结构化设计

结构化分析与设计

■ 提出

➢20世纪70年代中期，Stevens、Myers与Constantine等人在结构化程序设计的基础上，倡导结构化设计(Structured Design, SD)技术。

➢70年代后期，Yourdon等人倡导了与SD配套的结构化分析(Structured Analysis, SA)技术，合称为结构化分析与设计方法。

➢结构分析与设计方法是瀑布模型的首次实践。

设计在先,分析在后

结构化分析与设计

■结构化设计任务

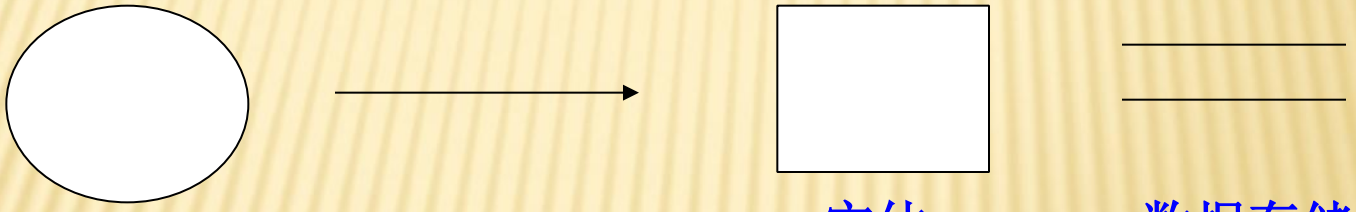
➢软件设计=总体设计+详细设计。总体设计的任务是把分析模型中的数据流图(DFD图)转换为软件结构图(SC图)，详细设计阶段的任务是对SC中的各个模块进行算法和数据结构设计。

➢软件设计指导思想：分解和细化是软件设计阶段的指导思想。

结构化分析与设计

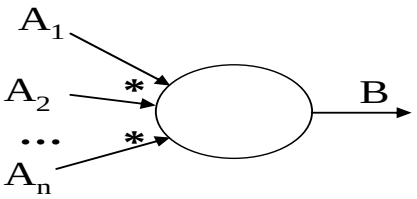
■数据流图(DFD)

➢**数据流图：**数据流图是用来刻画数据流和加工的信息系统建模技术。

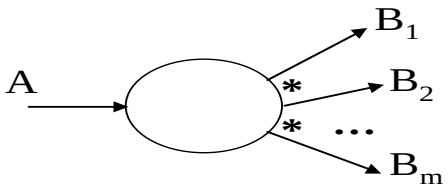


DFD图的基本符号

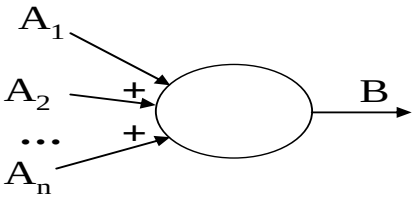
结构化分析与设计



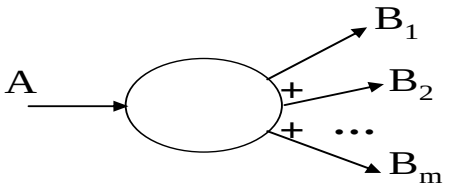
数据$A_1,A_2,\ldots A_n$同时输入才能变换成数据B.



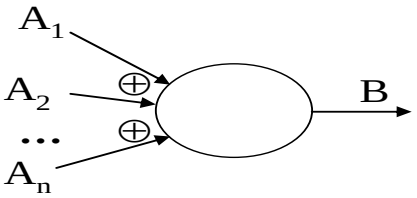
数据A变换成数据$B_1,B_2,\ldots,B_m$.



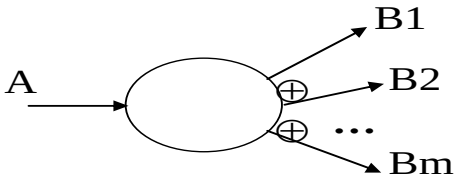
数据$A_1,A_2,\ldots,A_n$中至少有一个输入就能变换成数据B.



数据A变换成数据$B_1,B_2,\ldots,B_m$中的任意组合



数据$A_1,A_2,\ldots,A_n$中有且仅有一个输入就能变换成数据B.



数据A变换成数据$B_1,B_2,\ldots,B_m$中的一个，但不能同时有两个或两个以上

DFD图的扩展符号

结构化分析与设计

➢结构化设计

✓**体系结构设计：**定义软件主要组成部件之间的关系；

✓**接口设计：**描述软件内部、软件和接口、系统之间以及软件与人之间是如何通信的；

✓**过程设计：**将软件体系结构的组成部分转变成对软件组件的过程性描述。 ？

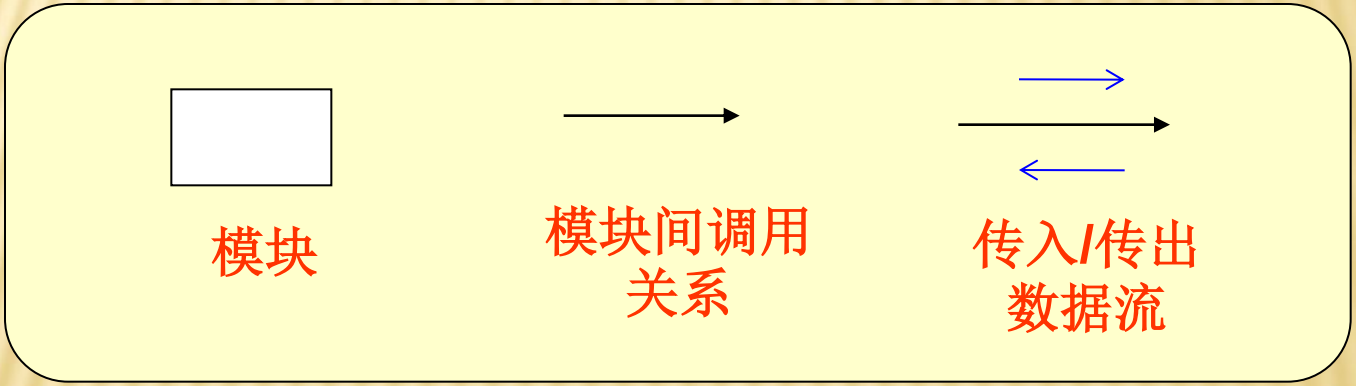
✓**数据设计：**将分析阶段创建的信息模型转变成软件实现所需的数据结构；

结构化分析与设计

■结构图(Structure Chart, SC)

➢**结构图：**软件结构概要设计阶段的工具，反映系统的功能实现以及模块与模块之间的联系与通信，即反映了系统的总体结构。

➢**结构图的表示方法：**用矩形表示模块，用带箭头的连线表示模块间的调用关系。在调用关系的两旁，标出传入和传出模块的数据流。

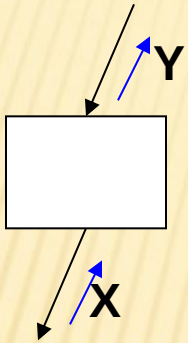


SC图的符号

结构化分析与设计

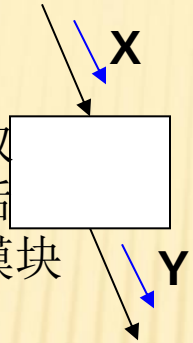
➢结构图的6种模块：传入、传出、变换、源、漏和控制。

从下属模块取得数据，经过处理，再将其传送给上级模块

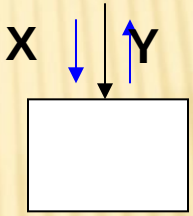


(a)传入

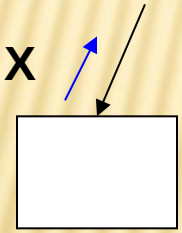
从上级模块取得数据处理后传送给下属模块



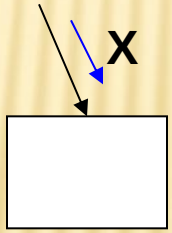
(b)传出



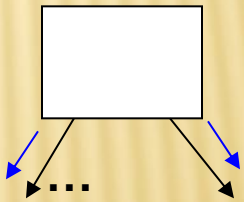
(c)变换



(d)源（作为数据输入模块）



(e)漏（作为数据输出模块）



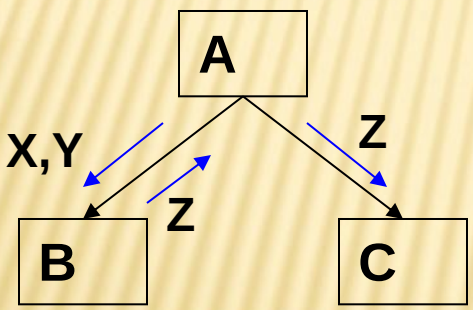
(f)控制

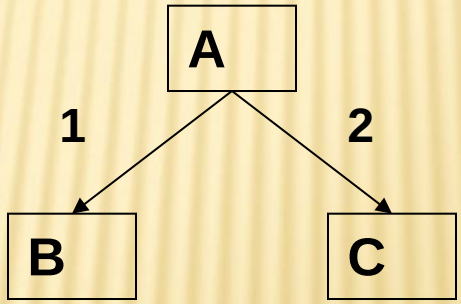
结构化分析与设计

➢SC图中的模块调用

✓**简单调用：** 调用线的箭头指向被调用模块。

例如，在图(a)中，允许模块A调用模块B和C，反之则不可以。调用B时，A向它传送数据流X和Y，B向A返回数据流Z。调用C时，A向C传送数据流Z。





1

2

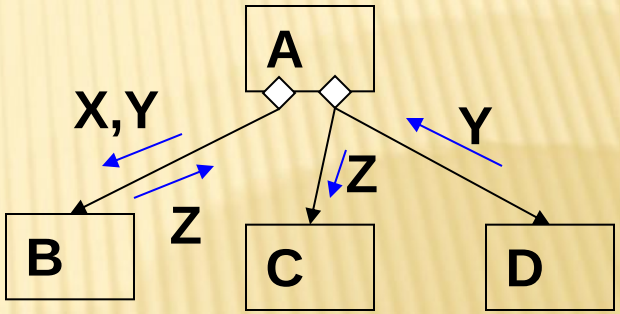
入	出
X,Y	Z
Z	-

(a)简单调用两种表示方法

结构化分析与设计

✓**选择调用：** 用菱形符号表示选择。

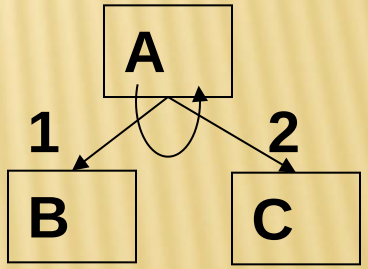
例如，图(b)的左侧的菱形表示：模块A根据它内部的判断，来决定要不要调用模块B；右侧的菱形表示：模块A按照另一判断结果，选择调用模块C或模块D。



(b)选择调用表示方法

✓**循环调用：** 用叠加在调用线始端的环形箭头表示循环。

✓例如，下图含义是：模块A将根据其内在的循环重复调用B，C等模块，直到在A模块内部出现满足循环终止的条件为止。



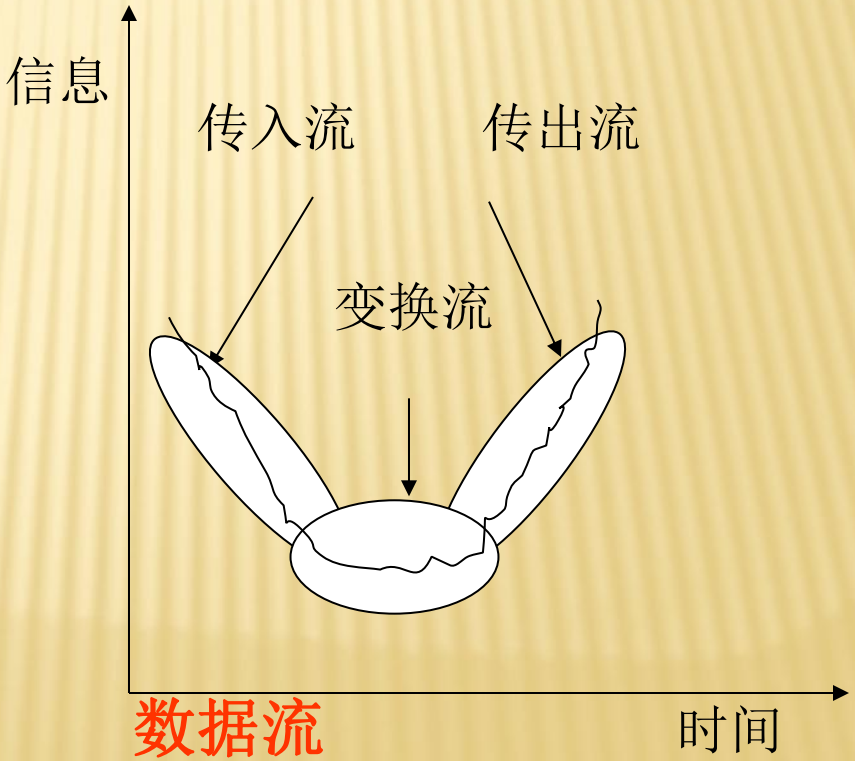
(c)循环调用表示方法

结构化分析与设计

➢数据流图的类型

✓**变换型结构：**这类结构由3部分组成：传入路径(Afferent Path)、变换中心(Transform Center)和传出路径(Efferent Path)。流经这3部分的数据流分别称为传入流、变换流和传出流。

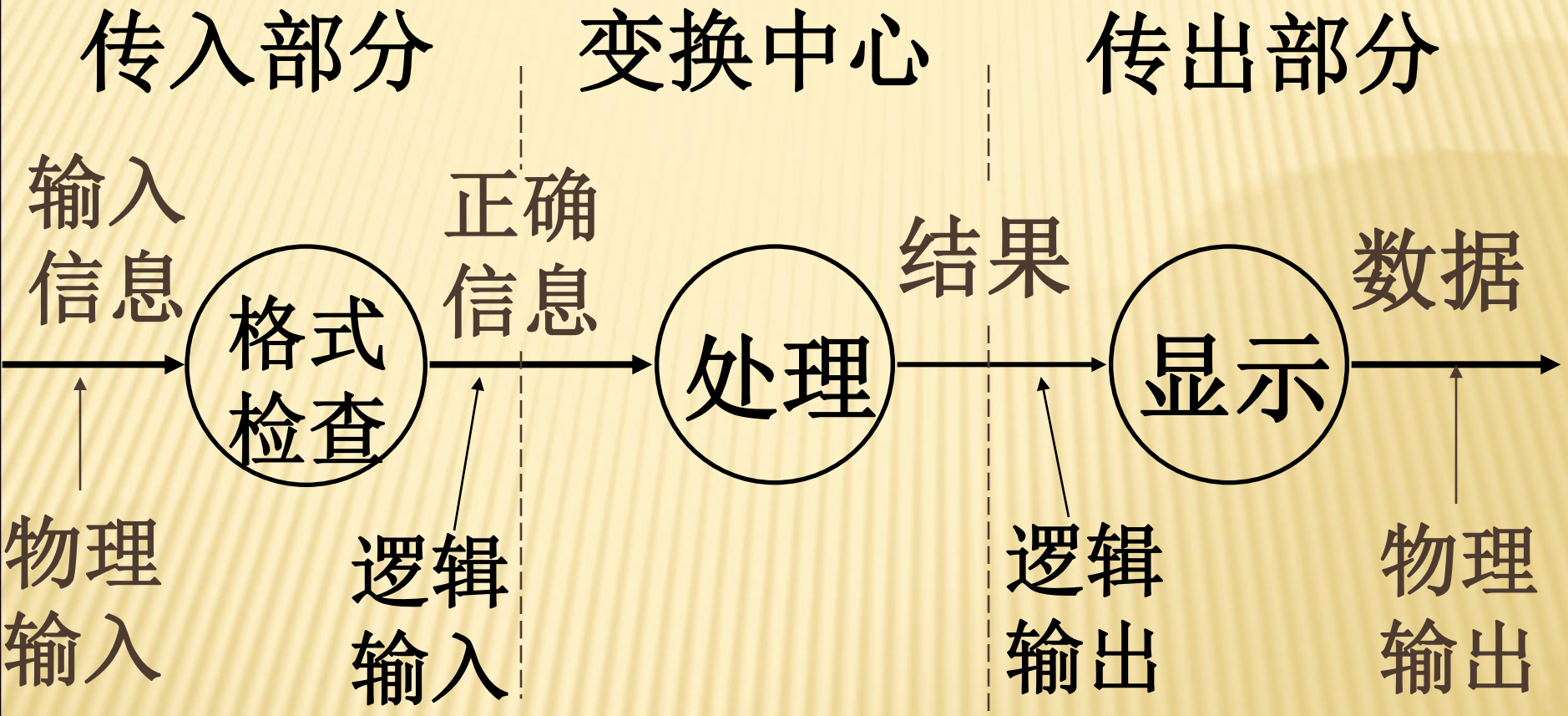
基本模型



数据流

变换型数据流举例(CPP)

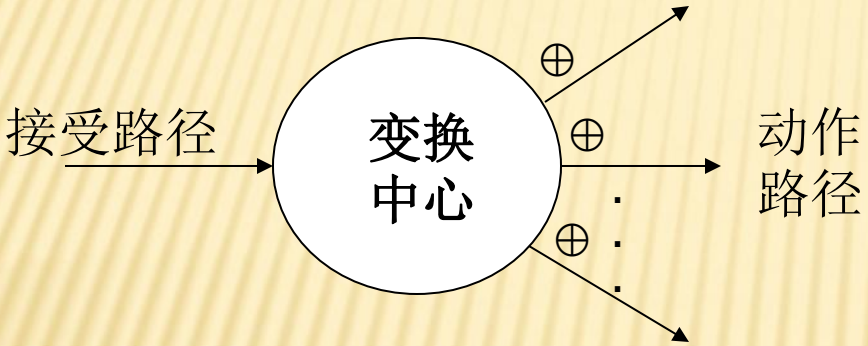
注意:一般的DFD底层图都可以概括成传入、变换、传出三部分



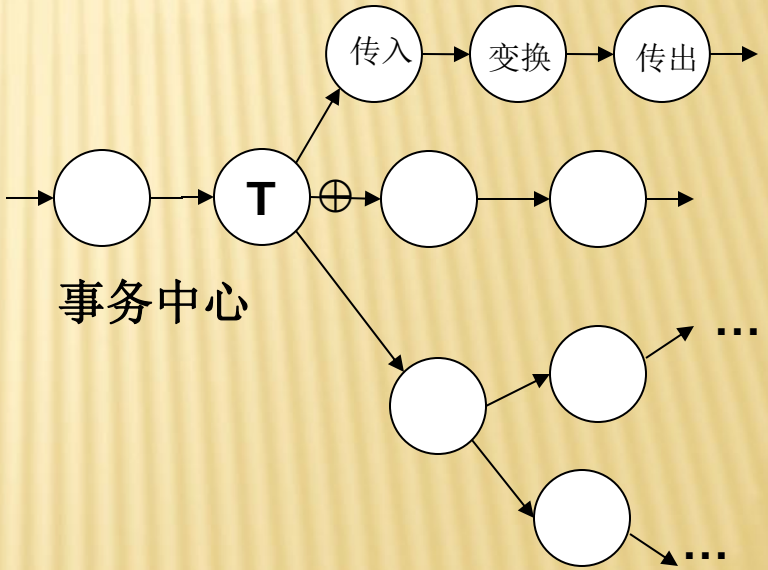
特点：具有明确的传入、变换(或称主加工）和传出界面的DFD

结构化分析与设计

✓**事务型结构：**这类结构由至少一条接受路径(Reception Path)、一个事务中心(Transaction Center)与若干条可选动作路径(Action Path)组成。



事务型结构的系统基本模型



同时存在两种结构的系统

结构化分析与设计

➢体系结构设计方法的步骤

✓复审DFD图，必要时可再次进行修改或细化；

✓鉴别DFD图所表示的软件系统的结构特征，确定它所代表的软件结构是属于变换型还是事务型；

✓按照SD方法规定一组规则，把DFD图转换为初始的SC图；

✓按照优化设计的指导原则改进初始的SC图，获得最终SC图。

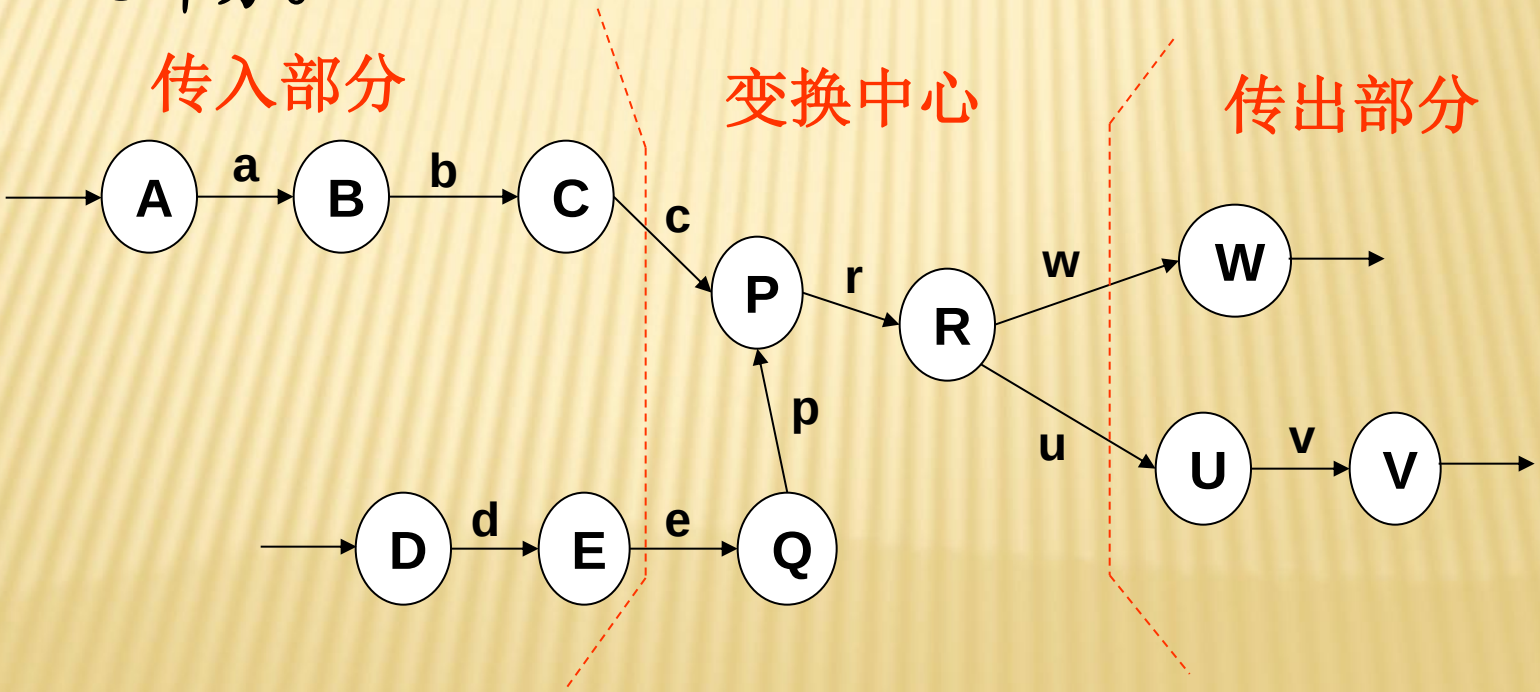
➢DFD向初始SC映射方法

结构化分析与设计

➢变换映射

✓步骤1：区分传入、传出和变换中心3个部分，划分DFD图边界。

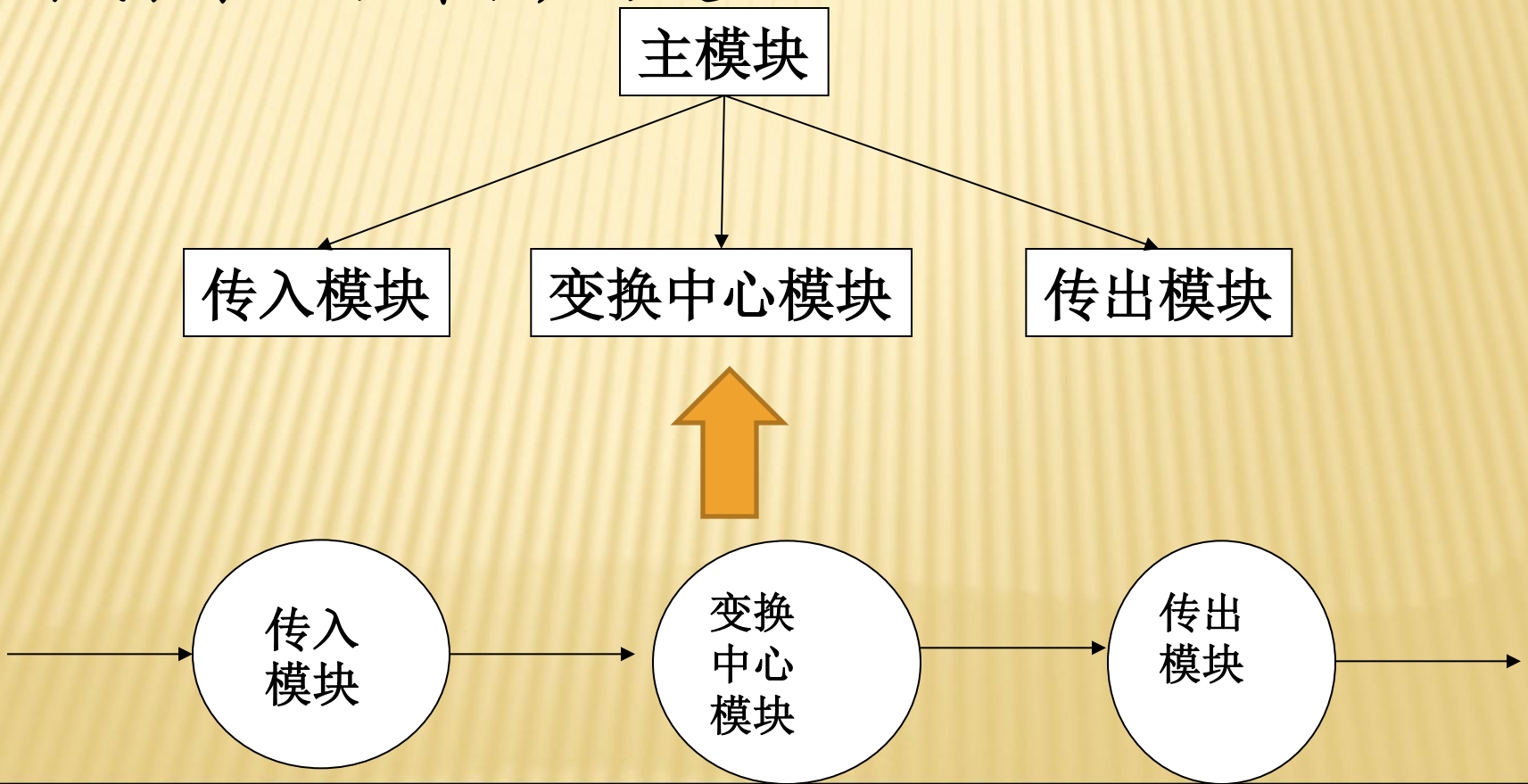
例如，对下面DFD图进行划分，区分传入、传出和变换中心3部分。



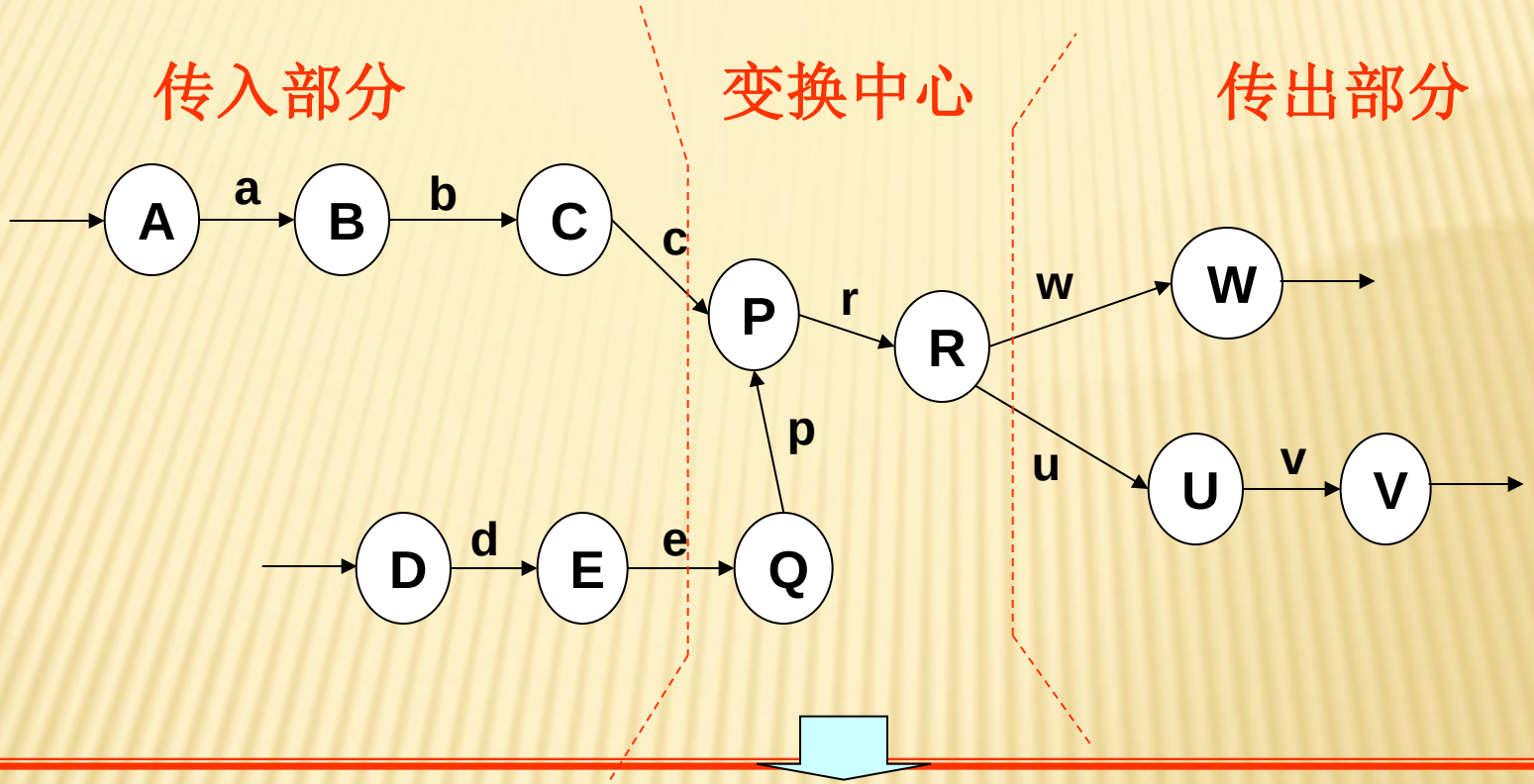
结构化分析与设计

✓步骤2：完成“第一级分解”，建立初始SC图的框架。

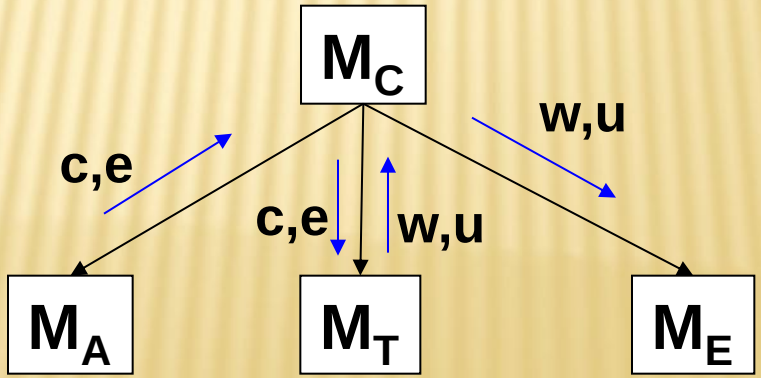
初始SC图的框架通常包括最上面的两层模块：顶层和第一层。任何系统的顶层只含有一个用于控制的主模块。它的下一层一般包括传入、传出和中心变换3个模块，分别代表系统的3个相应分支。



结构化分析与设计



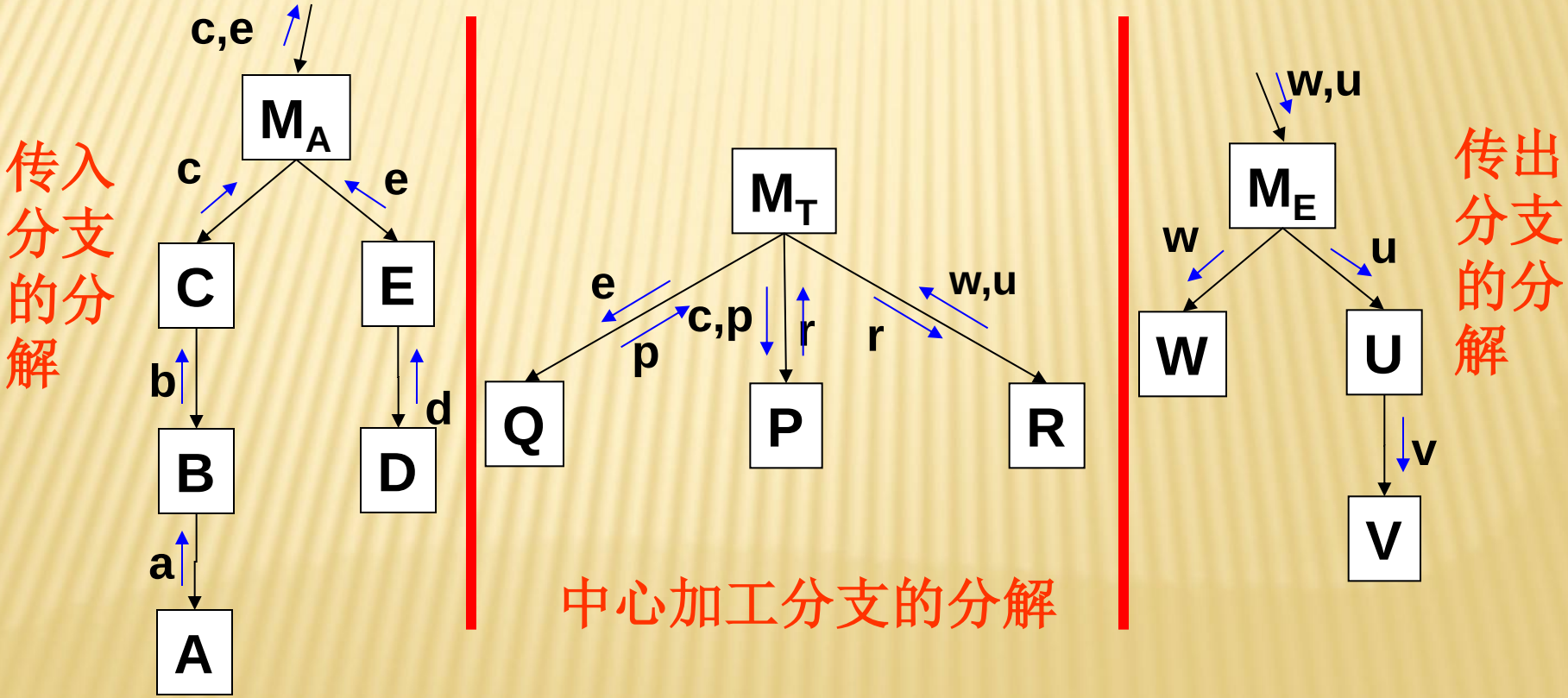
第一级分解后的SC图



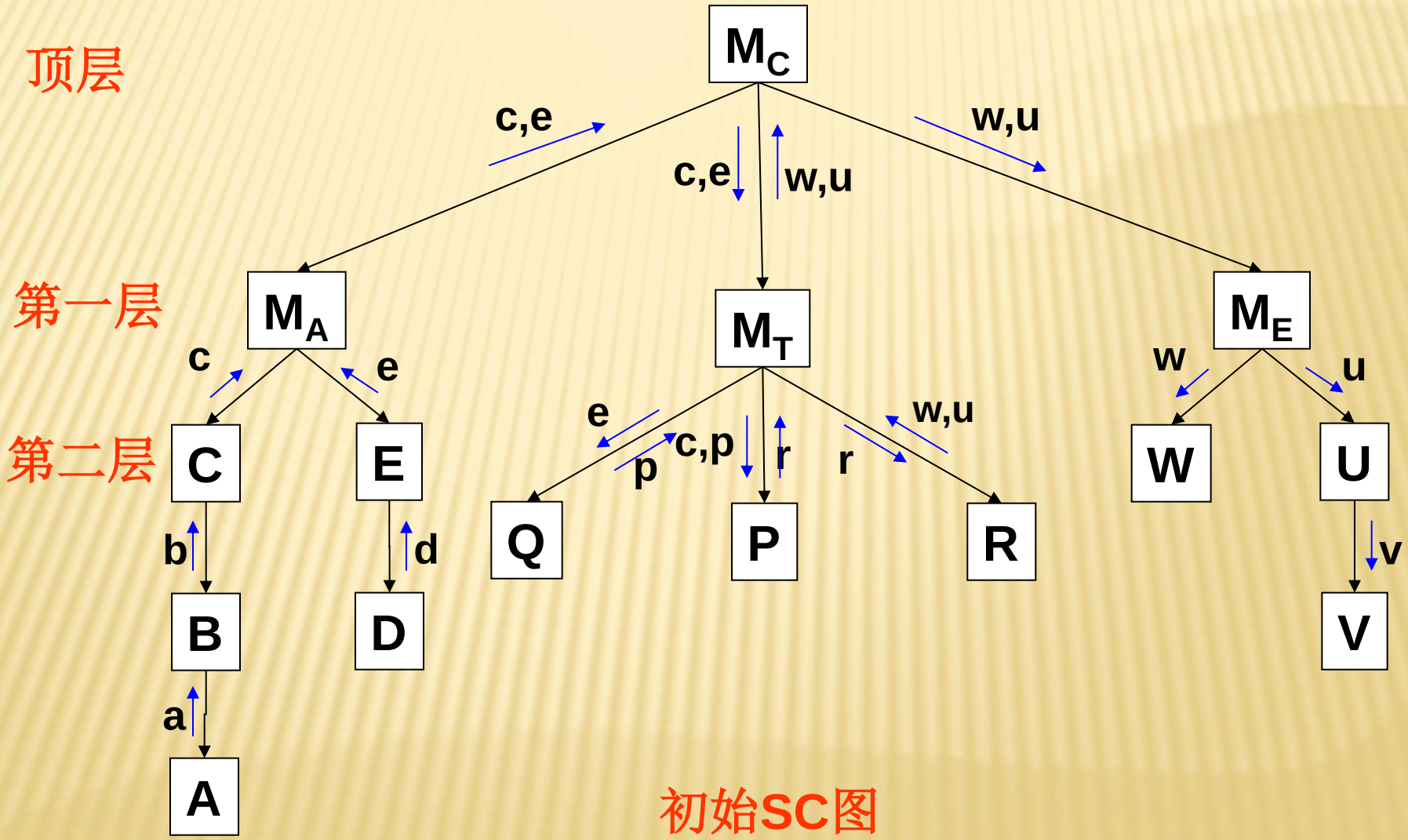
结构化分析与设计

✓步骤3：完成“第二级分解”，细化SC图的各个分支。

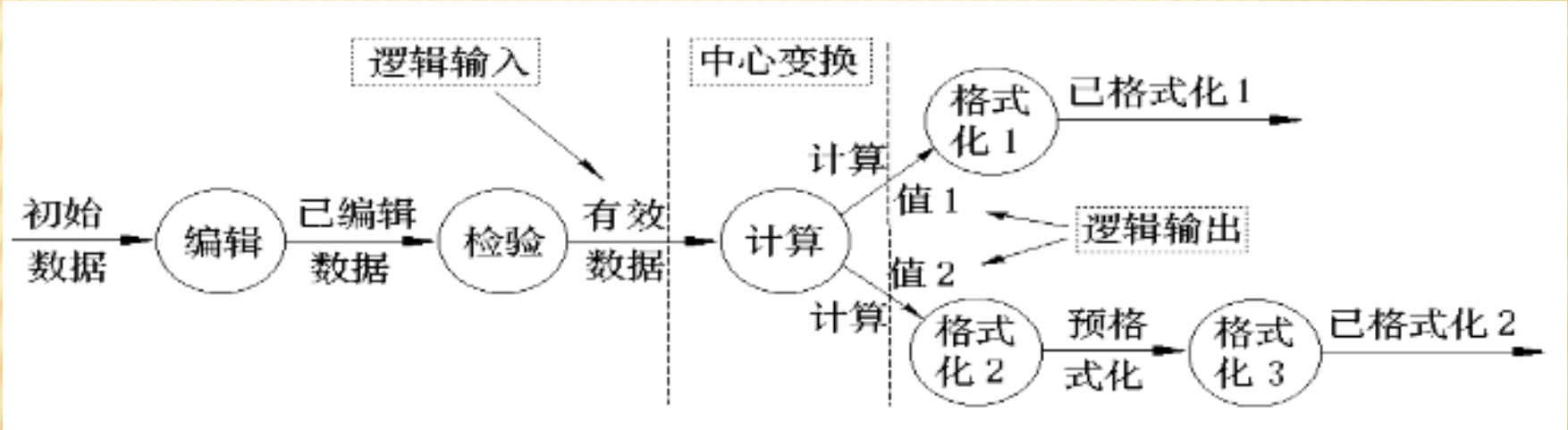
对上步的结果继续进行由顶向下的分解，直至画出每个分支所需要的全部模块，称为“第二级分解”或“分支分解”。这一步得到的结果便是系统的初始SC图。

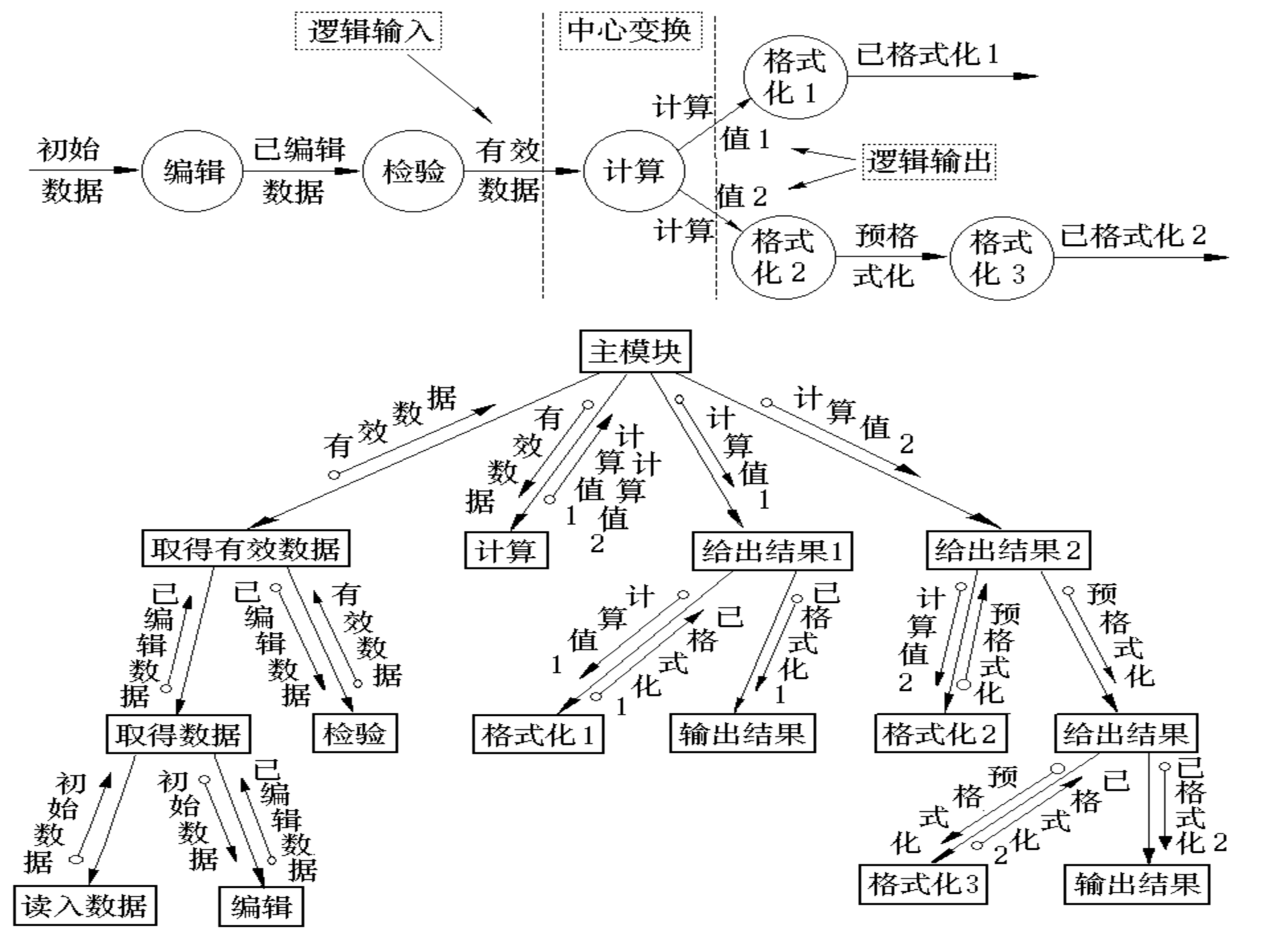


结构化分析与设计



初始SC图


逻辑输入
中心变换
初始
数据
编辑
已编辑
数据
检验
有效
数据
计算
计算
格式
化 1
已格式化 1
值 1
逻辑输出
值 2
计算
格式
化 2
预格
式化
格式
化 3
已格式化 2

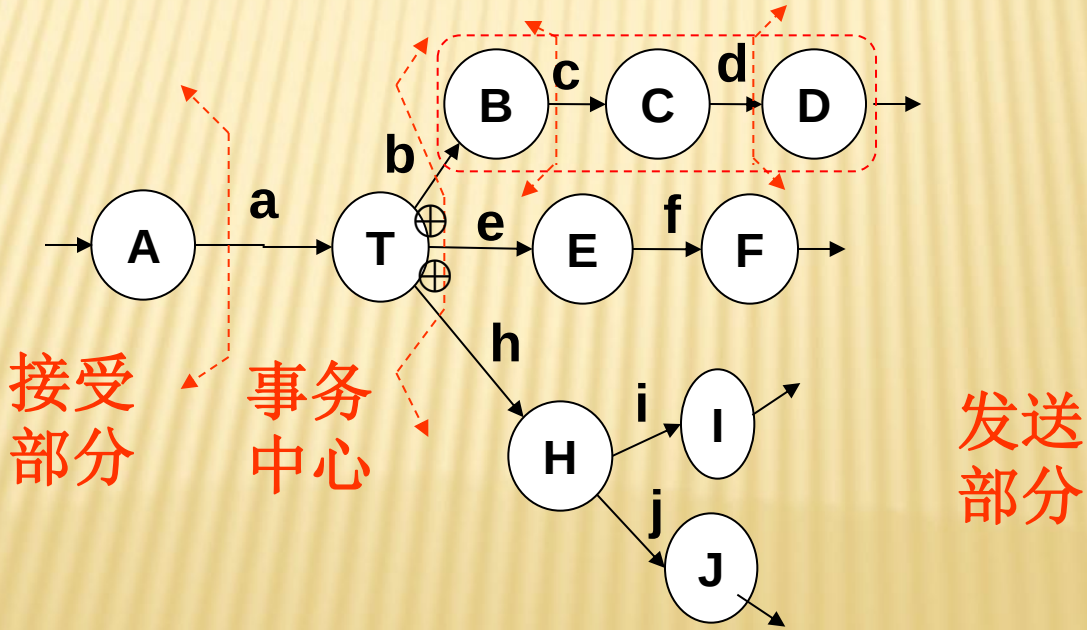

逻辑输入
中心变换
初始数据
编辑
已编辑数据
检验
有效数据
计算
计算值1
计算值2
格式化1
已格式化1
逻辑输出
格式化2
预格式化
格式化3
已格式化2
主模块
有效数据
取得有效数据
有效数据
计算
计算值1
计算值2
给出结果1
计算值2
给出结果2
已编辑数据
取得数据
已编辑数据
有效数据
检验
计算值1
已格式化1
格式化1
已格式化1
输出结果
计算值2
预格式化
格式化2
预格式化
给出结果
初始数据
读入数据
初始数据
已编辑数据
编辑
预格式化2
已格式化2
格式化3
已格式化2
输出结果

结构化分析与设计

➢事务映射

✓步骤1：在DFD图上确定事务中心、接受部分(包括接受路径)和发送部分(包括全部动作路径)。

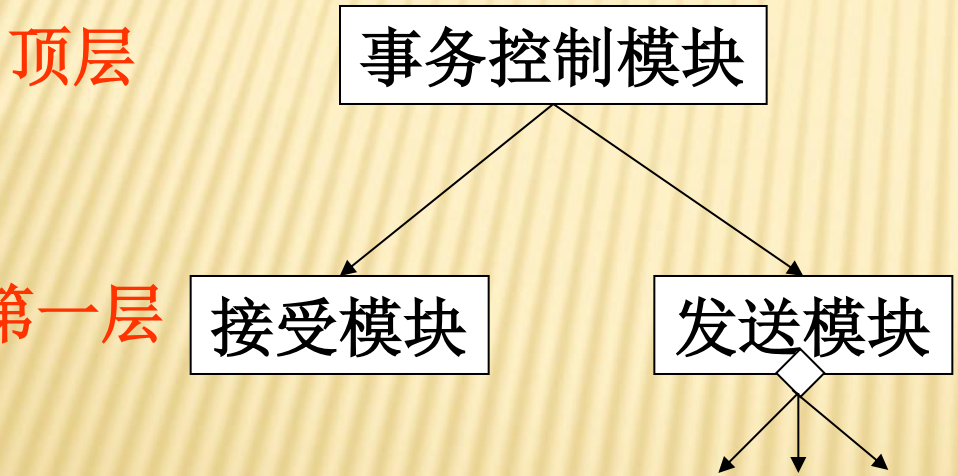
✓例如，对下面DFD图进行划分，区分接受部分、事务中心和传出部分。



结构化分析与设计

✓步骤2：画出SC图初始框架，把DFD图的3个部分分别映射为事务中心控制模块、接受模块和动作模块。

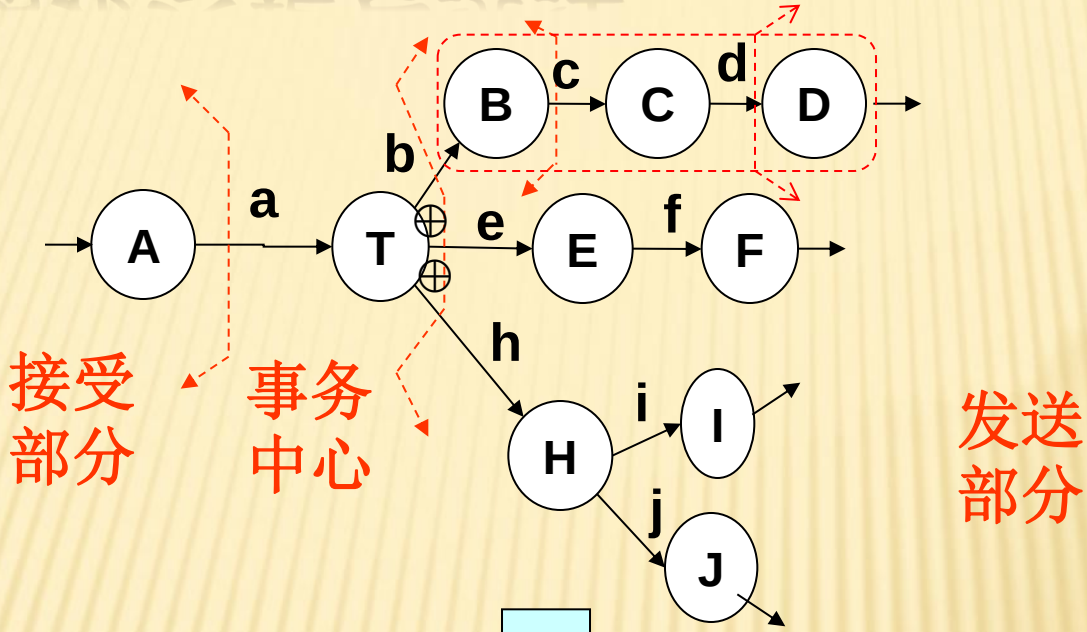
SC图的初始框架是分为两个层次，如图(a)所示。如果第一层模块比较简单，也可以并入顶层，形成如图(b)所示的单层结构。



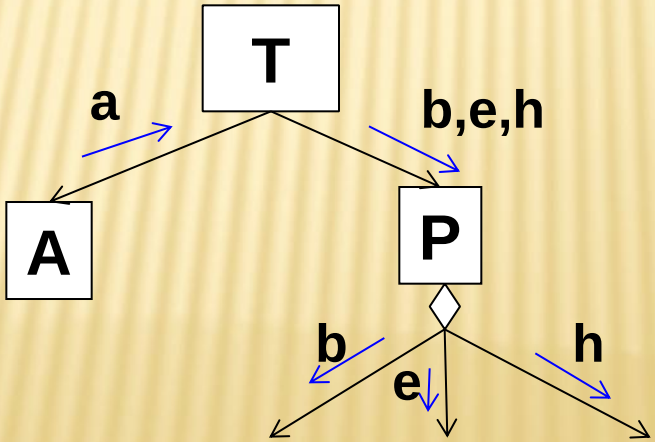
(a)典型的两层结构

(b)单层结构

结构化分析与设计



初始
SC
框架



结构化分析与设计

✓步骤3：分解和细化接受分支和发送分支，按成初始SC图。

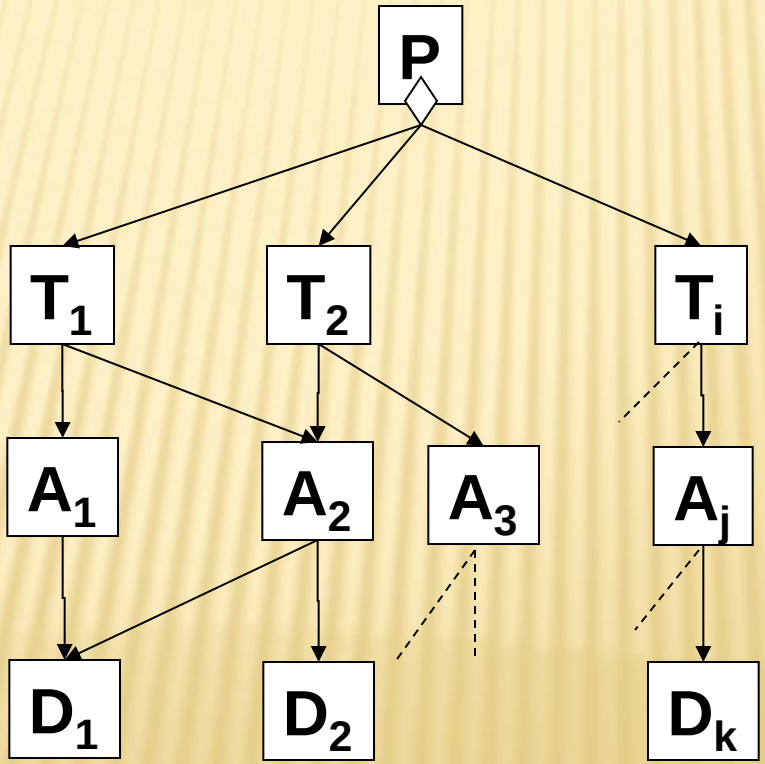
典型的动作分支结构分为四层：处理层P、事务层T、操作层A和细节层D。

处理层

事务层

操作层

细节层

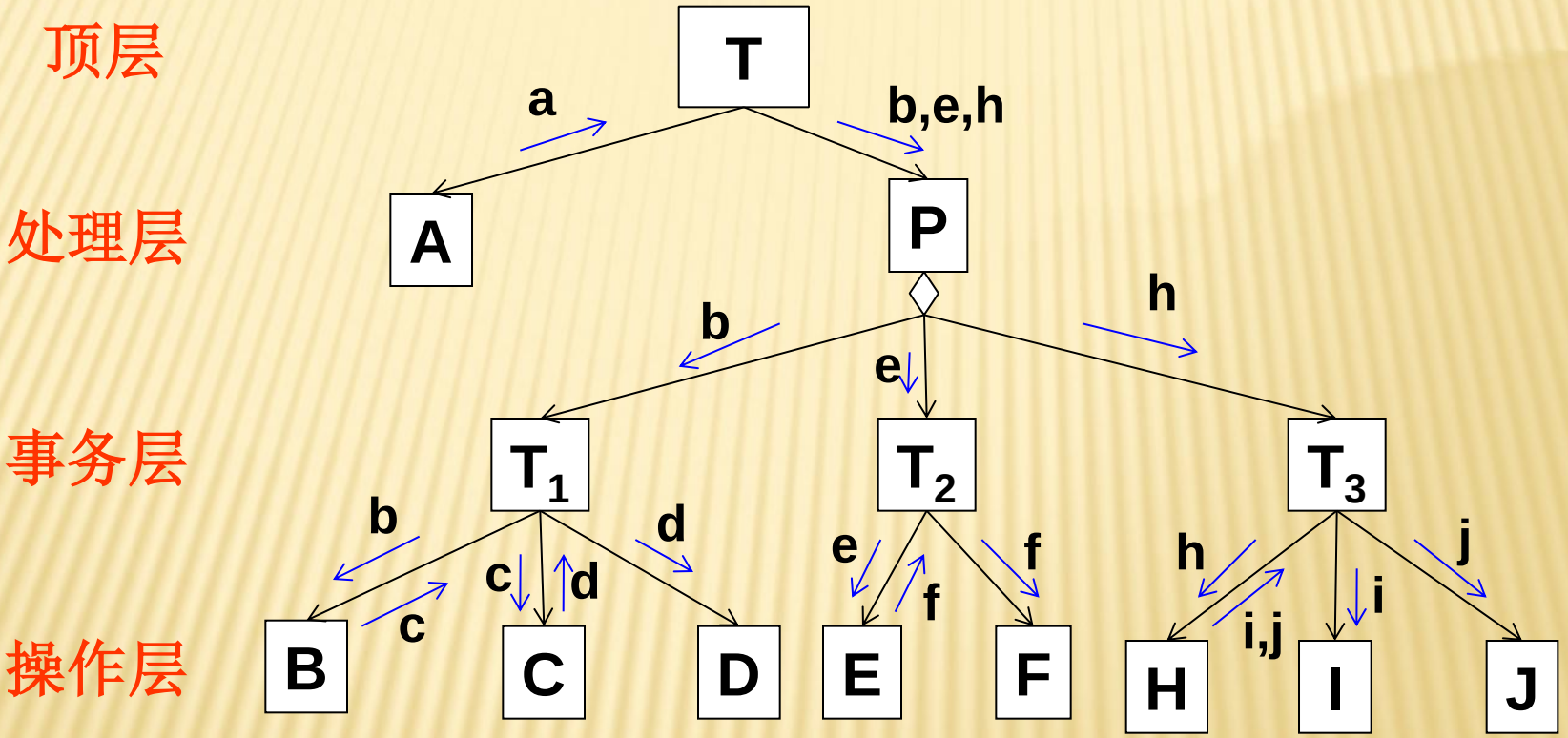


处理层相当于发送模块

每一个动作路径映射为一个事务模块

操作层和细节层的模块被上层模块所共享，可以被多个上层模块调用

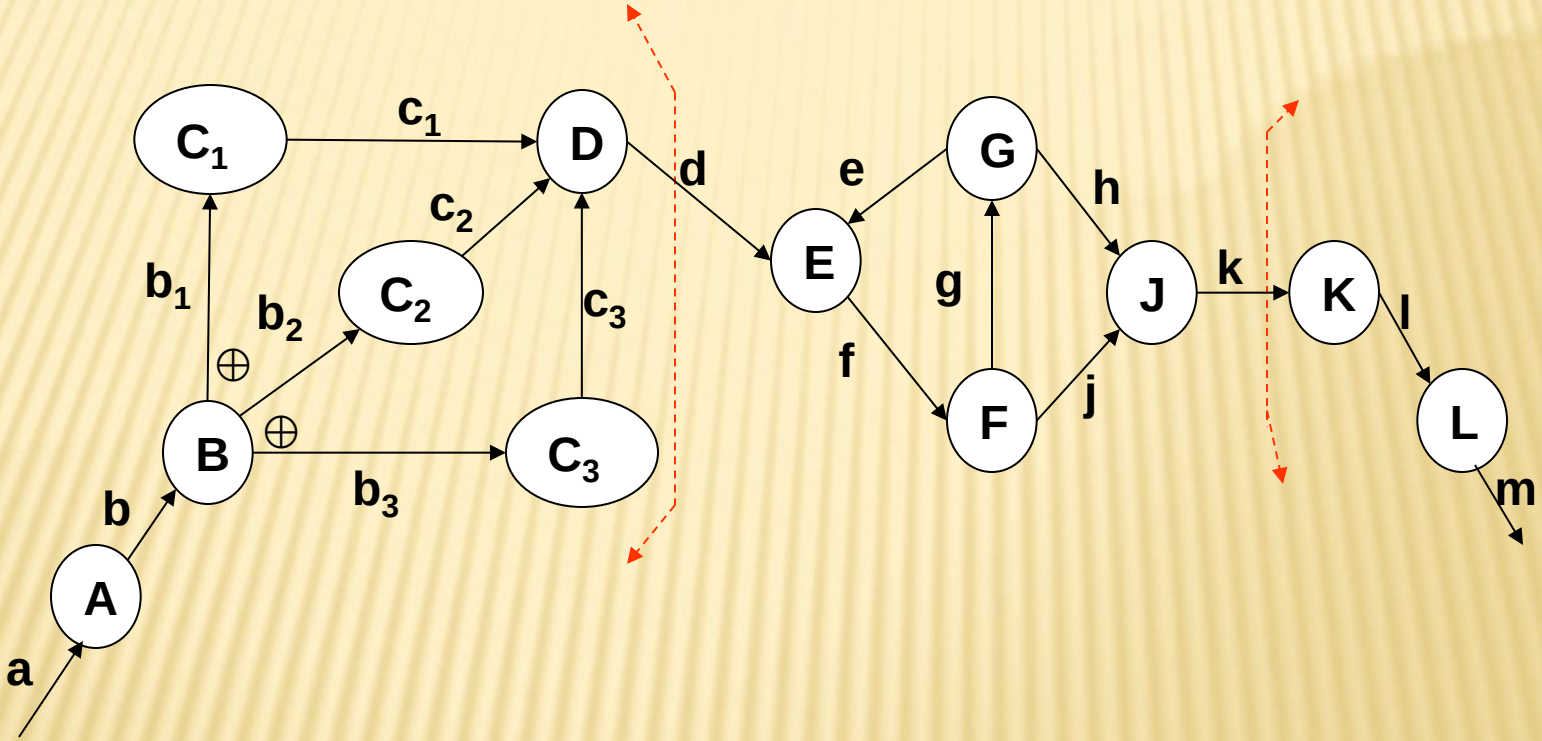
结构化分析与设计



初始SC图

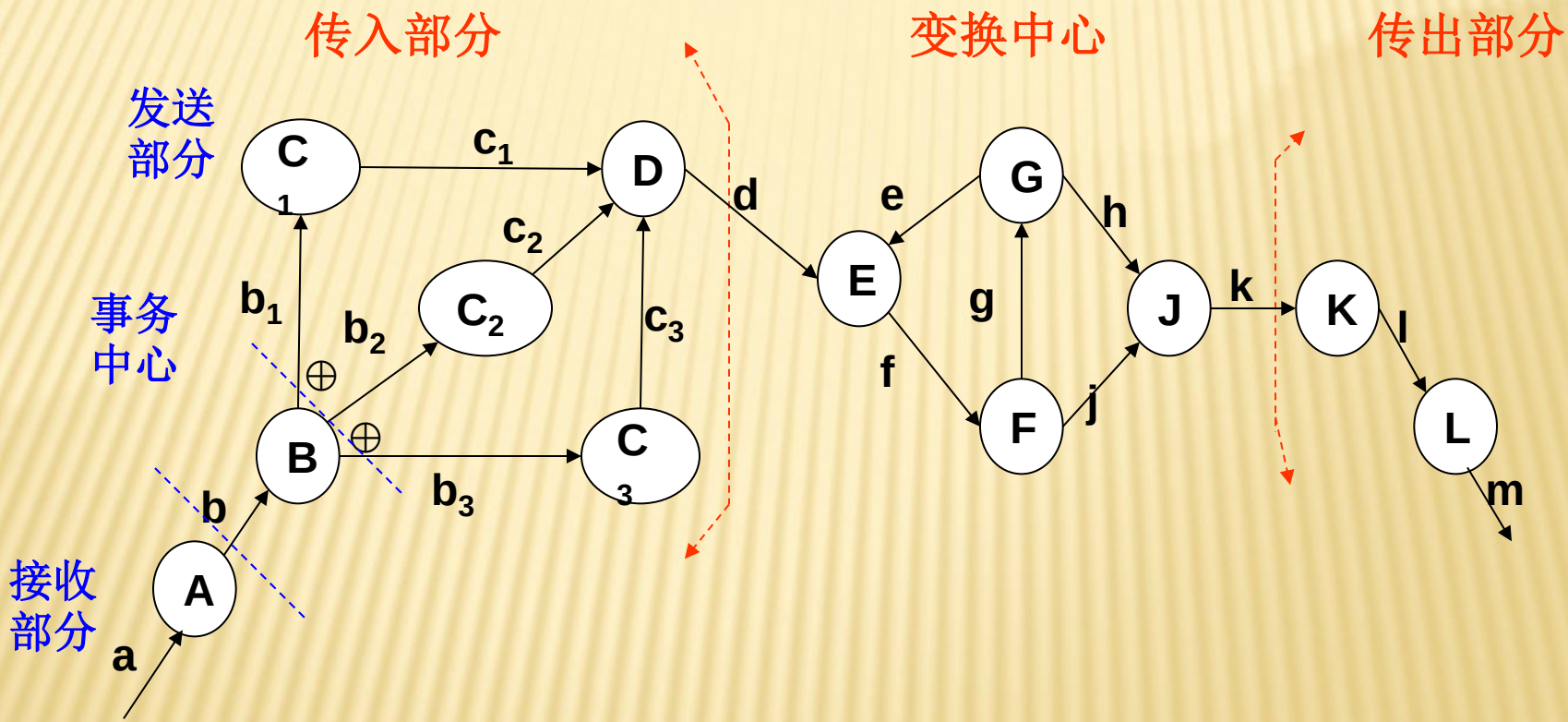
结构化分析与设计

- 思考题：将下列DFD图转换为初始SC图。

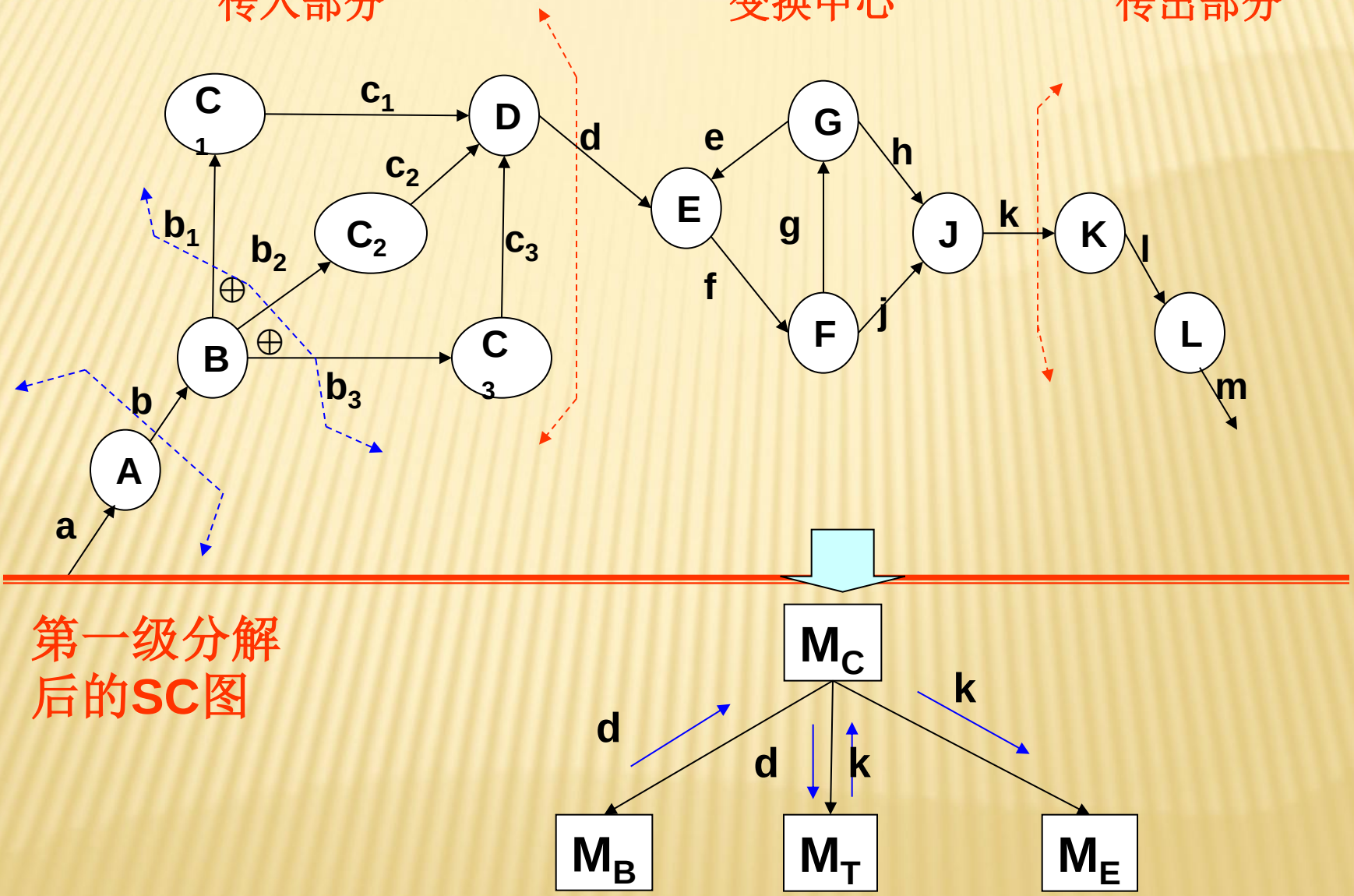


结构化分析与设计

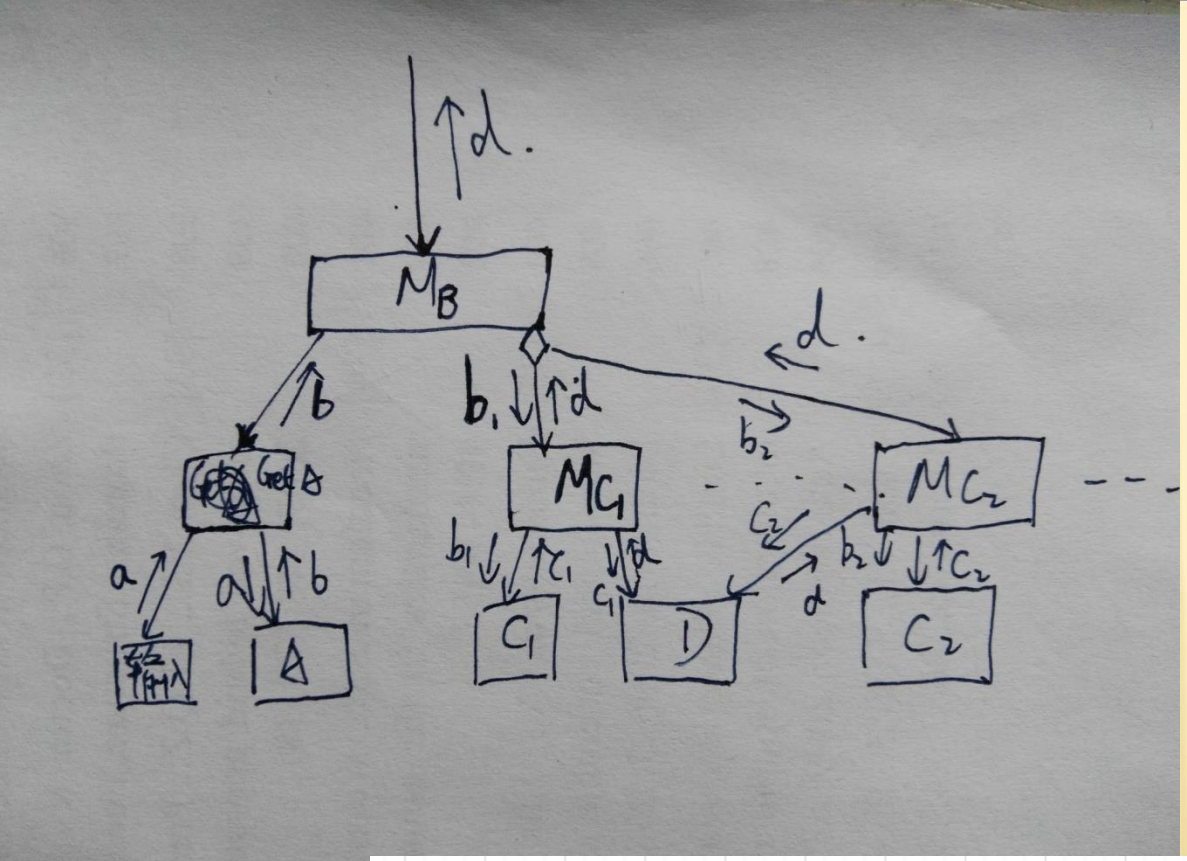
✓判断DFD图的结构类型。

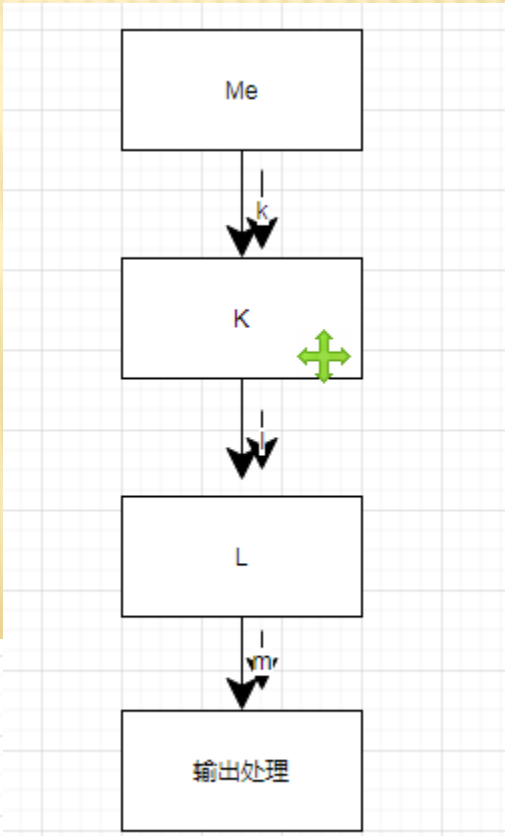


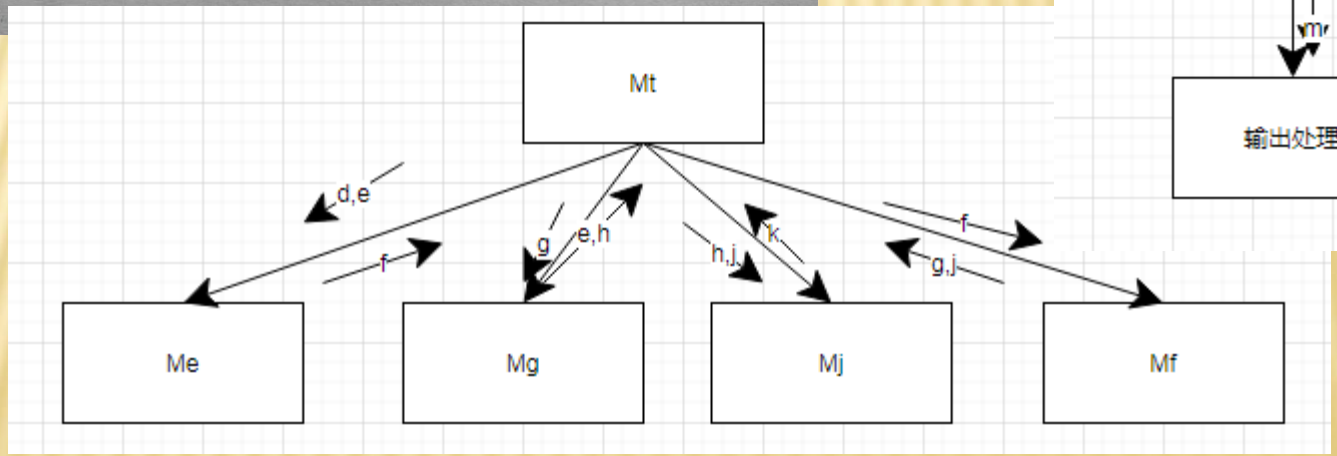
结构化分析与设计



第一级分解后的SC图


d.
MB
b
b1
d
MC1
b2
MC2
C2
a
C1
D
C2


Me
k
K
j
L
m
输出处理


Mt
d,e
f
g
e,h
h,j
k
g,j
f
Me
Mg
Mj
Mf

➢优化初始SC图的指导原则

✓对模块划分的原则

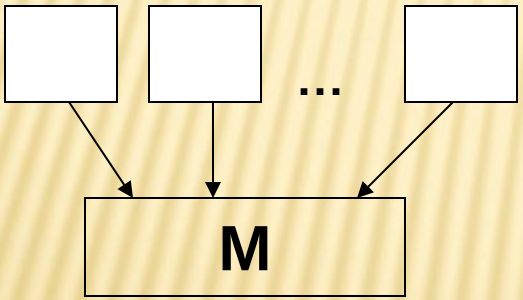
- 以提高模块的独立性为首要标准
- 提高内聚，降低耦合，简化模块接口
- 少用全局性数据和控制信息
- 模块大小适中(10到100行)

结构化分析与设计

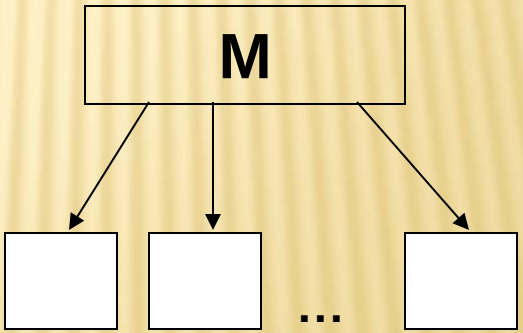
✓高扇入/低扇出的原则

- **扇入：** 调用一个给定模块的次数。扇入高则上级模块多，能够增加模块的利用率。

- **扇出：** 一个模块直接调用其它模块数目。扇出低则表示下级模块少，可以减少模块调用和控制的复杂度。通常扇出数以3-4个为宜，最好不超过5-7个。



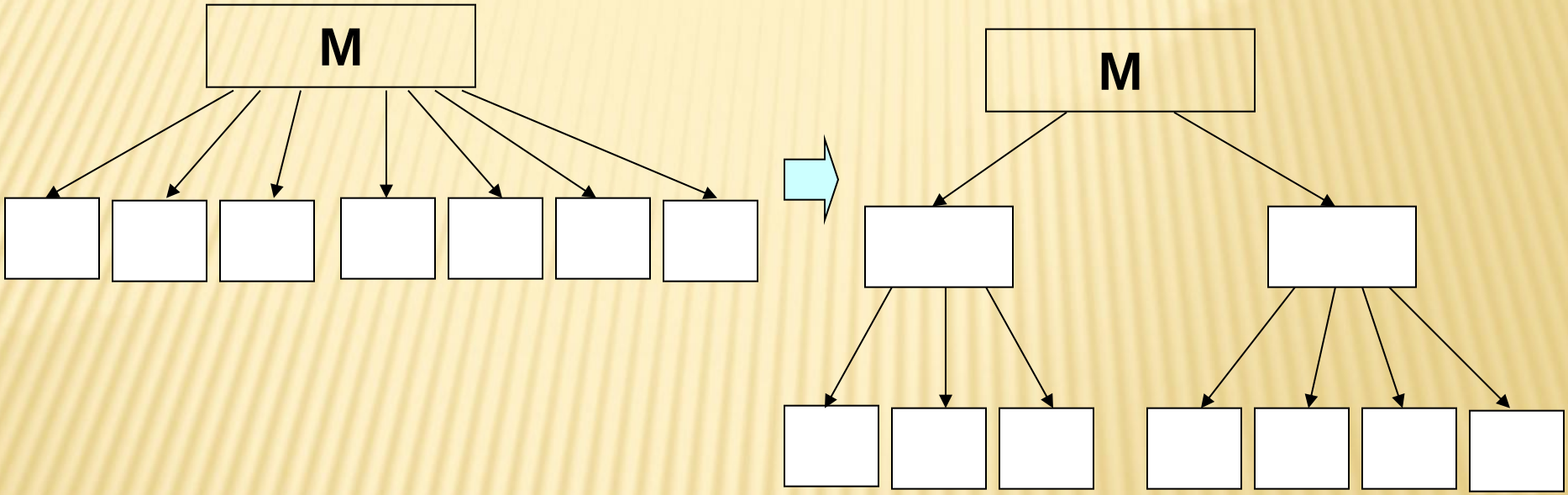
M的扇入



M的扇出

结构化分析与设计

- 通过增加中间层减少扇出

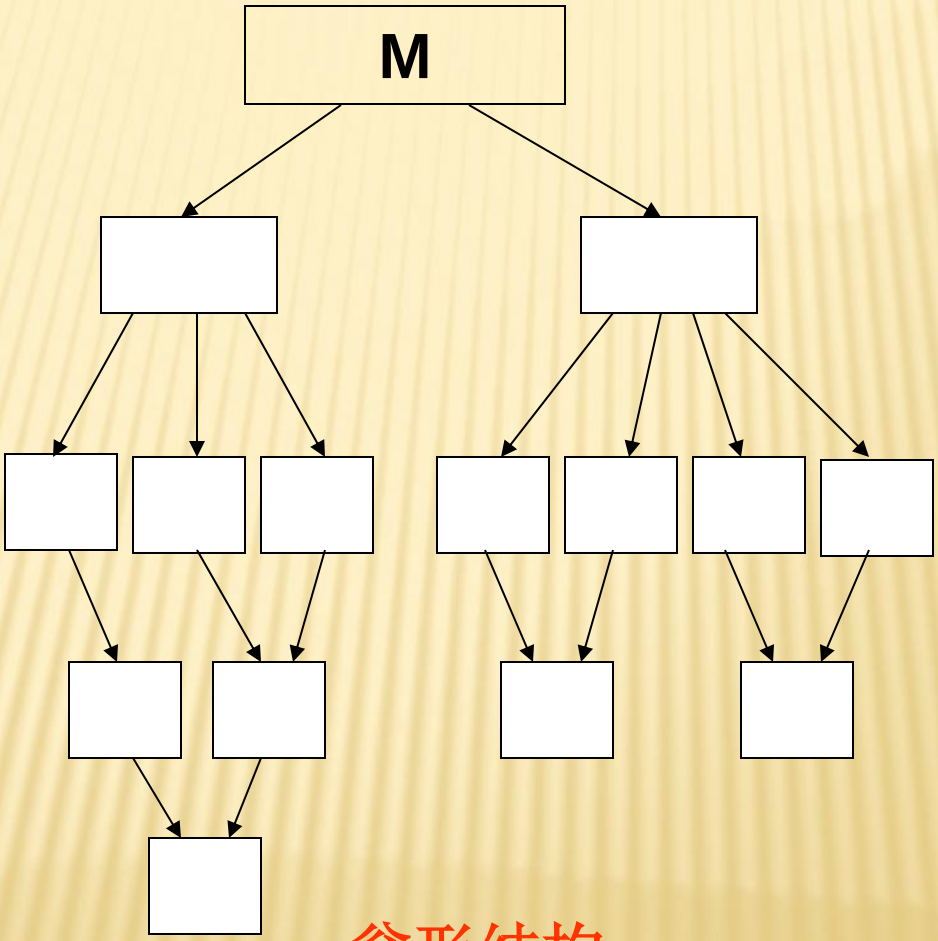


煎饼形结构

塔形结构

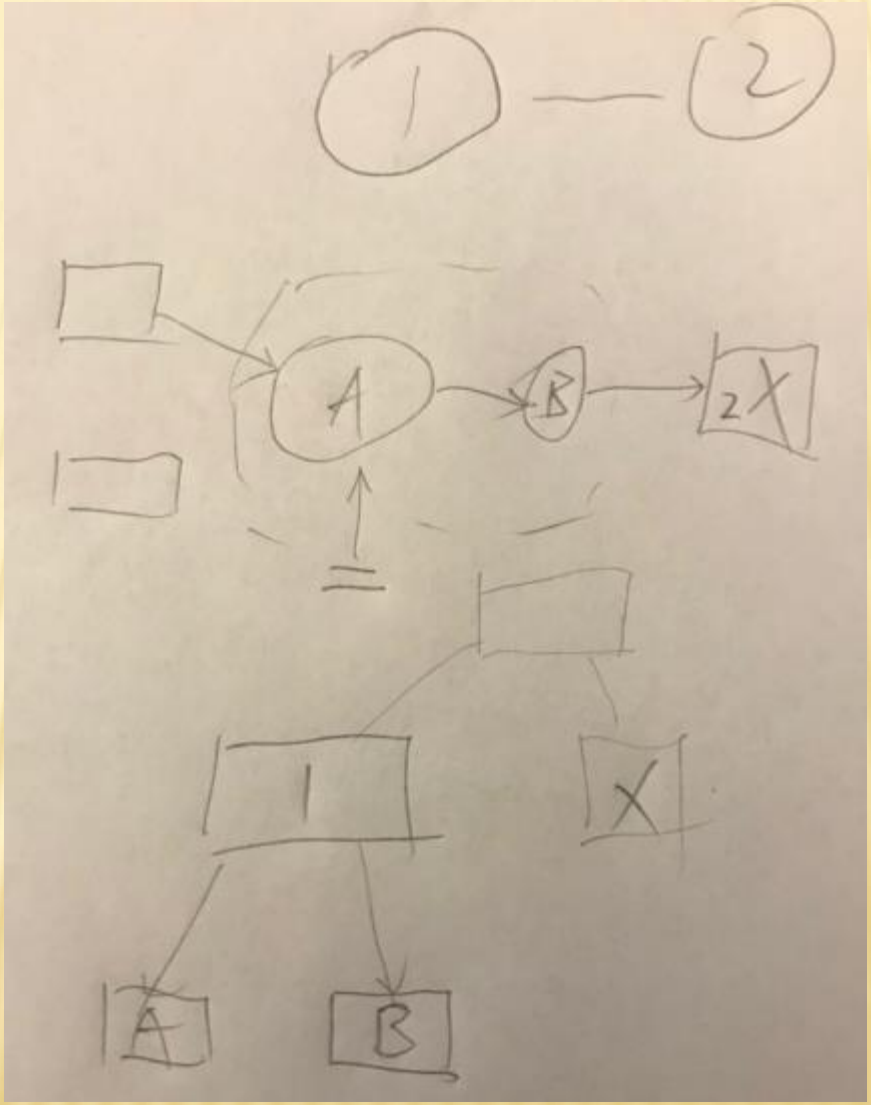
结构化分析与设计

- 设计良好的软件通常具有瓮形结构，两头小，中间大。



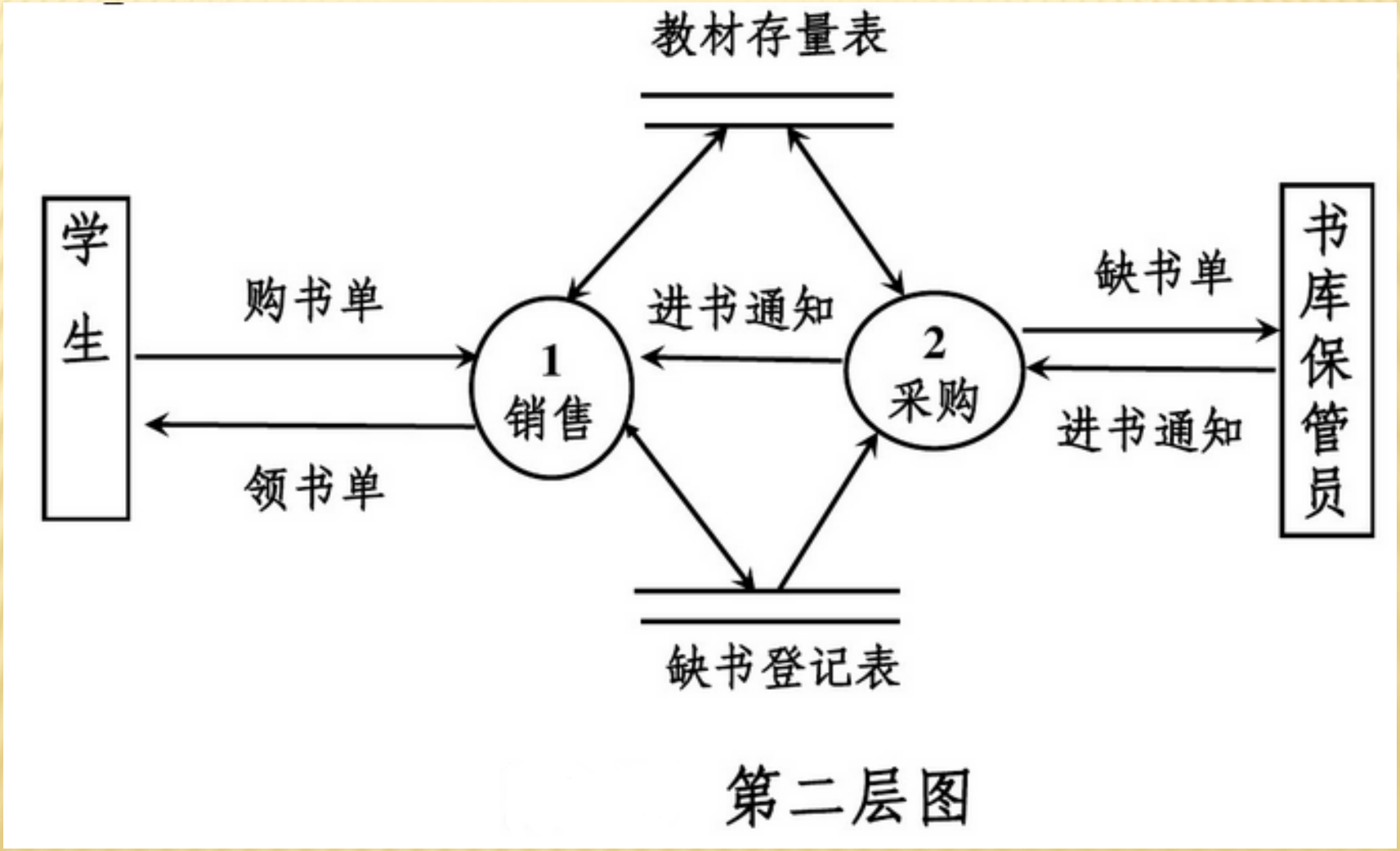
瓮形结构

从完整dfd转向sc



结构化分析与设计

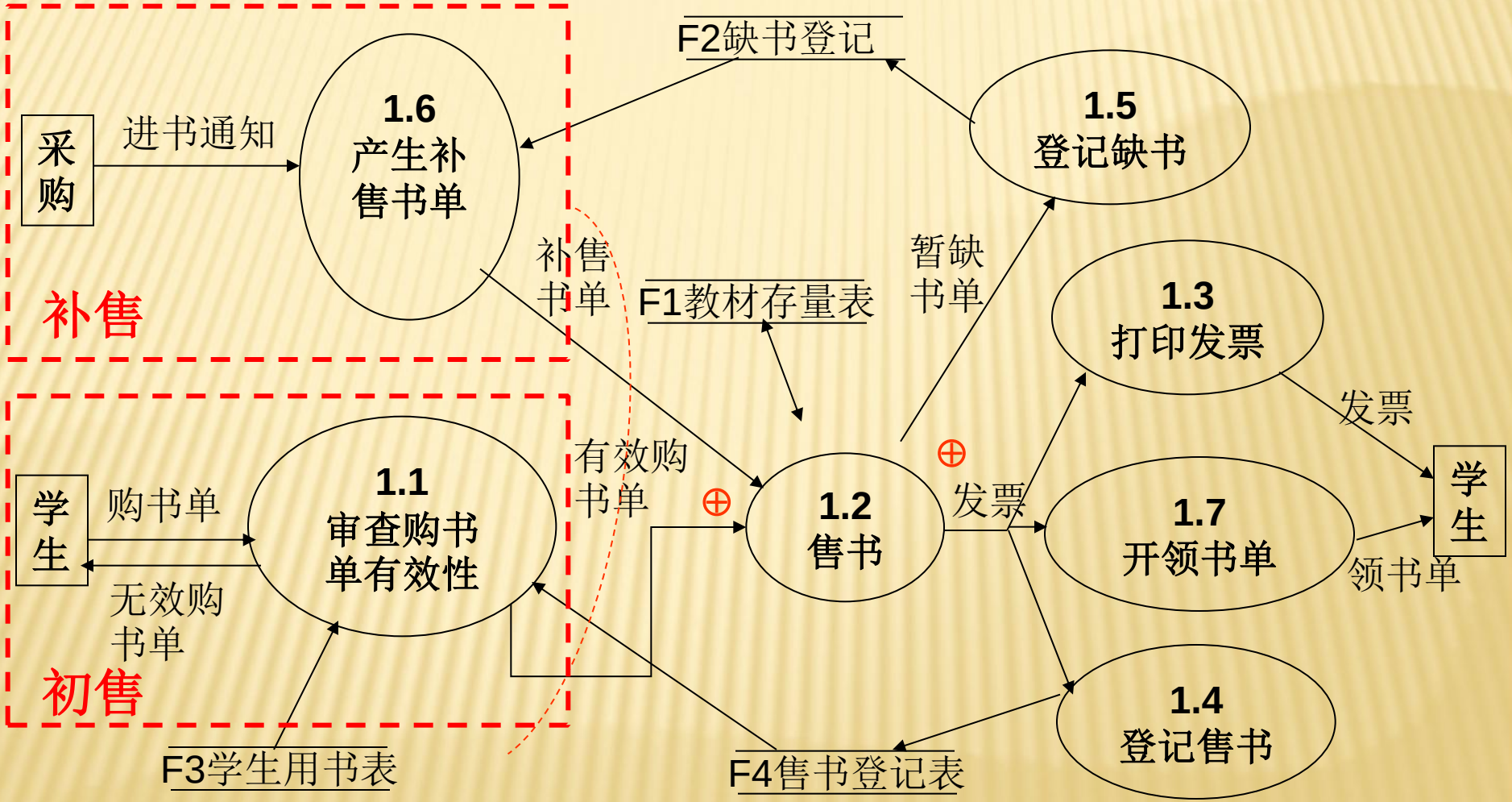
■教材购销系统的结构设计示例



第二层图

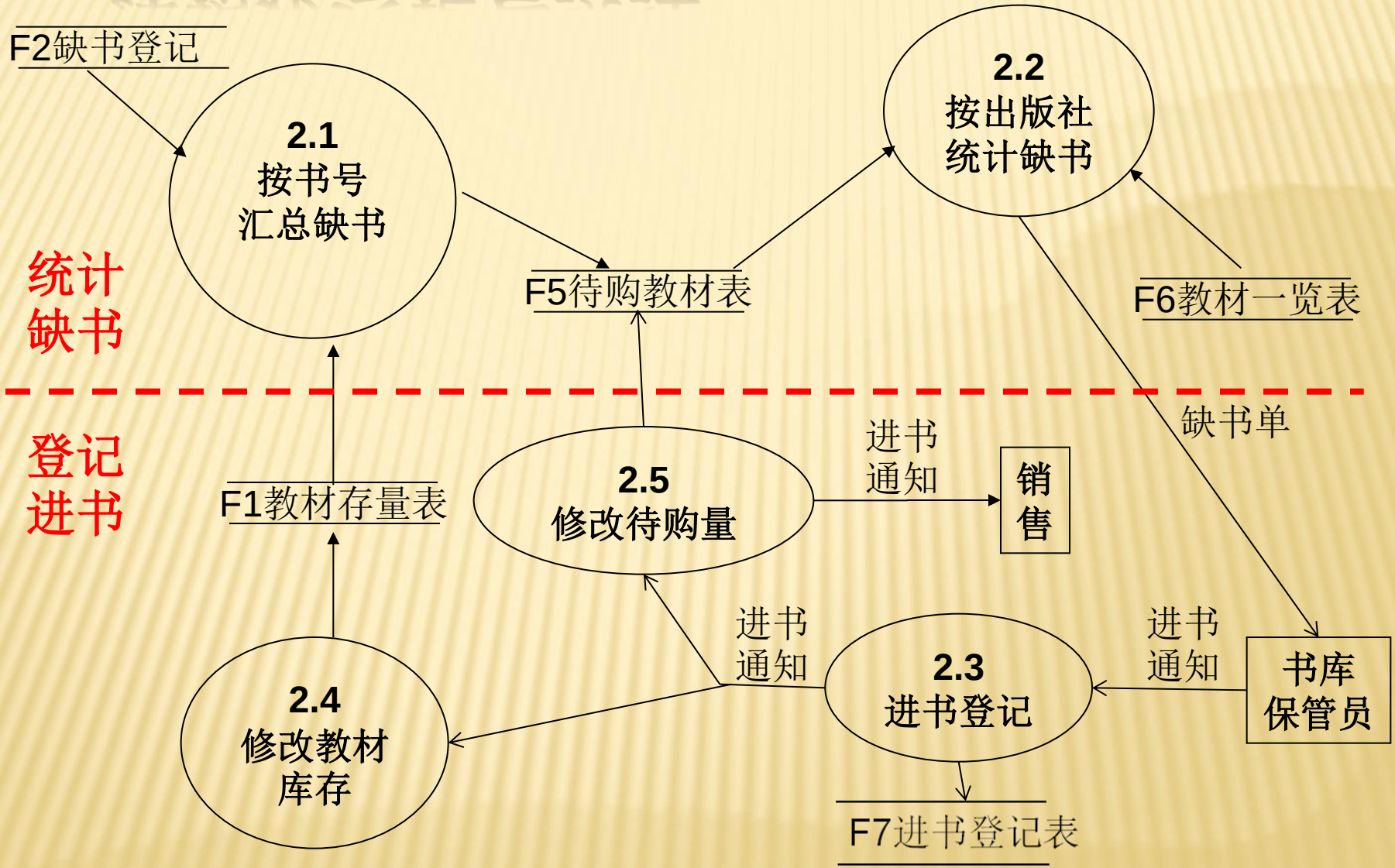
结构化分析与设计

■教材购销系统的结构设计示例



修改后销售子系统DFD图

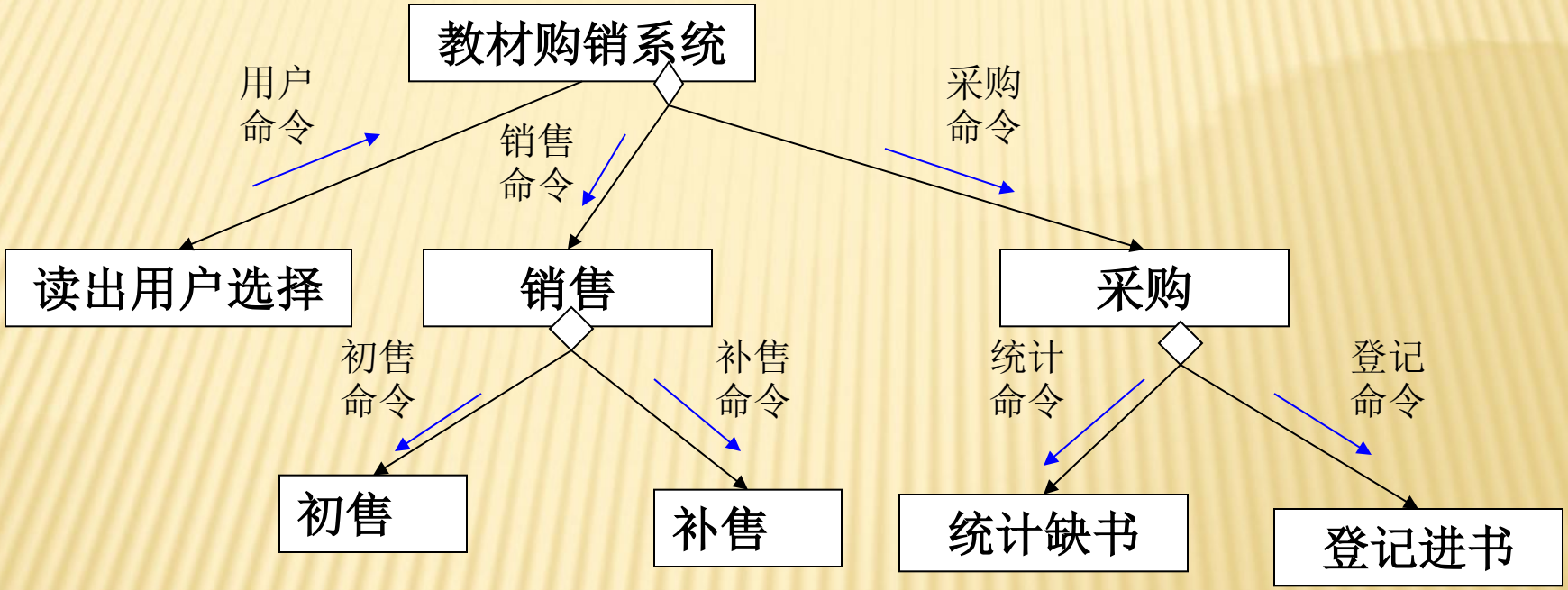
结构化分析与设计



修改后采购子系统DFD图

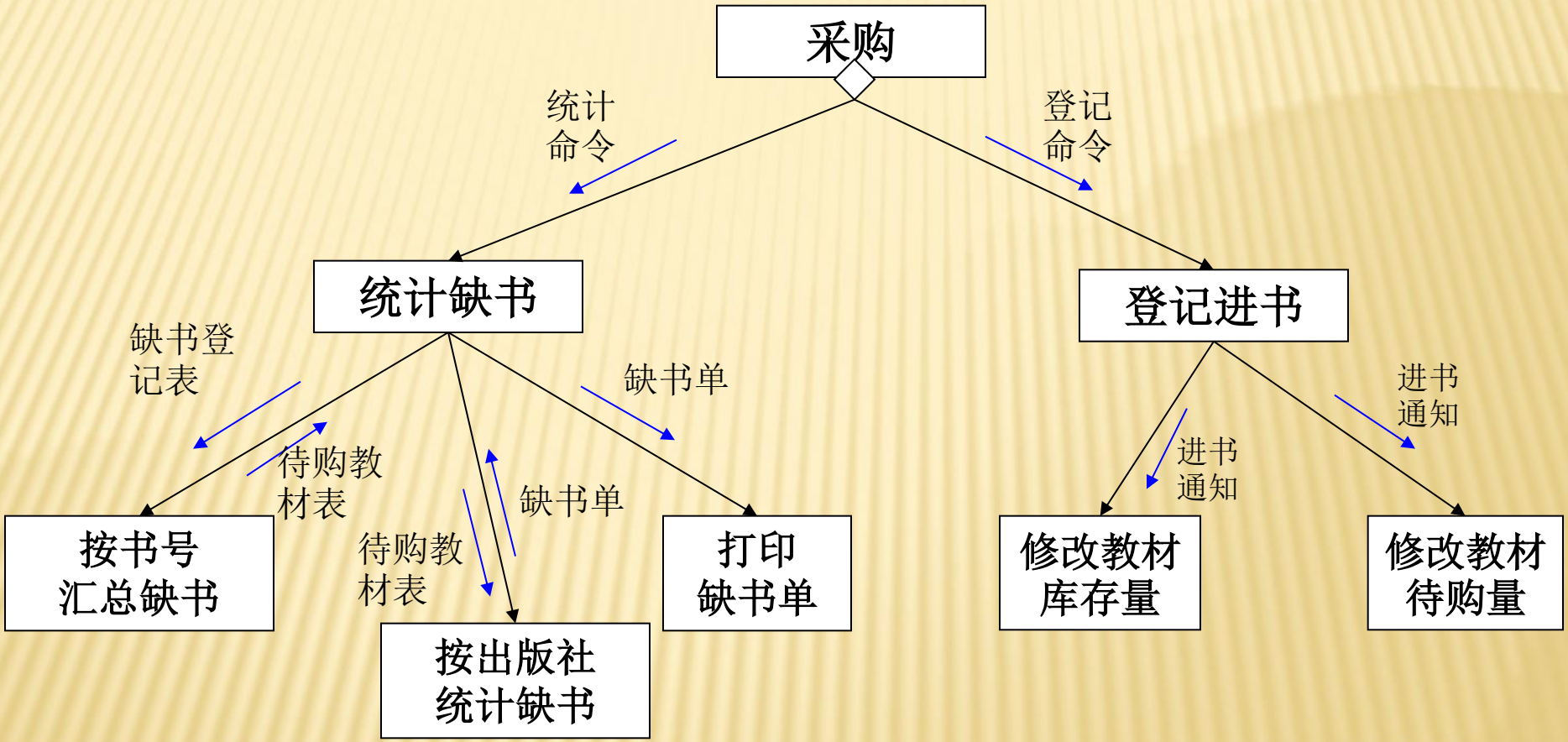
结构化分析与设计

➢初始SC图



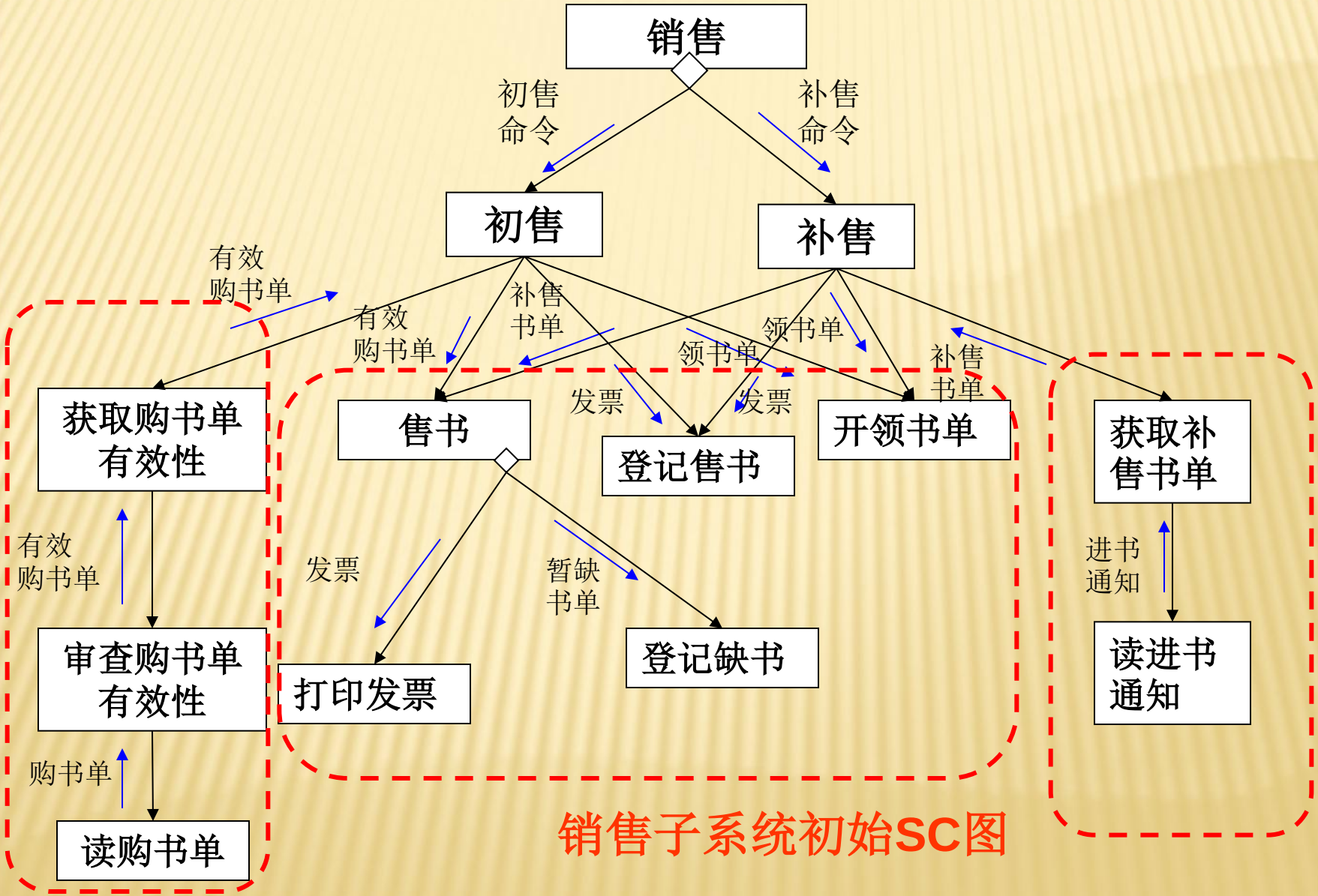
教材购销系统上层框架

结构化分析与设计



采购子系统初始SC图

结构化分析与设计



销售子系统初始SC图