

结构化方法

内容

1. 结构化分析方法

- ✓ 数据流图及数据字典

2. 结构化设计方法

- ✓ 体系结构设计
- ✓ 详细设计

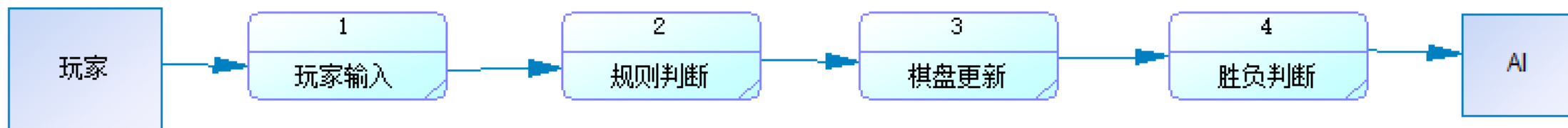


1.1 结构化分析方法

- 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**数据流图**的分析模型，通过分析数据流发现系统功能和行为，并对其进行划分。
- **结构化分析**：核心是通过对系统的功能分解，构建**功能模型**（即输入、处理和输出）。
- **面向对象分析**：核心是**对象模型**，强调系统中的对象及其协作。

1.2 数据流图

- 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**。
 - ✓ 例如人机对弈系统，采用结构化方法，构建DFD图如下

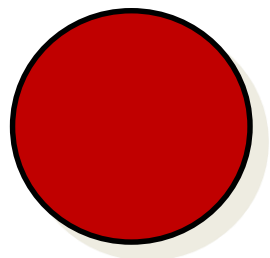


数据流图符号表示

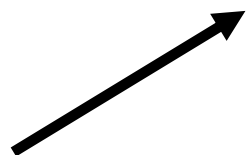


外部实体

- ✓ 系统之外的人员、组织或其它系统
- ✓ 表示数据流的源点或终点



加工或处理



数据流

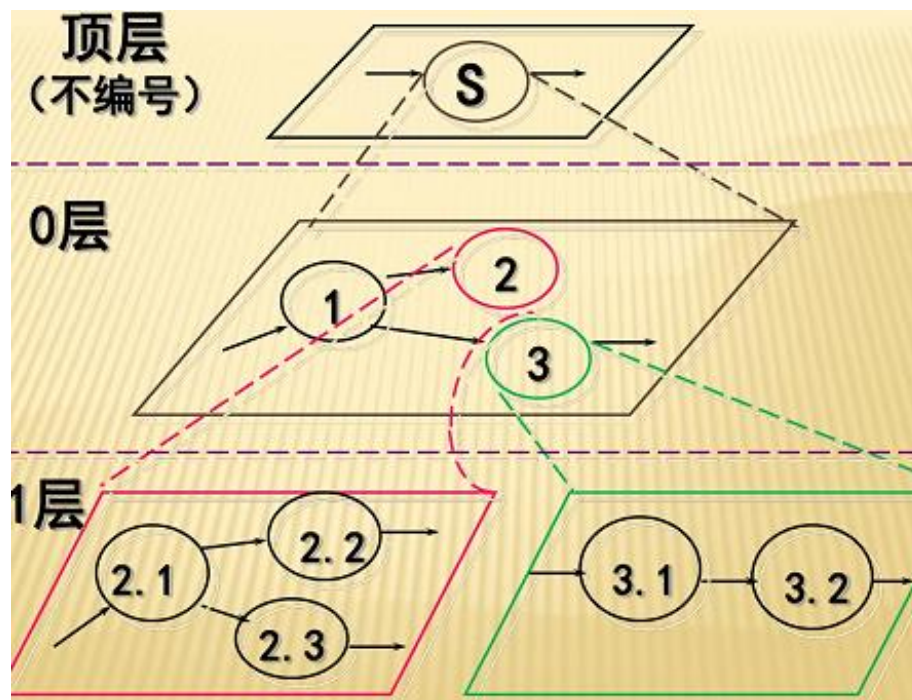
- ✓ 数据流可能存在于
外部实体与加工之间;
加工与加工之间;
加工与数据存储之间



数据存储

数据流图分析方法(1/3)

- 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。



数据流图分析方法(2/3)

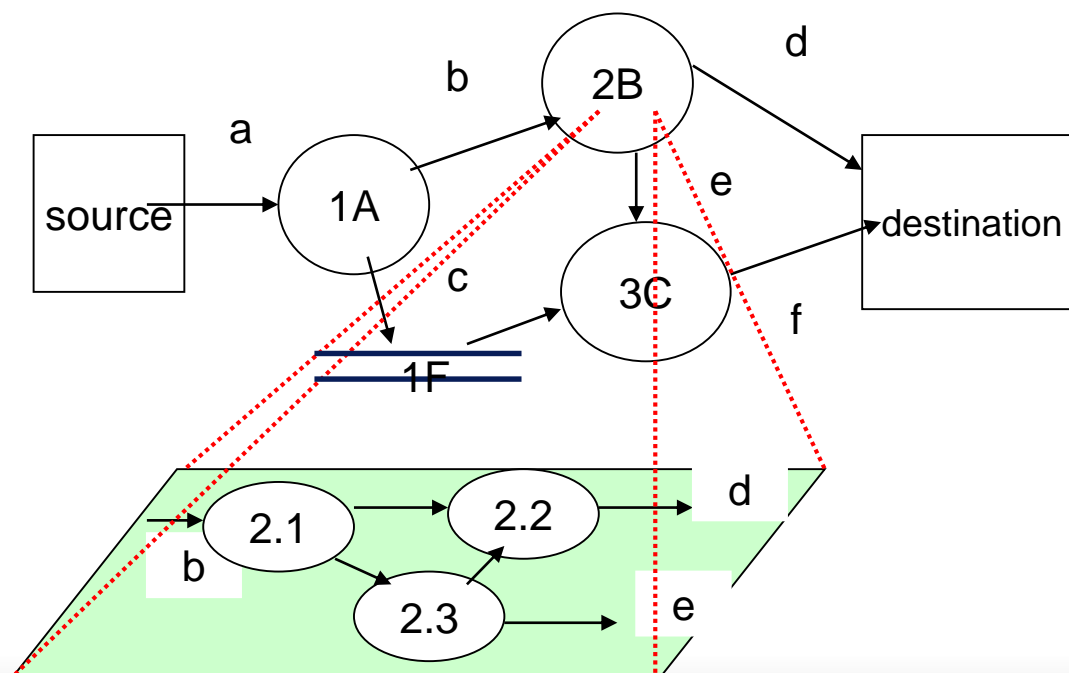
□建立顶层数据流图

- ✓顶层数据流图只含有一个加工，表示整个待开发系统。
- ✓作用：确定系统的外部实体，以及外部实体与系统之间的交互关系。

数据流图分析方法(3/3)

□对数据流图逐层分解

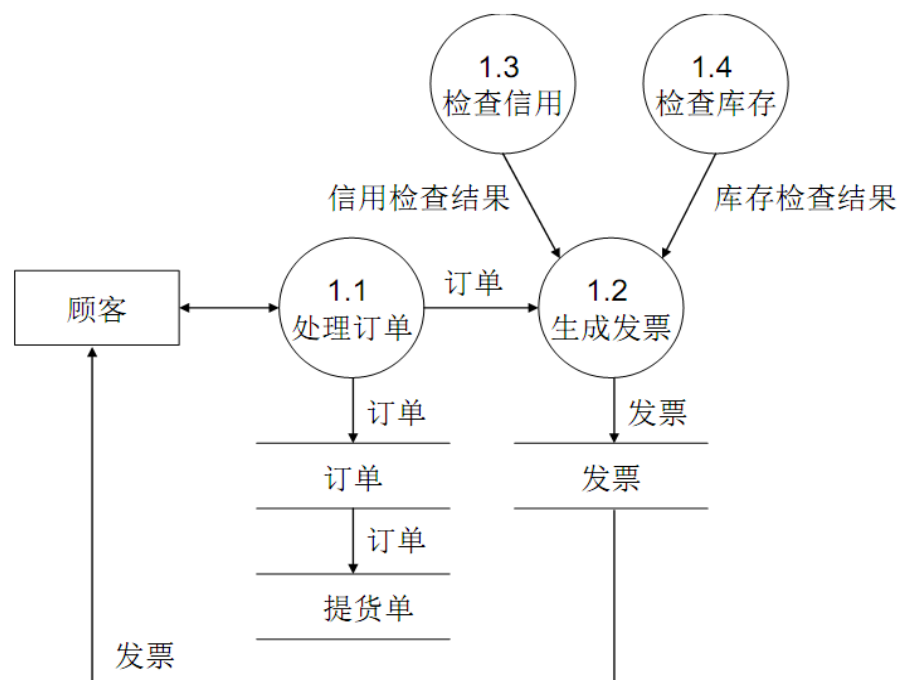
- ✓ 先将顶层DFD中的系统分解为若干个**子系统**，每个子系统表示为一个子加工。
- ✓ 继续在下层DFD中对子系统进行分解，直至每个加工**足够简单**。



数据流图建模约束 (1/2)

- 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

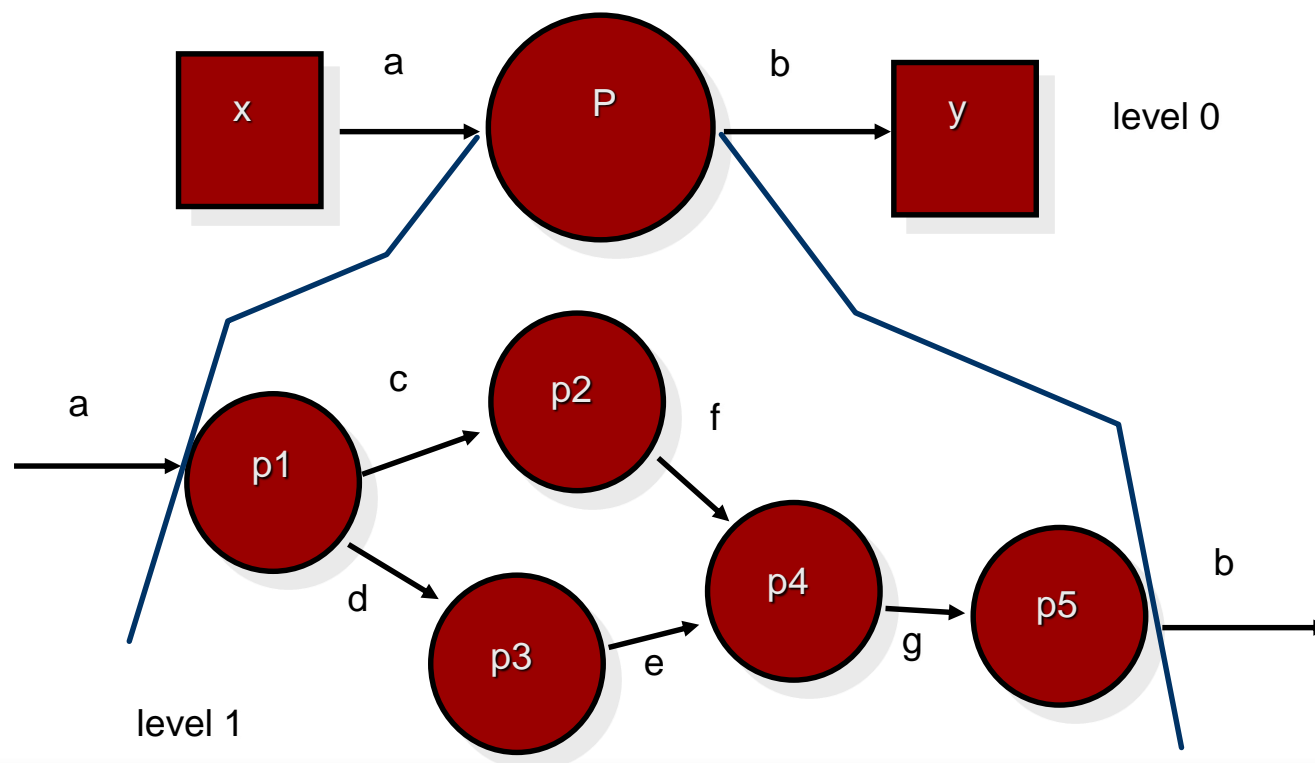
右边DFD图中有哪些错误？



数据流图建模约束 (2/2)

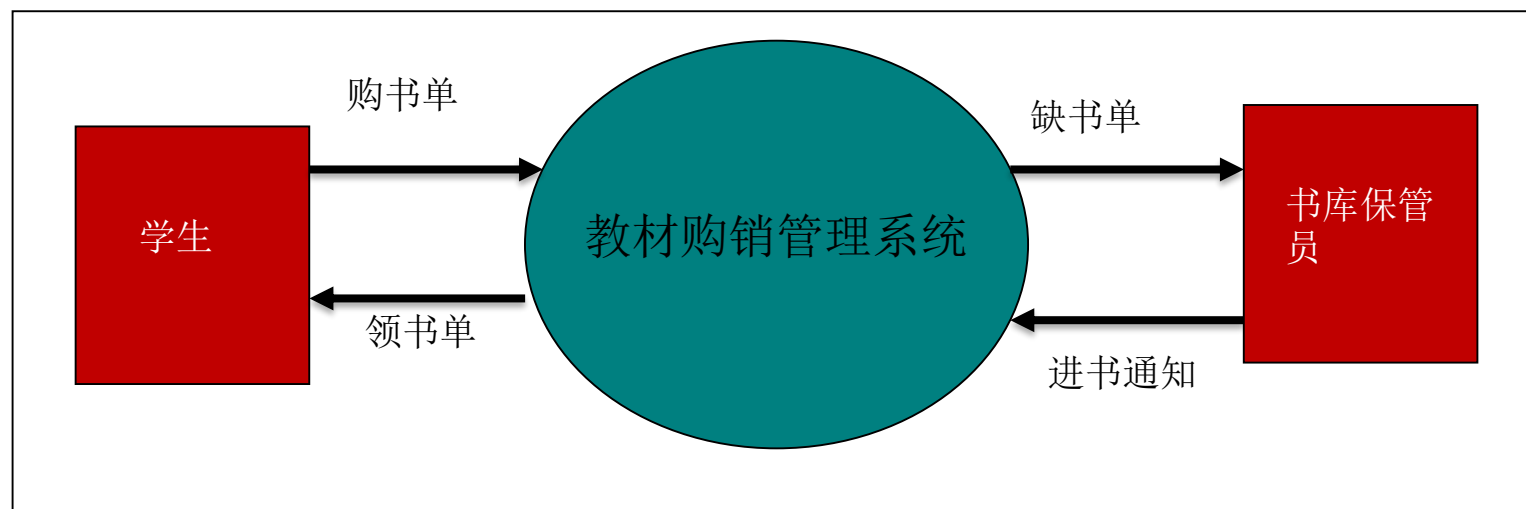
□保持父图与子图平衡

- ✓父图与子图的平衡是指：DFD子图必须与其父图对应加工的**输入**
/输出数据流保持一致



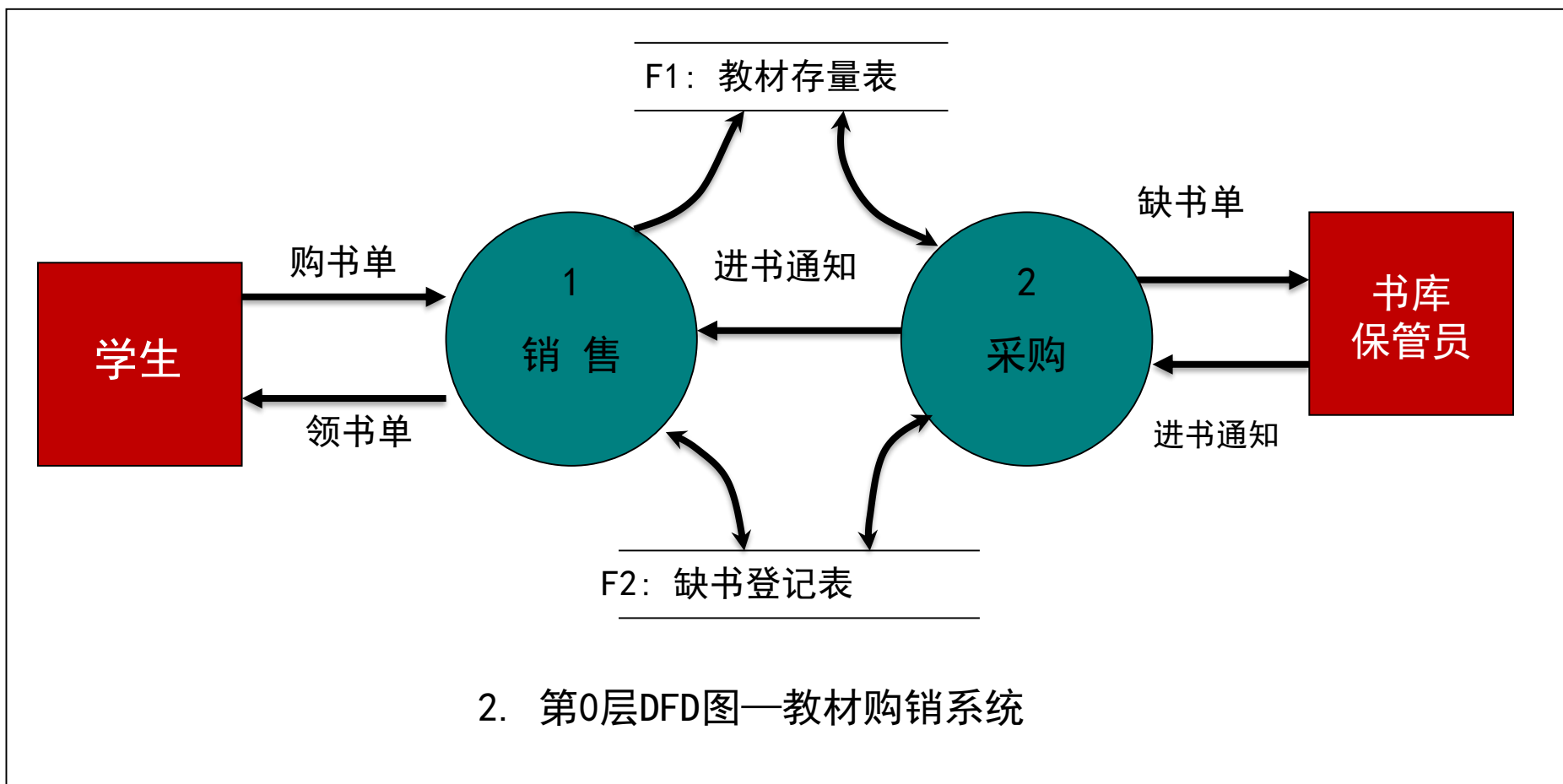
数据流图建模案例

- 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，则登记缺书信息，并发缺书单给书库保管员，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，进行教材补售。



数据流图建模案例

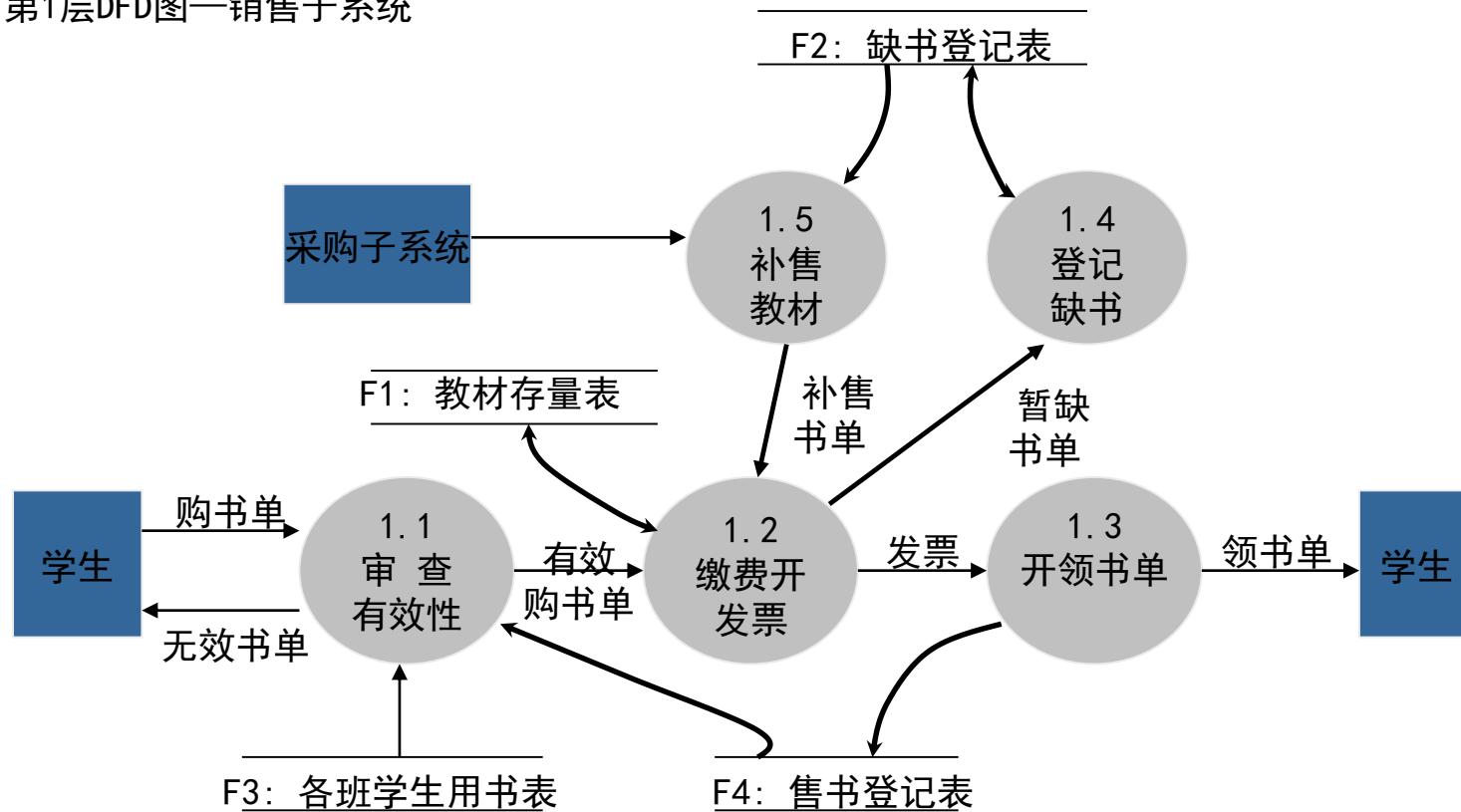
□分解细化



数据流图建模案例

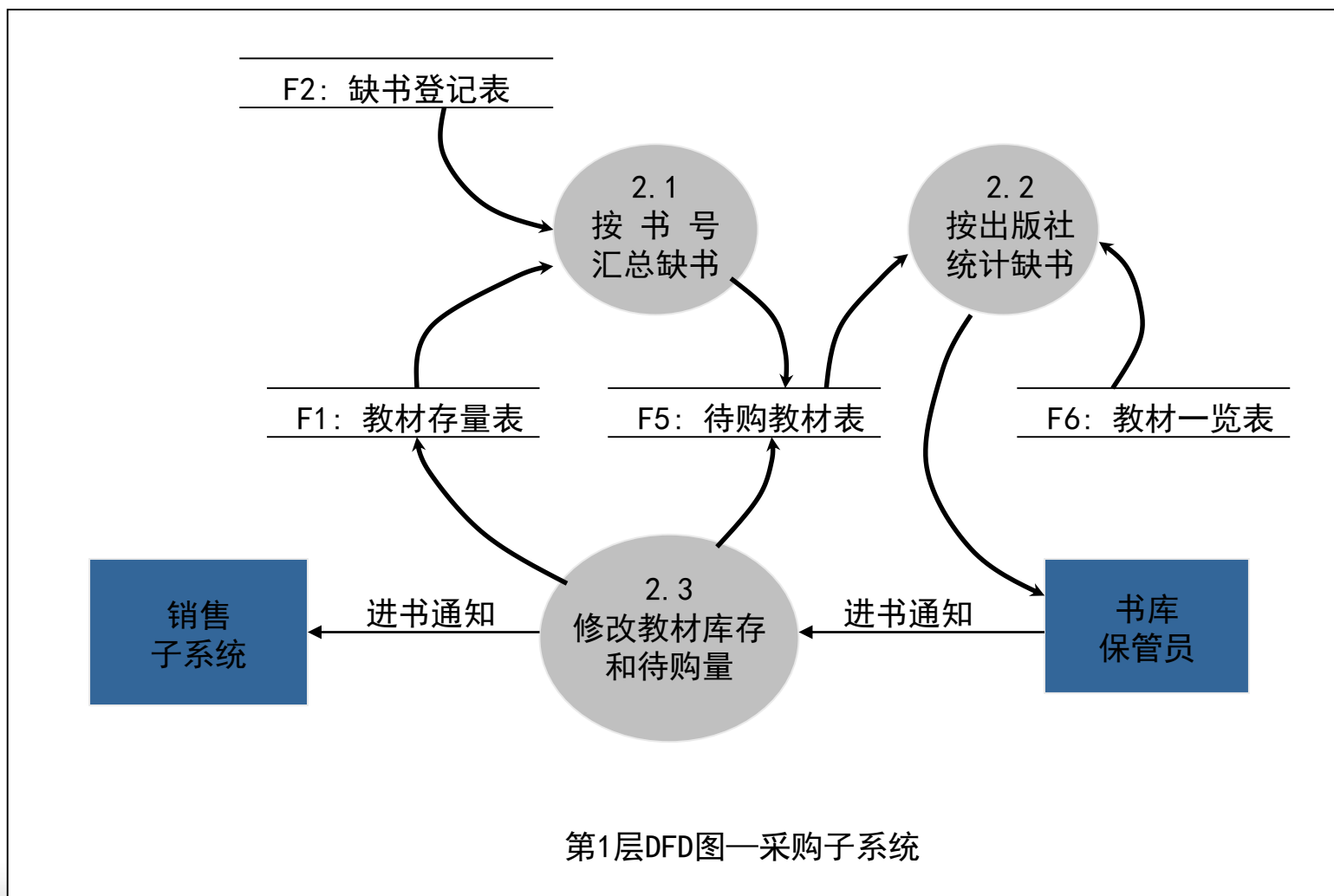
□分解销售子系统

第1层DFD图—销售子系统



数据流图建模案例

□分解采购子系统



1.3 数据字典

□ 为什么需要数据字典

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）对其补充和完善。

□ 数据字典主要构成

- ✓ 数据流
- ✓ 数据项
- ✓ 数据存储
- ✓ 加工处理

□ 1.数据流

发票= 学号+姓名+{书号+[书名]+单价+数量+总价}+书费合计

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}

[illegible]

数据字典实例

□ 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

内容

1. 结构化分析方法

- ✓ 数据流图及数据字典

2. 结构化设计方法

- ✓ 体系结构设计
- ✓ 详细设计



2.1 何为结构化设计

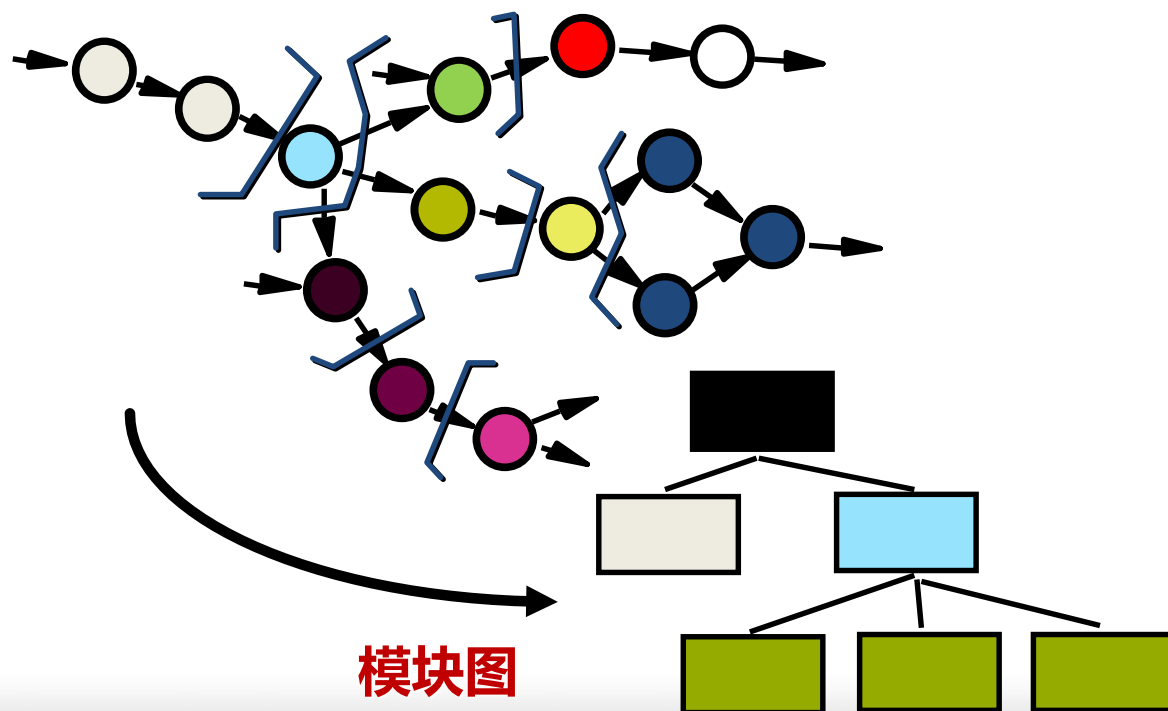
□目标是：基于结构化需求分析成果(DFD)，通过设计产生以**功能模块**为核心的软件设计模型。

□涉及两方面的工作

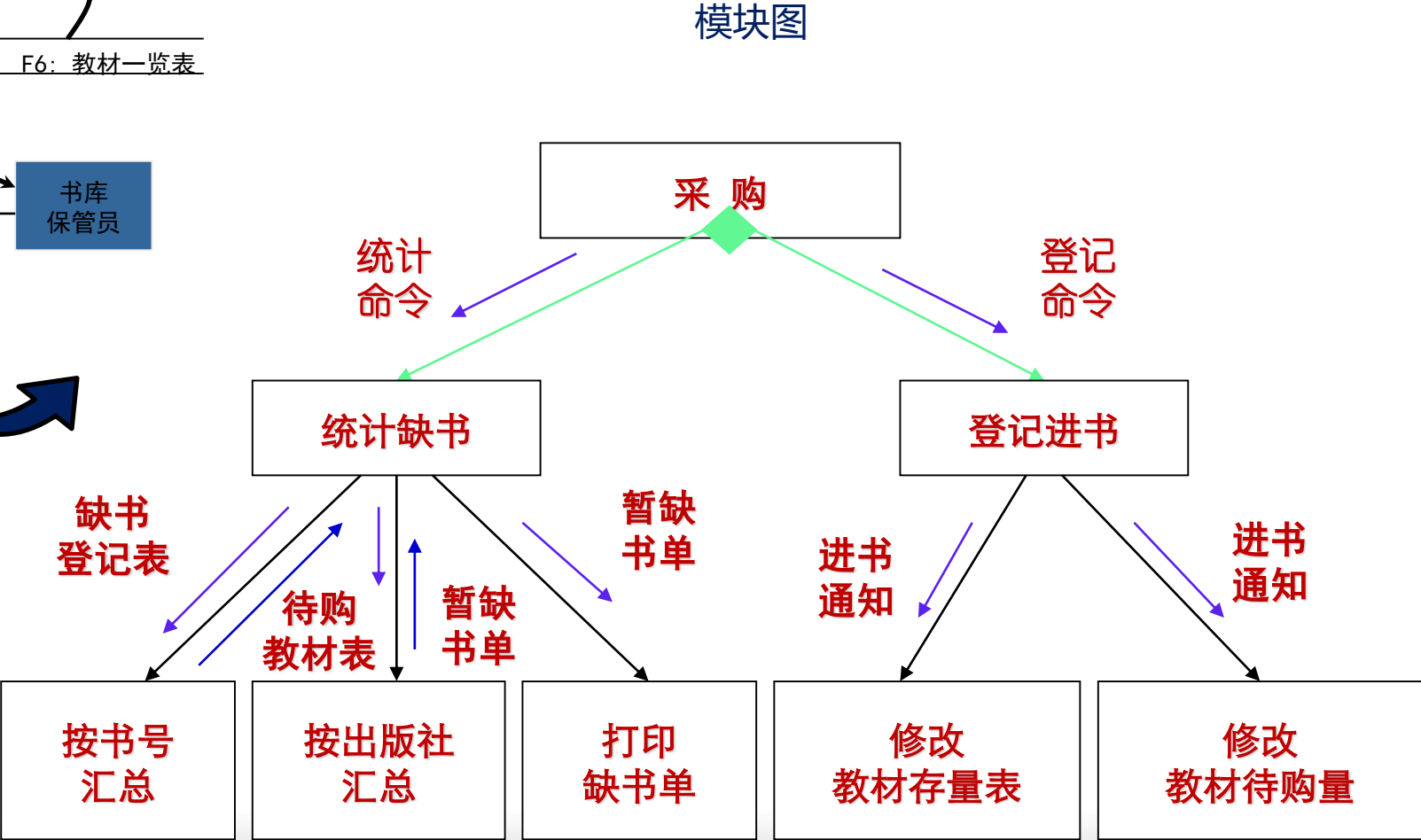
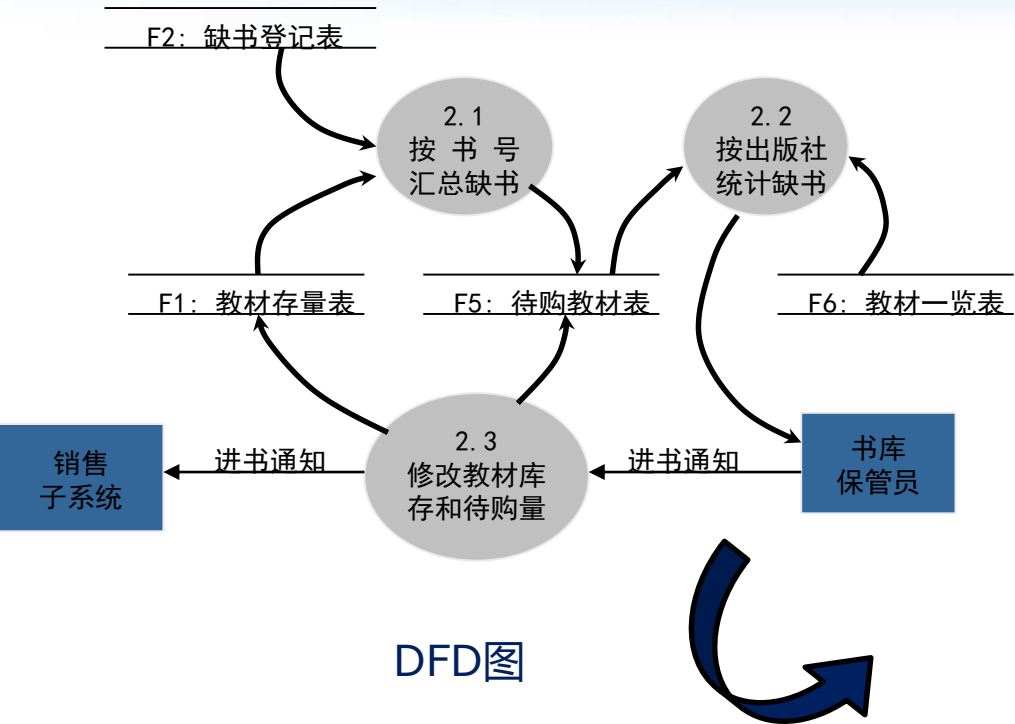
- ✓体系结构设计
- ✓详细设计

2.2. 体系结构设计

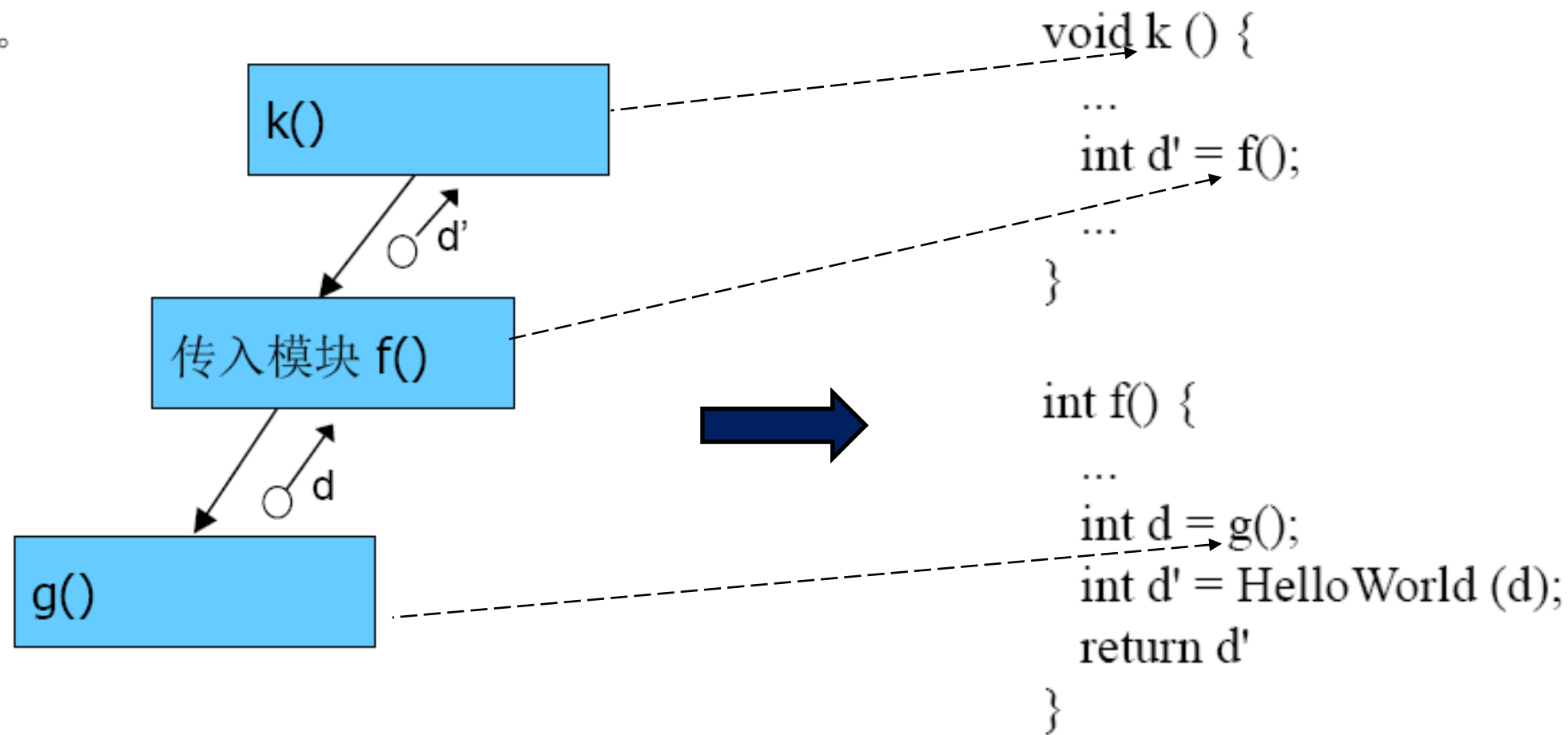
- 基于**数据流图**等需求模型，通过一系列设计变换，产生以**模块图**为核心的软件表示模型
- **模块图**：展示了系统模块间的层次结构和调用关系。矩形框表示模块，箭头表示模块之间的调用关系。



教材采购子系统体系结构设计实例



理解模块及调用关系



模块图

模块图的代码实现

结构化设计方法

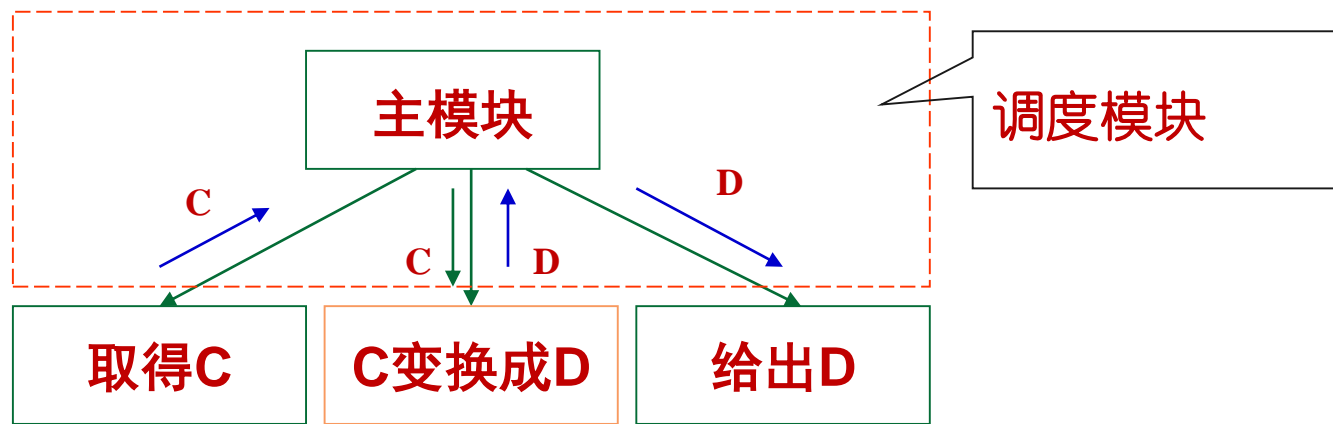
□结构化设计基本步骤

- ✓ (1) 分析确定数据流图类型：**变换型 或 事务型**。
- ✓ (2) 标识数据流图中输入、输出流边界。
- ✓ (3) 依据**变化或事务转换规则**，将DFD图映射为顶层和第一层模块结构。
- ✓ (4) 对第一层中的模块继续细化和分解。
- ✓ (5) 利用启发式规则优化初始系统结构图。

变换型数据流图及变换规则

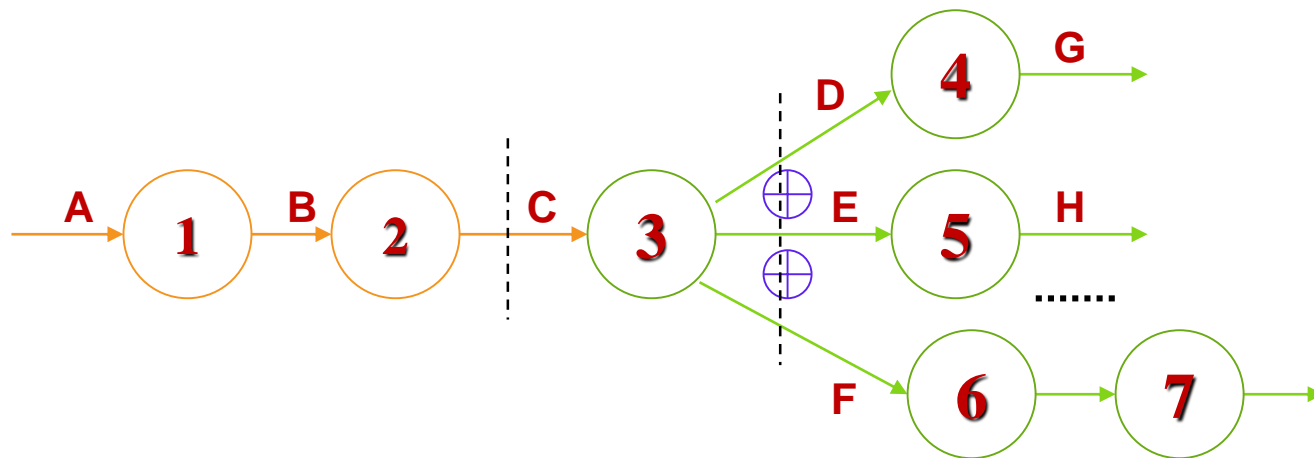


变换型数据流图

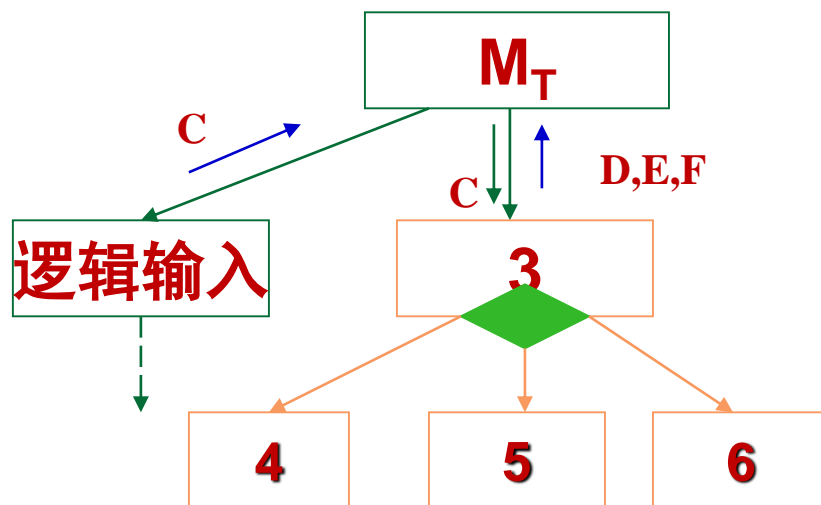


变换规则

事务型数据流图及变换规则



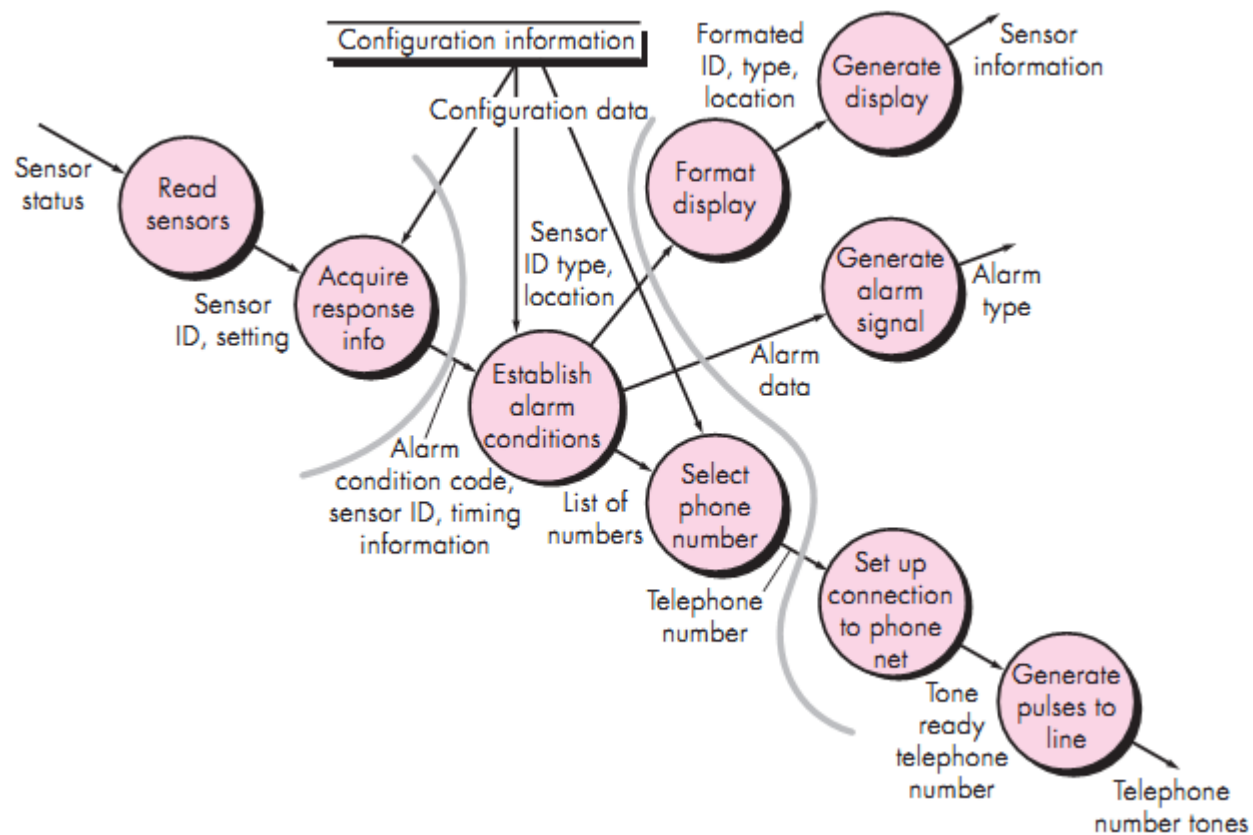
事务型数据流图



变换规则

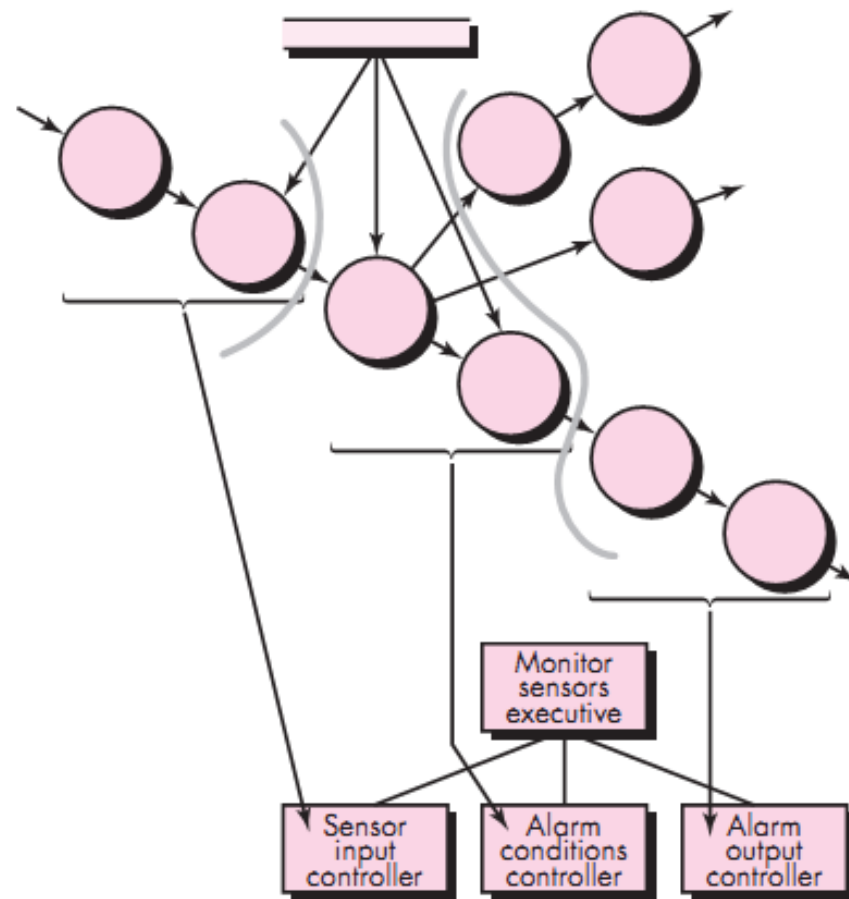
体系结构设计实例1

步骤1: 标识流边界



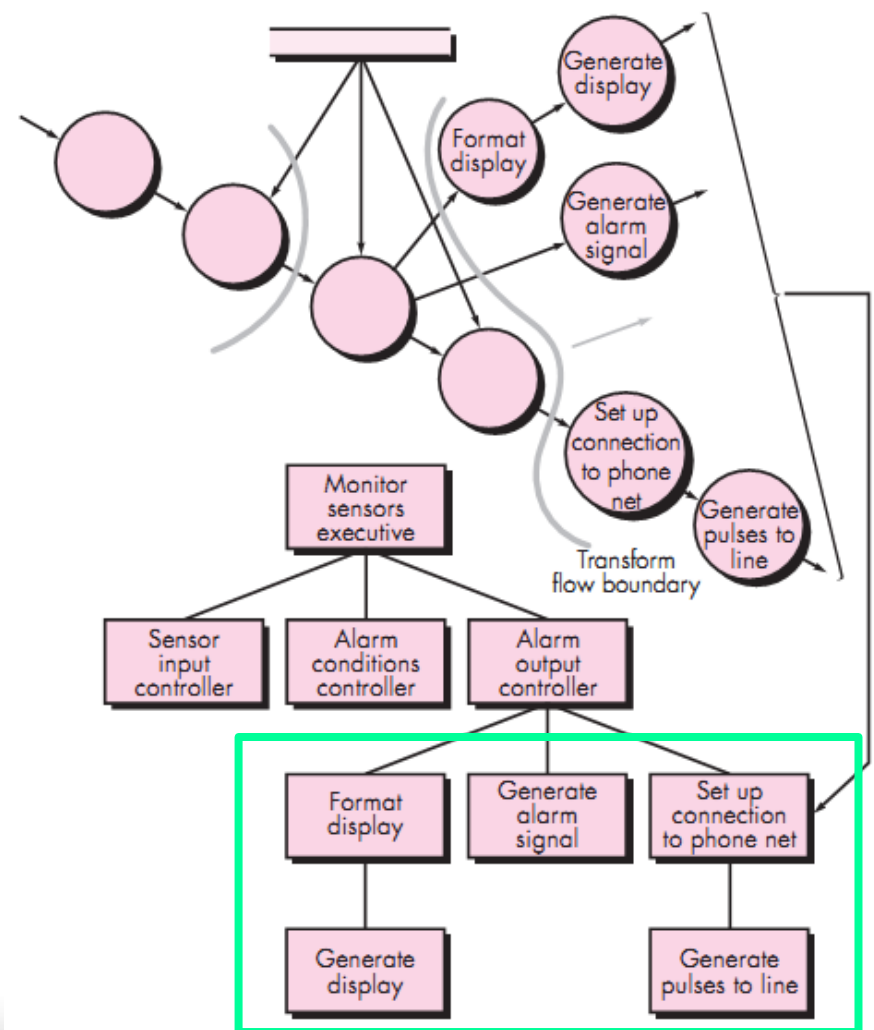
监控传感器DFD

步骤2: 第一级分解

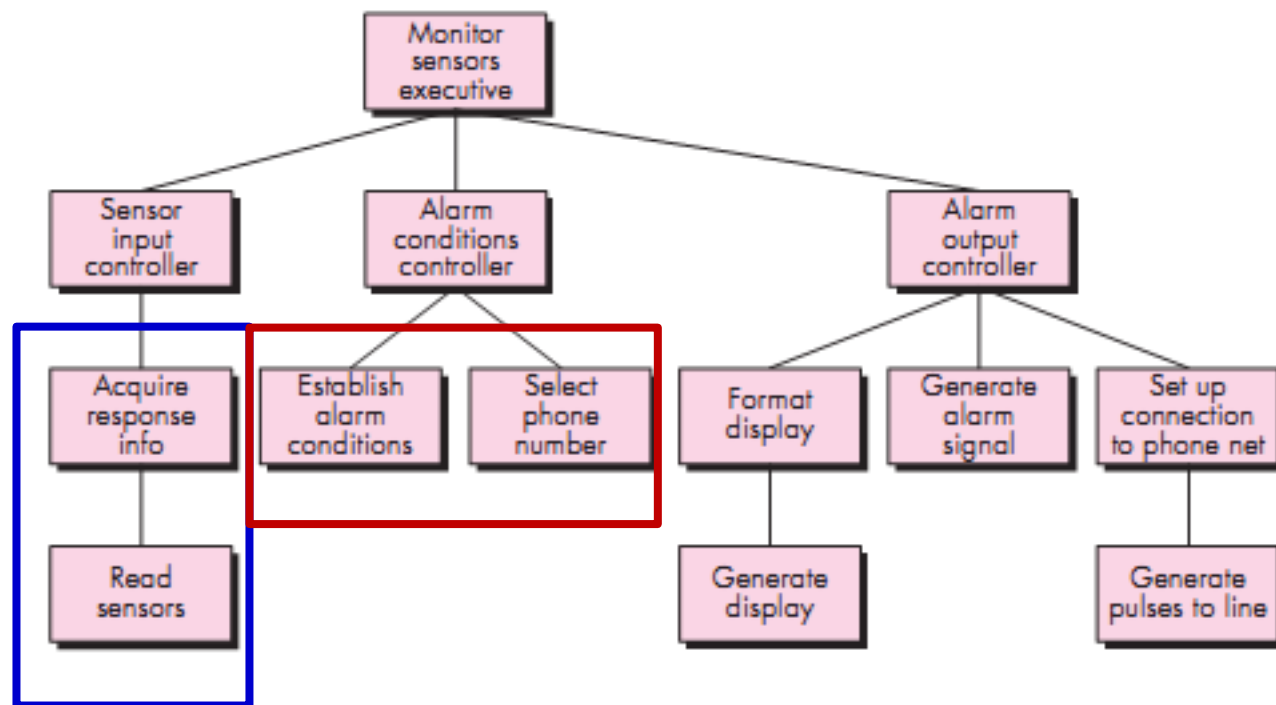


体系结构设计实例1

步骤3: 对输出模块进行细化

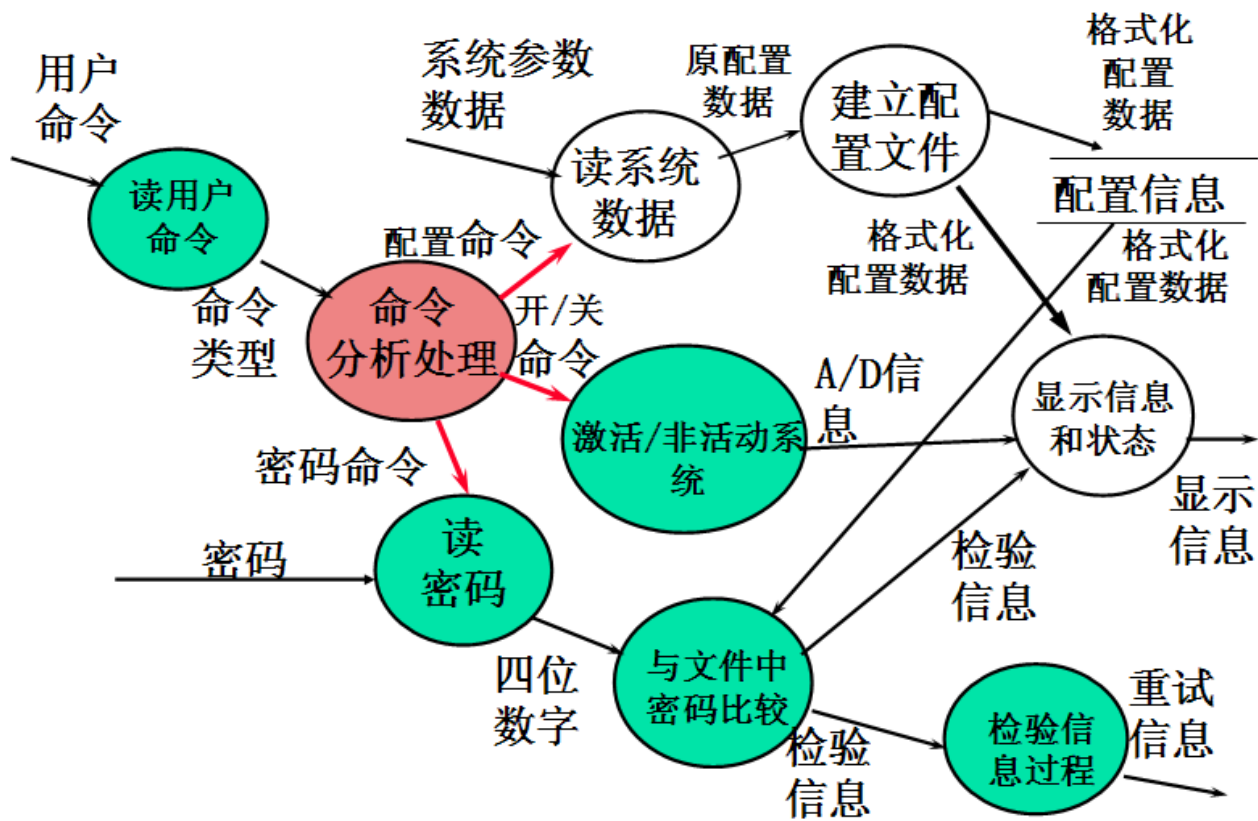


步骤4: 对输入和变换中心模块细化

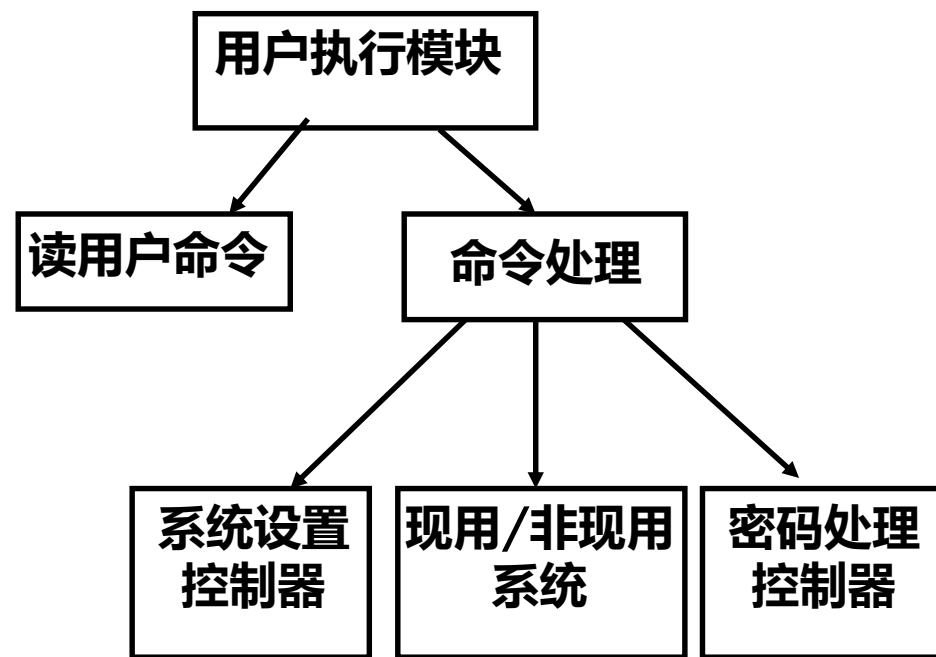


体系结构设计实例2

某工控系统DFD

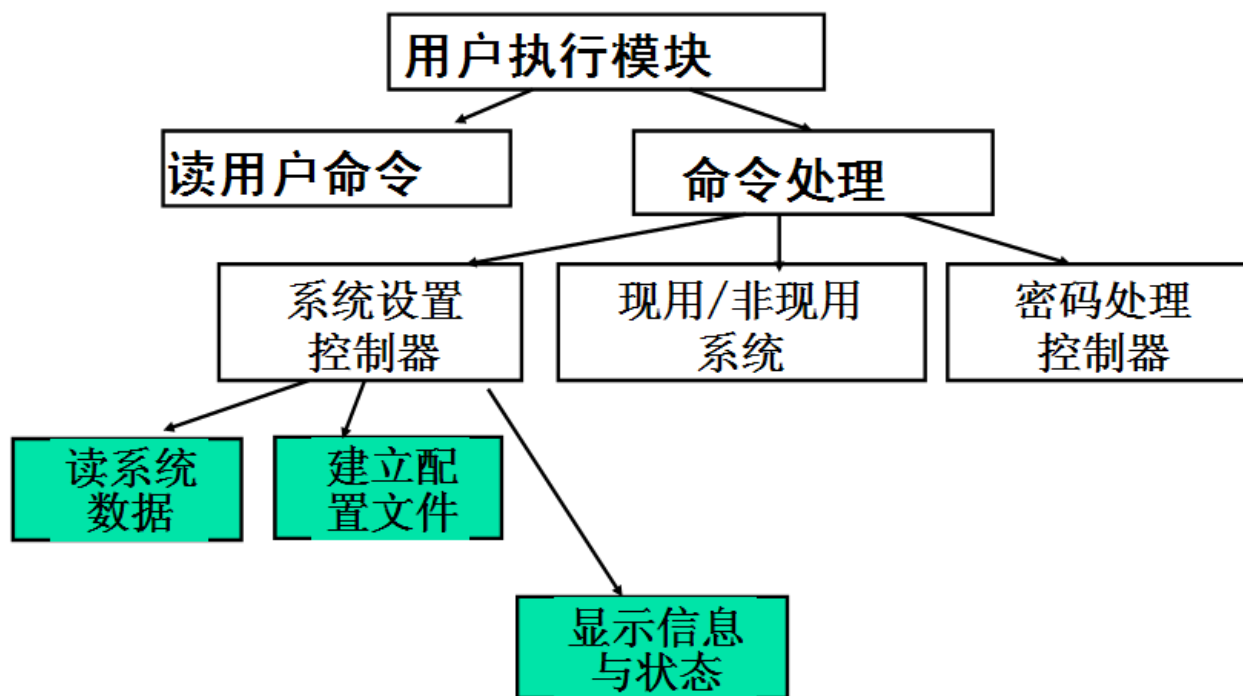


步骤1: 第一级分解



体系结构设计实例2

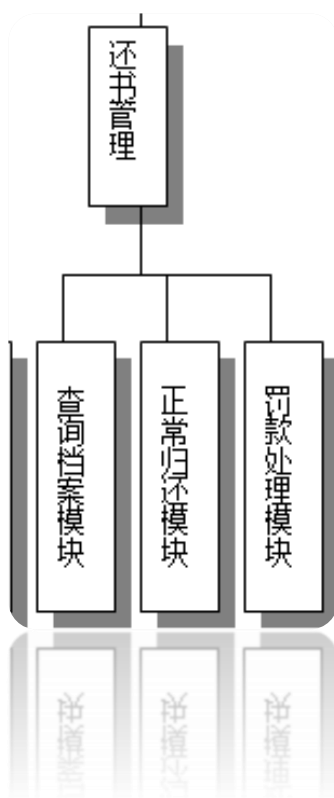
步骤2: 对输出模块进行细化



步骤3: 对输入模块进行细化

2.3 详细设计

□对每个模块的功能、接口、输入输出以及内部逻辑进行详细描述



接口：接收读者号、图书号，归还日期， 返回罚款金额

处理功能和算法描述：

输入

读者号

图书号

归还日期

罚款处理模块

通过读者号，图书号获取图书借阅日期

超期天数=归还日期-借阅日期

罚款金额=超期天数*0.10 元

INSERT INTO 罚款记录 (读者号, 读者名, 图书号,
借阅日期, 归还日期, 罚款金额, 操作员)

Return 罚款金额

问题和讨论

