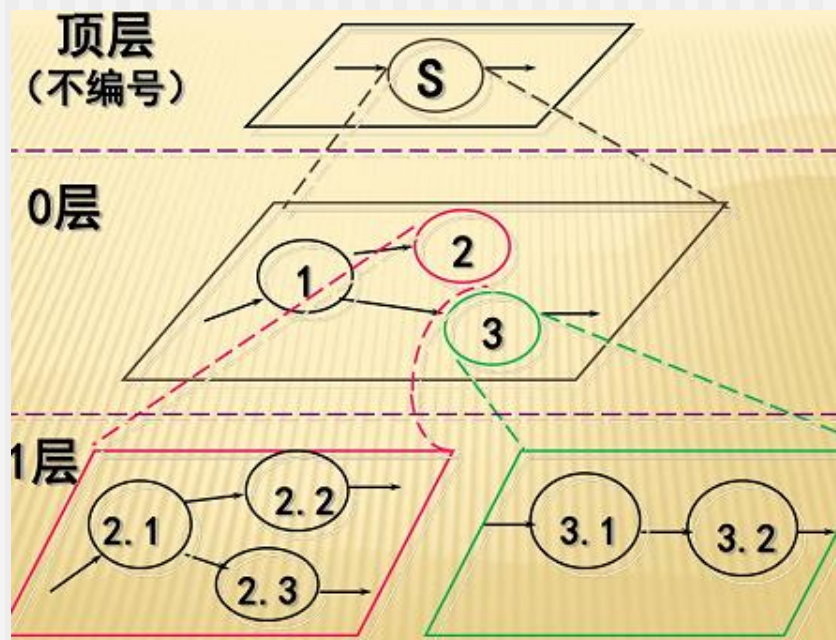
Data Stores

Data is often stored for later use.



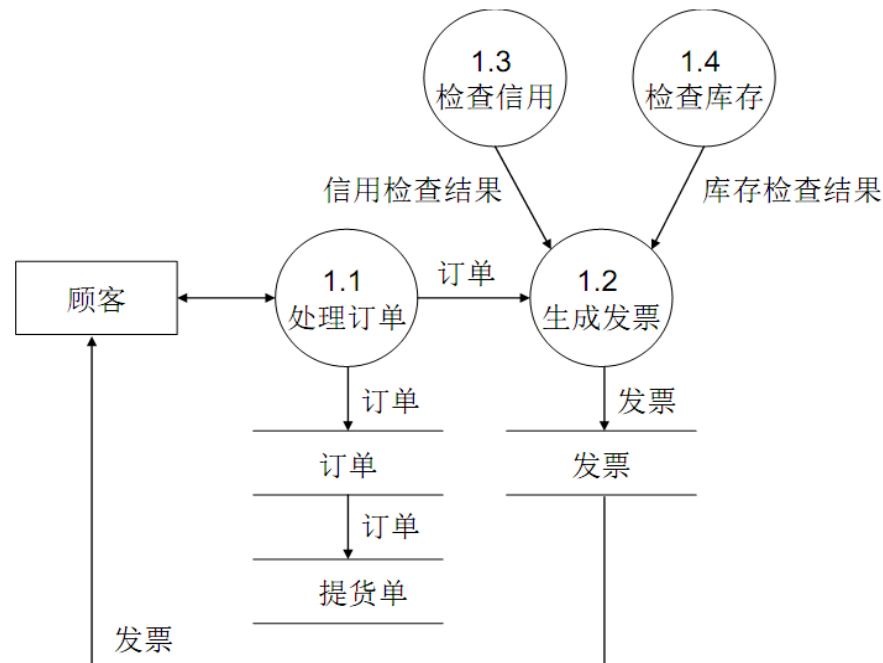
DFD建模方法(1)

- DFD建模采用自顶向下逐层分解的方法
- 通过分层，实现对复杂业务从抽象到具体的递进分析
- DFD图由顶层图、第0层细化图、第1层细化图...等一组图形构成，逐级细化。



DFD建模方法(2)

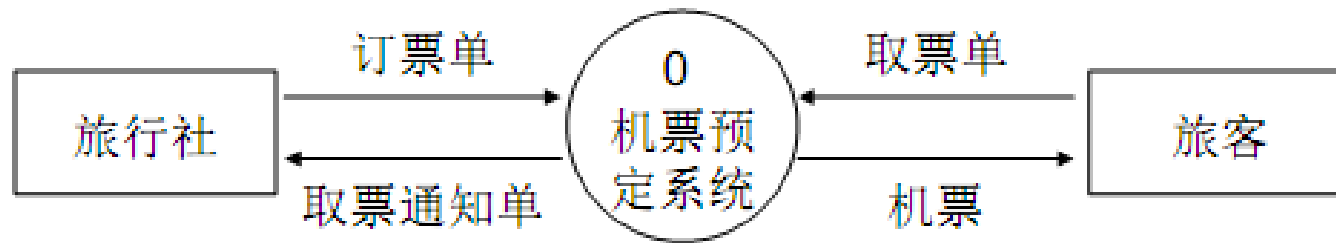
- 每个加工需编号
- 每个加工必须既有输入数据流，又有输出数据流
- 数据不能直接从一个数据存储流入另一数据存储，也不能直接从数据存储到外部实体



DFD建模方法(3)

■ 顶层数据流图

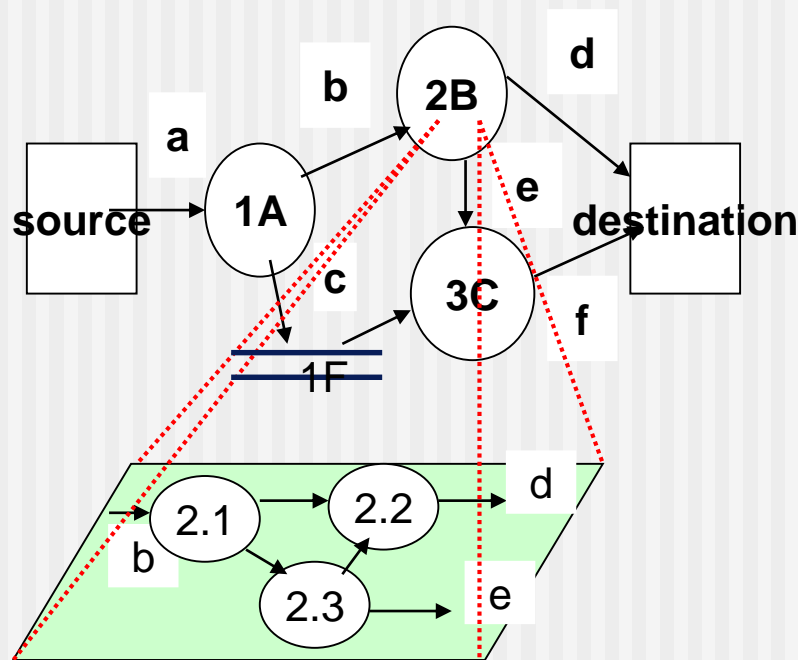
- ✓ 顶层数据流图只含有一个加工，表示整个待开发系统
- ✓ 作用：确定与系统相关的**外部实体**，以及外部实体与系统之间的数据流。



DFD建模方法(4)

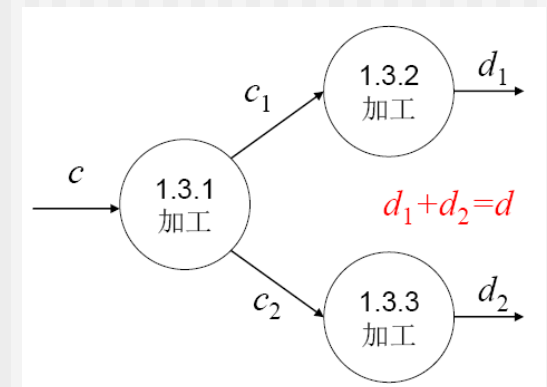
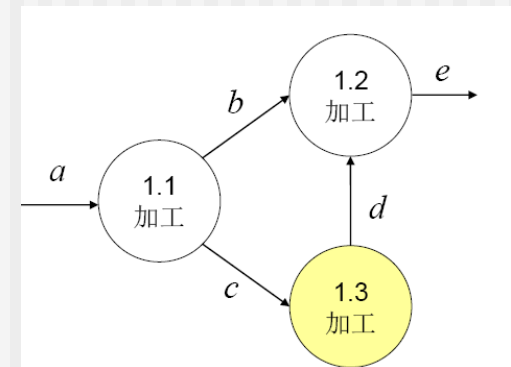
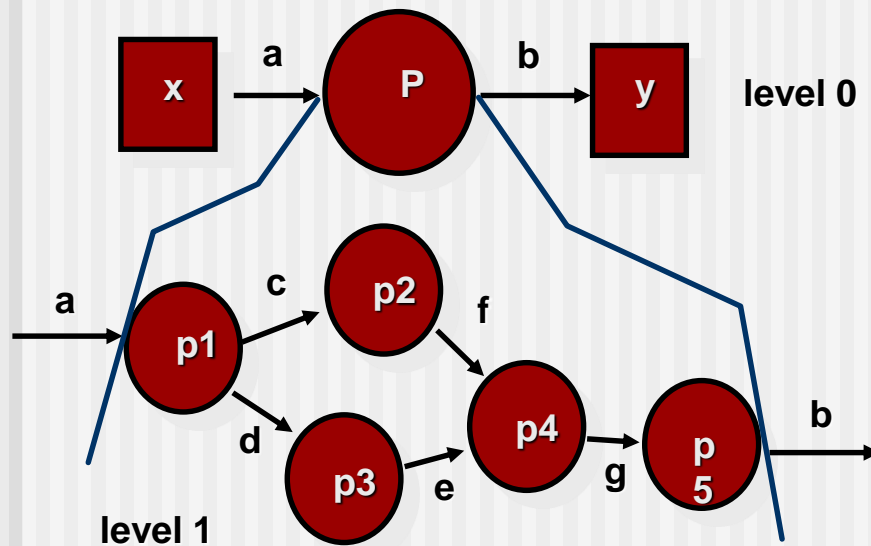
■ DFD子图分解

- ✓ 先将顶层DFD中的系统分解为若干个子系统，每个子系统表示为一个子加工。
- ✓ 继续在下层DFD中对子系统进行分解，直至每个加工足够简单为止。



DFD建模方法(5)

■ 保持父图与子图平衡

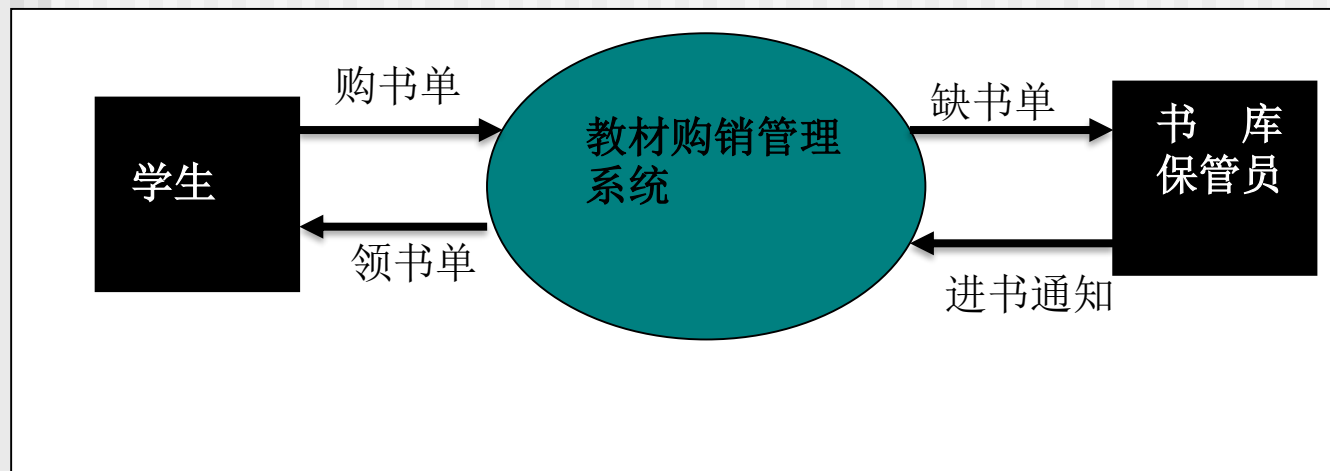


Example of DFD(1)

教材购销系统可分为两个子系统：**销售子系统**和**采购子系统**。

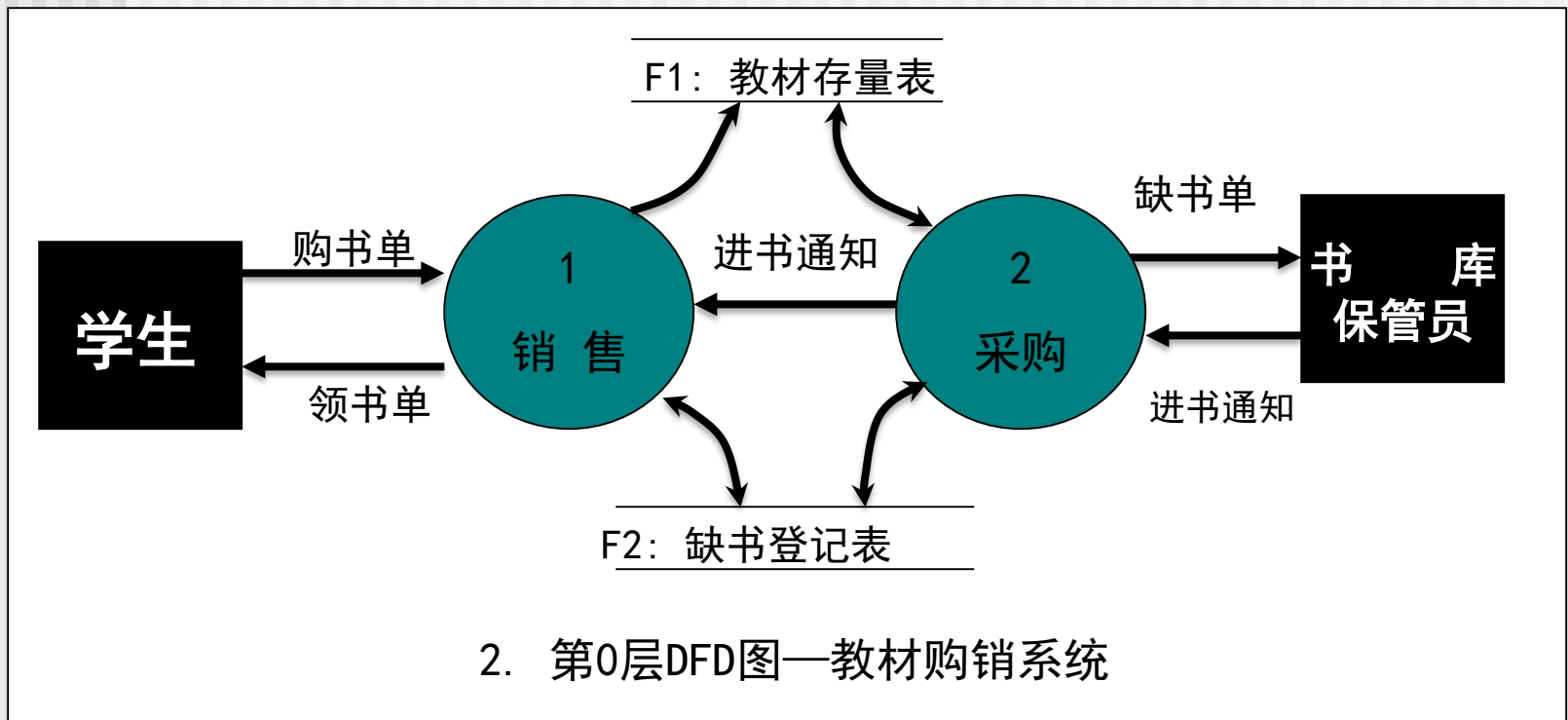
销售子系统：首先由学生提交购书单，经教材科发行人员审核是否有效后，缴费并开发票、登记购书情况并返给学生领书单，学生凭领书单即可领书。

采购子系统：若教材脱销，则登记缺货信息，并发缺货单给书库保管员，书库保管员线下完成采购；新书采购到即入库登记，并发进书通知告知系统。



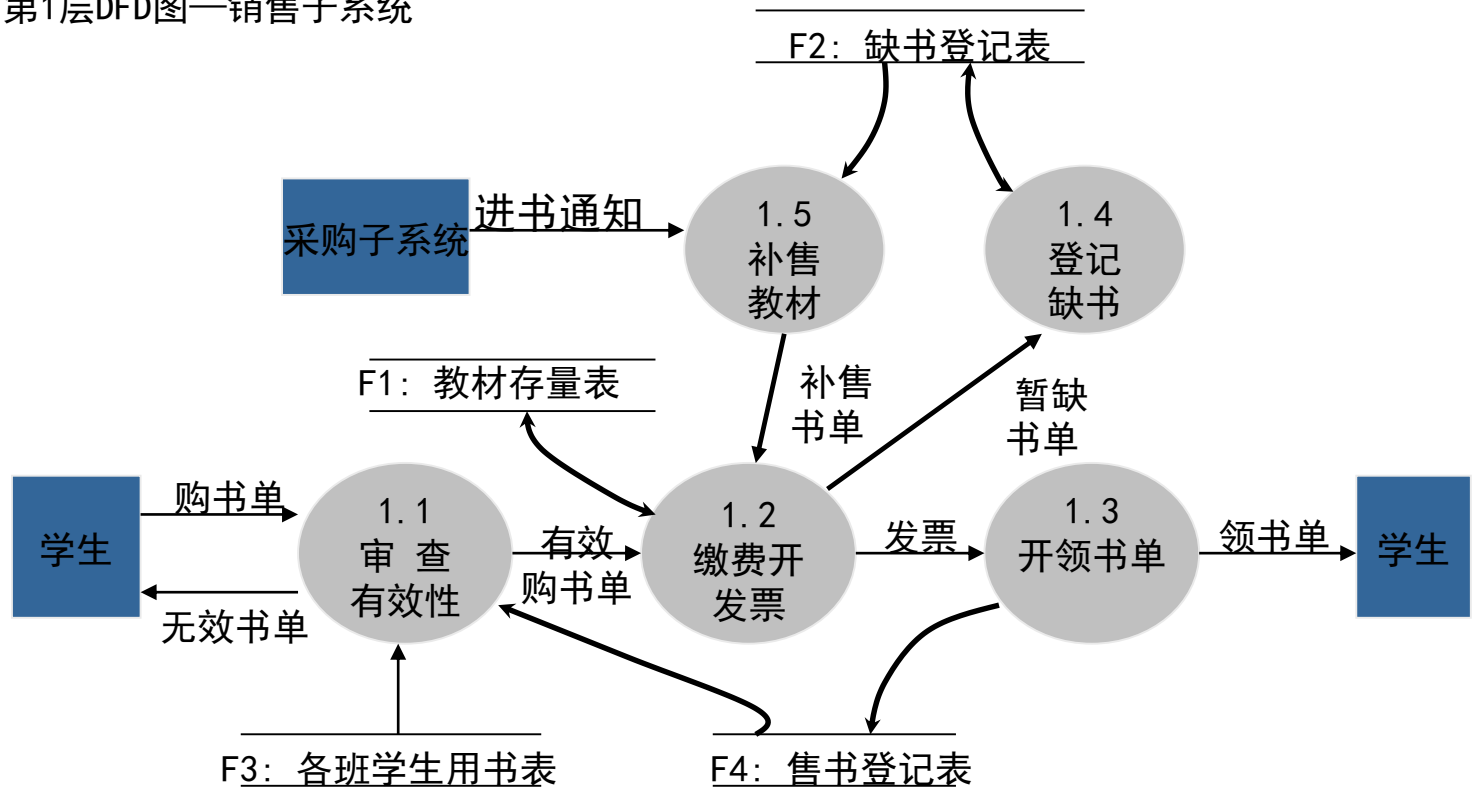
教材购销管理系统顶层DFD

Example of DFD(2)

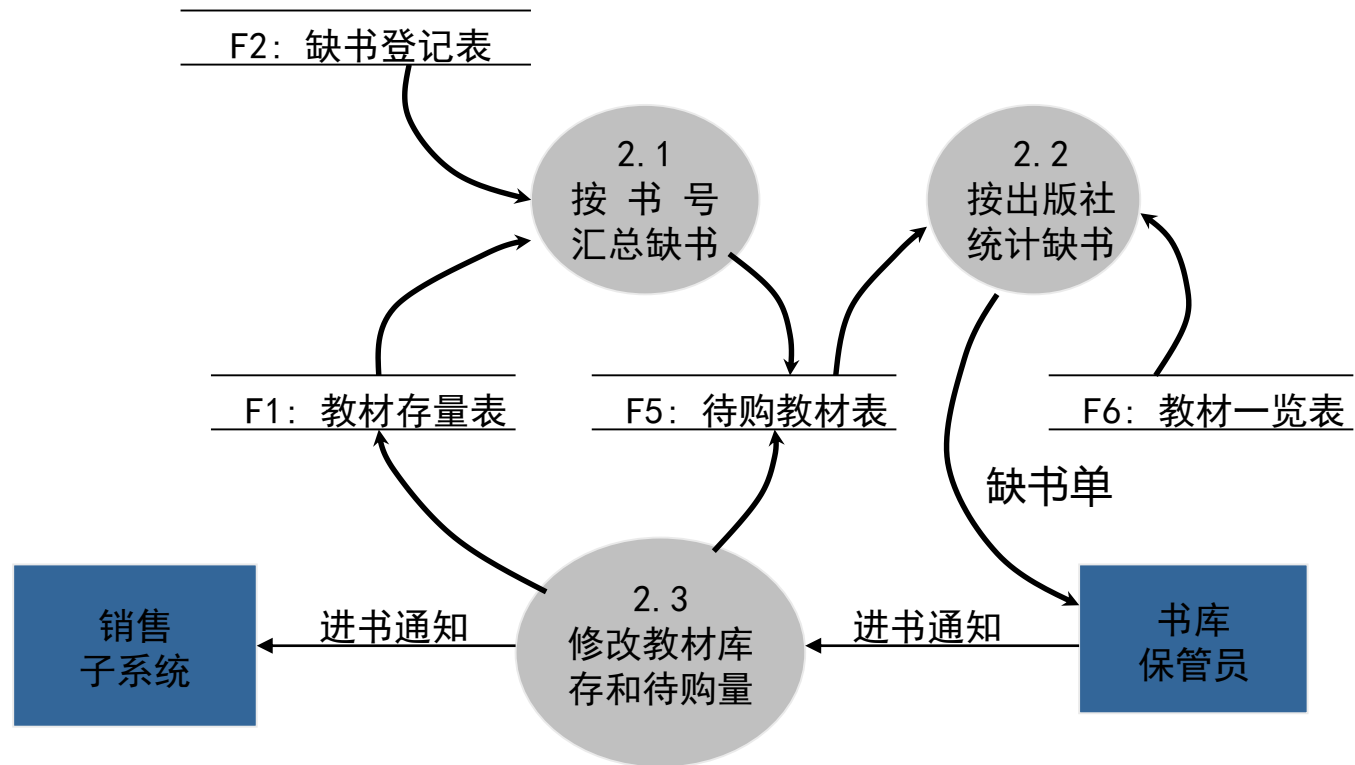


Example of DFD(3)

第1层DFD图—销售子系统



Example of DFD (4)



第1层DFD图—采购子系统

Data Dictionary

数据流图表示的模型中缺乏细节描述，即无法准确、完整地定义各个图元，因此需要数据字典（data dictionary, DD）来补充和完善。

数据字典主要由四类元素的定义组成：

数据流

数据项

数据存储

加工处理

Data Dictionary

1.数据流

领书单 = 学院+专业+班级+学号+姓名+{书号+[书名]+数量}+日期

发票= 学号+姓名+{书号+[书名]+单价+数量+总价}+书费合计

2.数据存储

教材存量表 = {书号+单价+数量}

暂缺书单 = 学号+姓名+ {书号+数量}

补售书单 = 学号+姓名+ {书号+数量}

教材科领书单					
学院:		专业:		班级:	
学号:		姓名:			
书 号	书 名	数 量	书 号	书 名	数 量
20 年 月 日					

发 票					
学号:			姓名:		
书号	书名	单价	数量	金额	备注

Data Dictionary

3. 加工处理

由输入数据、加工逻辑和输出数据等组成。加工逻辑阐明把输入数据转换为输出数据的策略。

输入：购书单

输出：有效购书单 或 无效购书单

加工逻辑：

1. 如果购书单中任一书号不在允许购书书目中，则为无效购书单；
2. 如果购书单中任一书号已经存在对应的班级购买记录，则为无效购书单；
3. 通过上述审核规则的购书单为有效购书单。

加工“审查有效性”的结构化说明

Control Flow Modeling

- Some applications are “driven” by **events** rather than data, produce **control information** rather than reports or displays, and process information with heavy **concern for time and performance**.
- 适合对**实时系统、控制类系统**等建模
- 以**State diagram**为主，描述系统对**外部事件**响应的行为模型

Example :State Diagram

