《系统分析与设计》
System Analysis and Design

任课教师：范国祥

电　　话：0451-86418876-811(O)

13199561265(微信同号)

邮　　箱：fgx@hit.edu.cn

哈工大计算学部/

国家示范性软件学院

软件工程教研室

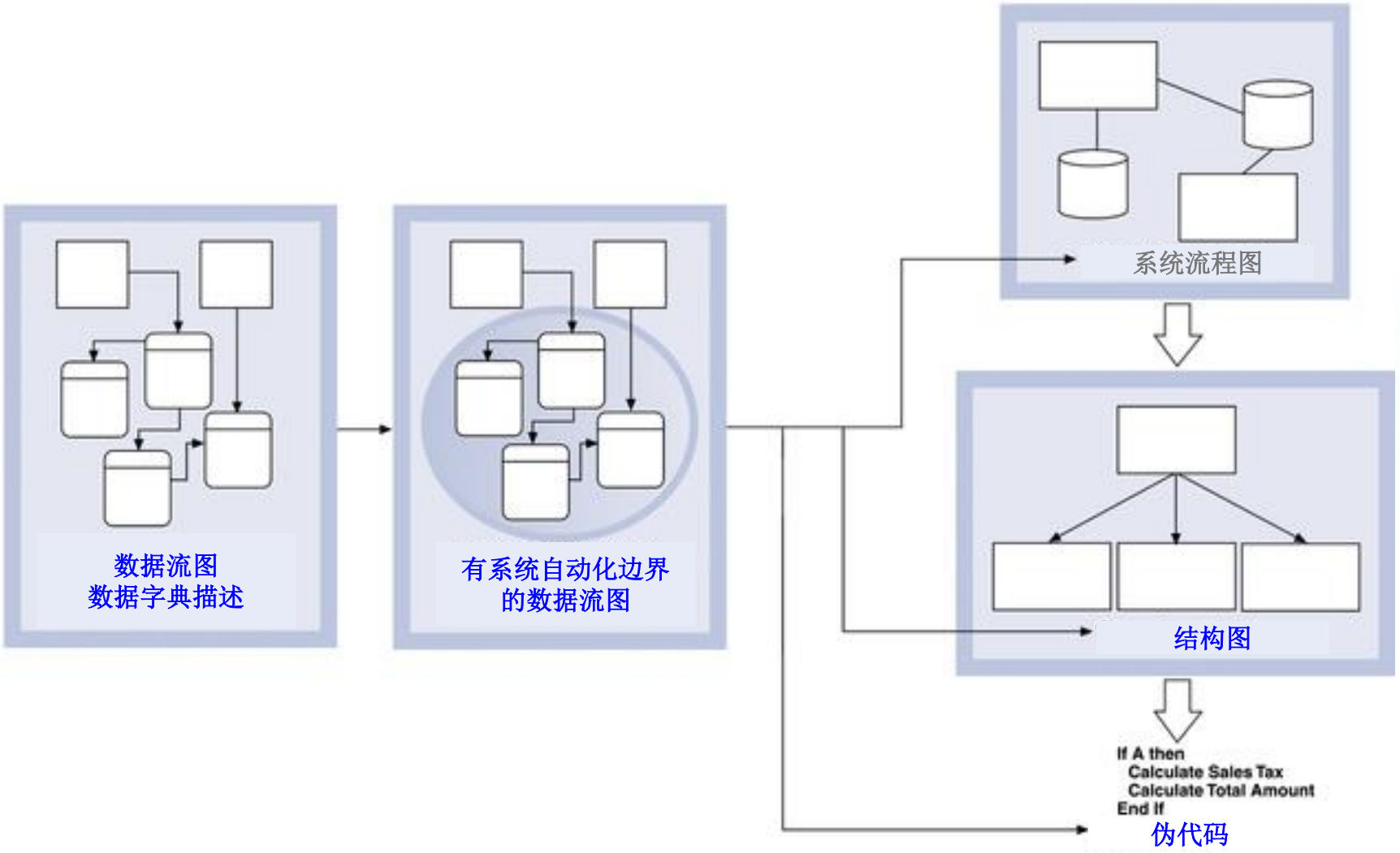
2022.03

结构化设计的内容

- 结构化的系统设计主要考虑：
  - 模块的层次化
  - 模块之间的接口
  - 数据库设计（后续章节讲授）
  - 用户界面设计（后续章节讲授）
- 为每个模块设计内部逻辑
- 采用自顶向下的方法进行设计
  - 具有系统自动化边界的DFD
  - 结构图

结构化的系统设计

结构化设计模型

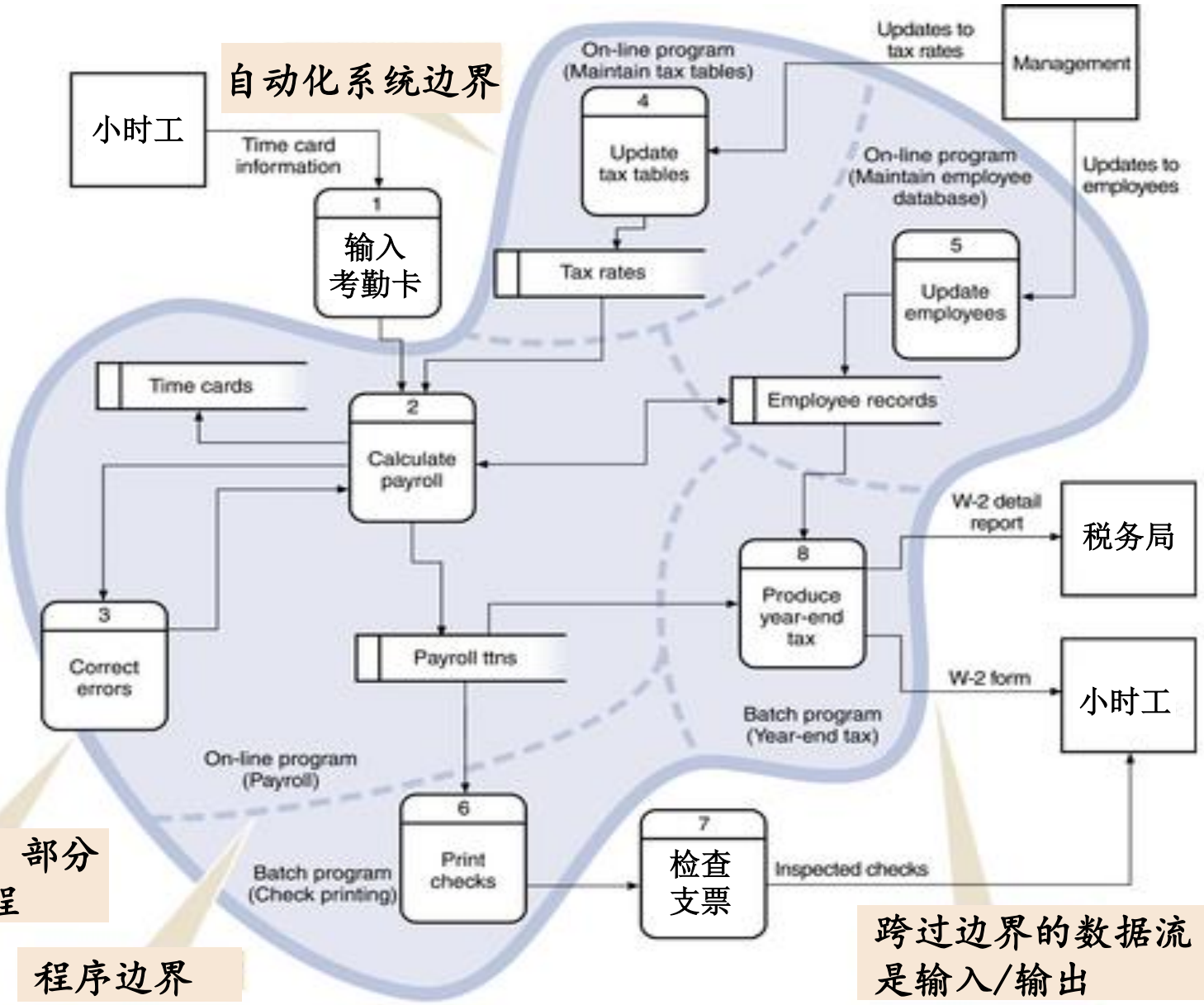


自动化系统边界划分（Automation System Boundary）

- 将数据流图划分为手工处理部分和系统能自动完成的部分
- 程序的处理过程可以在系统边界内部或外部
- 数据流可以在系统边界内部或外部
- 穿过系统界线的数据流代表了系统的输入和输出
- 在最终的系统中，数据流将成为用户界面中的表单、报表、供其他系统使用的数据文件等

结构化的系统设计

自



部分输入、部分输出的过程

程序边界

跨过边界的数据流是输入/输出

带自动化系统边界的DFD

结构图（structure chart）

结构图定义：以模块为基础、以模块间的调用为关联所构成的图称模块结构图，简称结构图

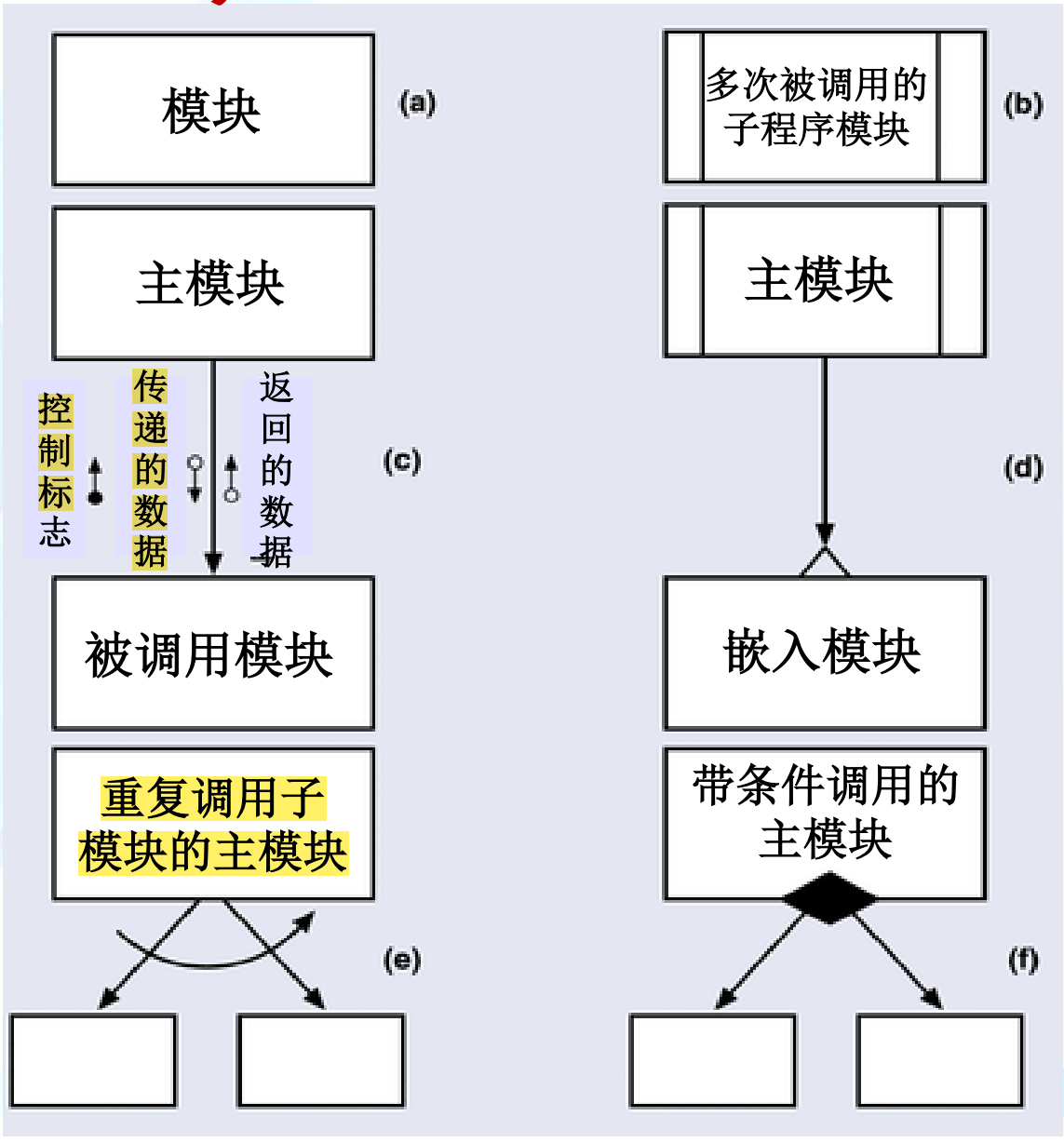
结构图的作用：

- 通过层次的方法来描述系统每部分的功能和子功能
- 展示计算机程序模块间的联系

结构化的系统设计

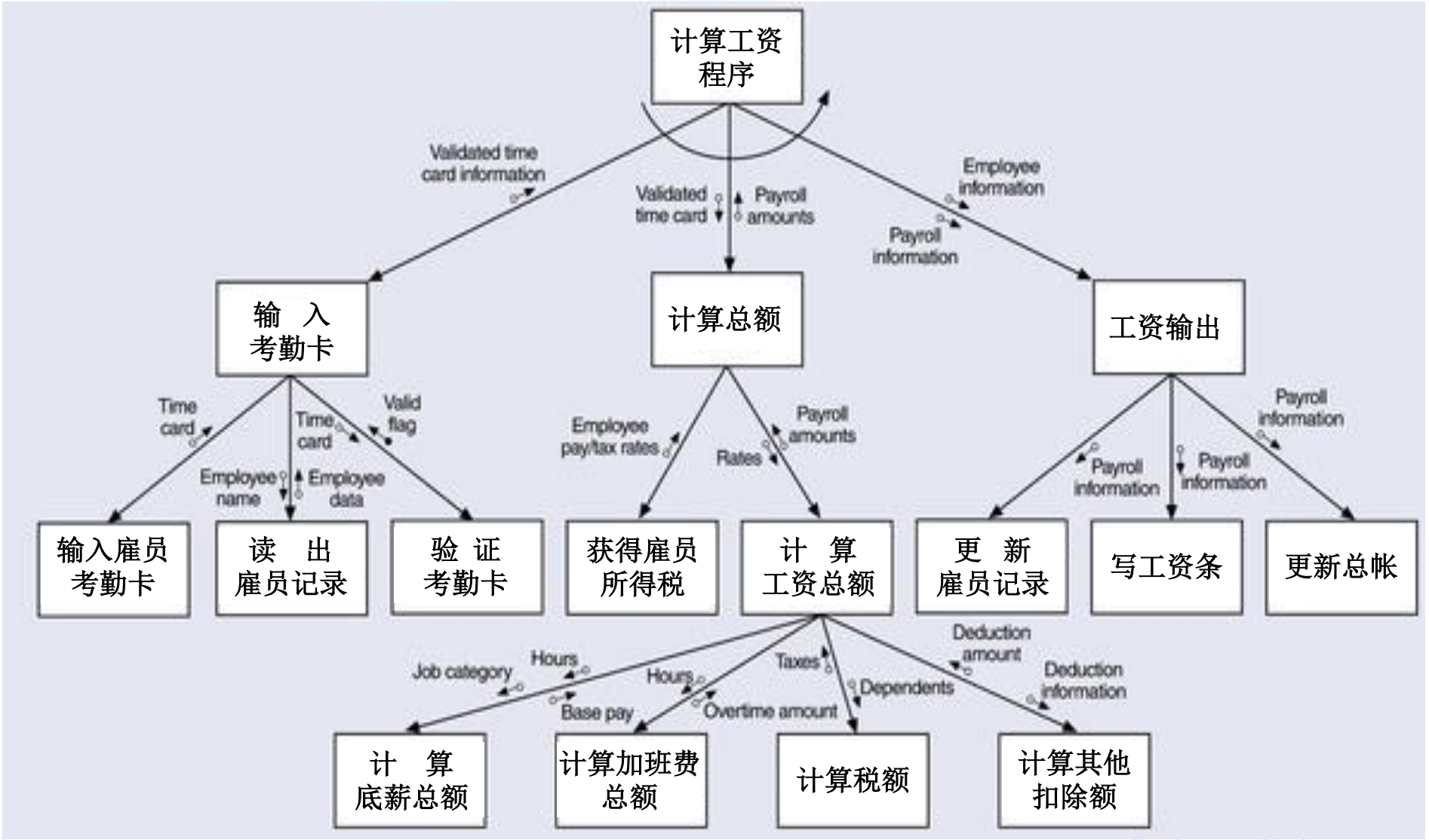
结构图（structure chart）

结构图符号



结构化的系统设计

结构图（structure chart）



结构图示例：工资计算系统的结构图

结构图的创建方法

从一个DFD片断建立一个结构图的步骤

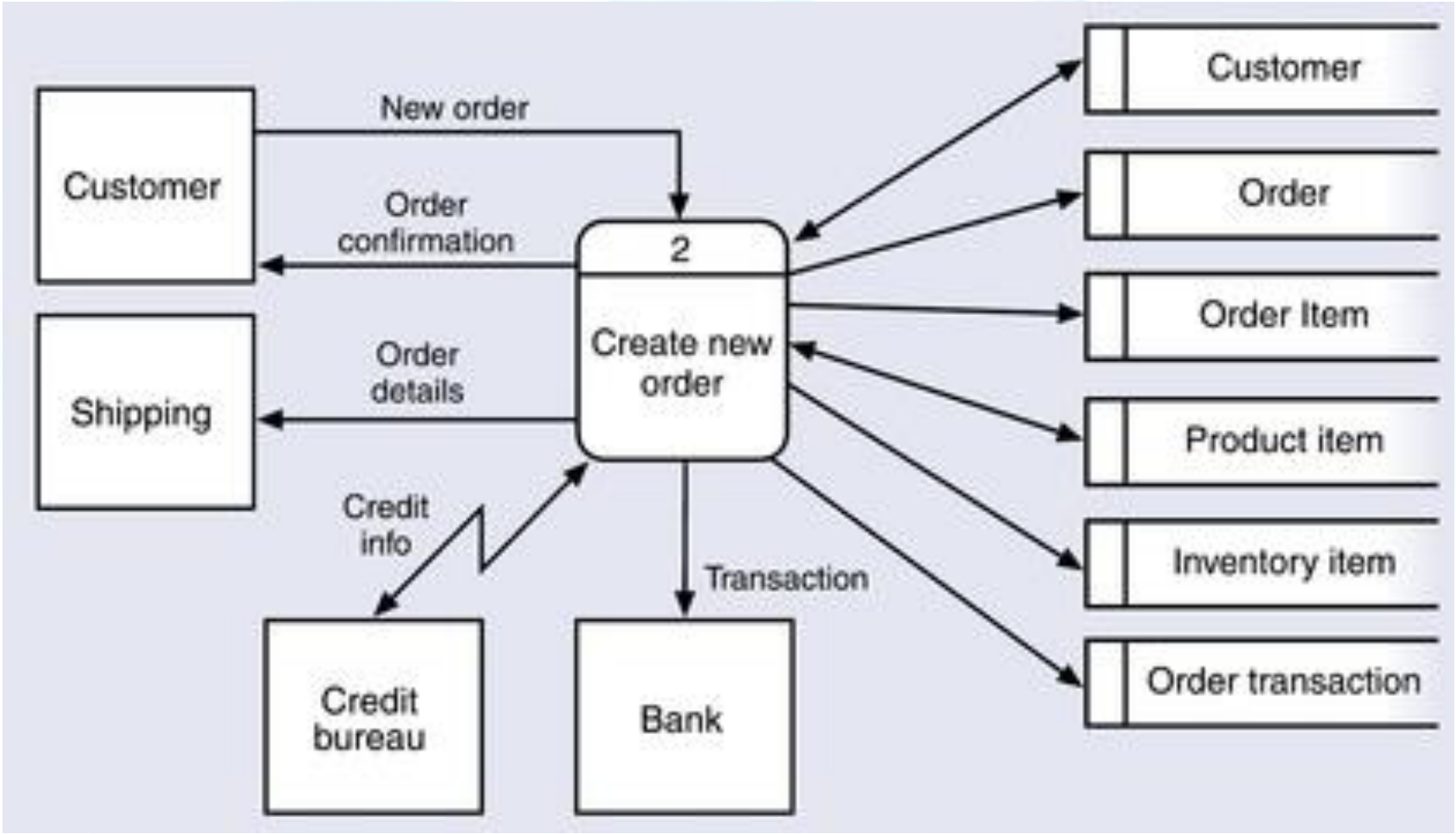
- 确定主要的信息流
- 找出能代表输入和输出间最基本变化的过程
- 重画数据流图并把输入放在左边，输出放在右边
- 初步建立一个结构图草图
- 加入其它模块实现下列功能
  - 数据输入、数据处理、数据输出
- 使用结构化语言或决策树添加模块间逻辑
- 进一步求精

结构图的创建方法

- 转换分析方法
  - “处理”对输入到输出的转换（I-P-O）
  - 结构图包括输入子树、计算子树和输出子树
  - 用数据流图片断作为输入
- 事务分析方法
  - 含有“处理”分支的情况
  - 某“处理”对输入数据流进行分析，根据分析结果选择不同的“处理”

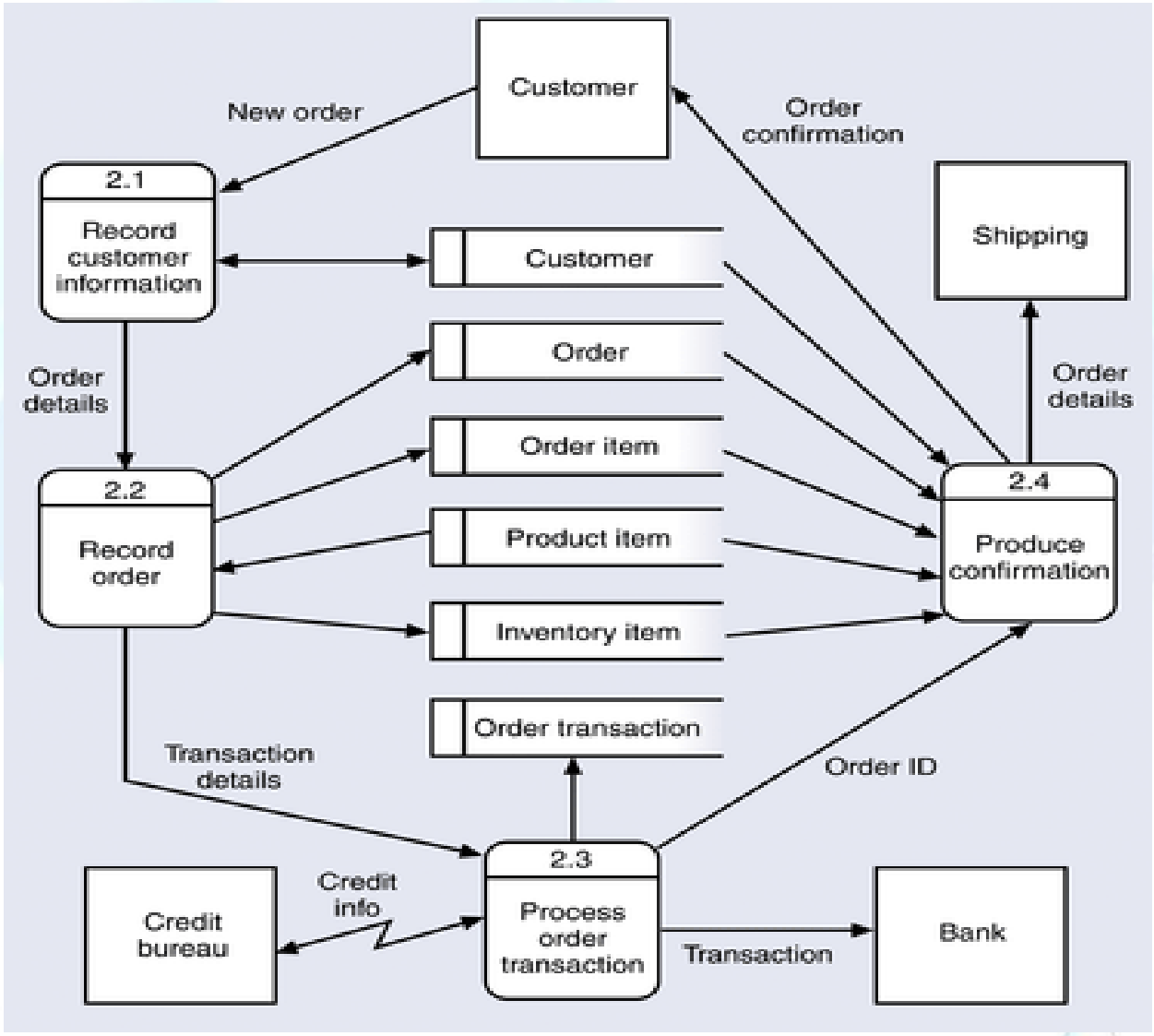
结构化的系统设计

结构图的创建方法



“创建新订单”DFD片断

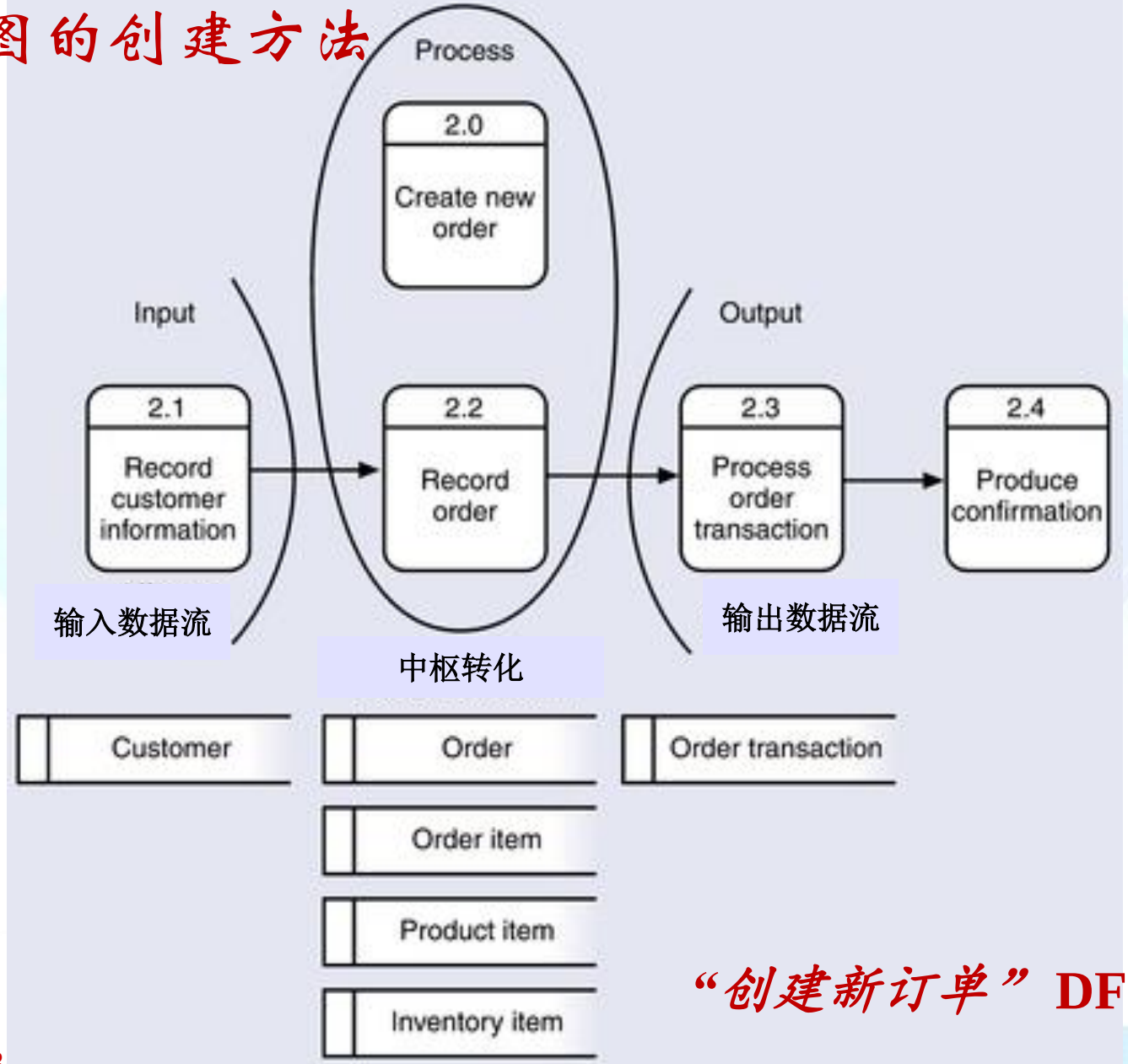
结构图的创建方法



“创建新订单” DFD的分解图

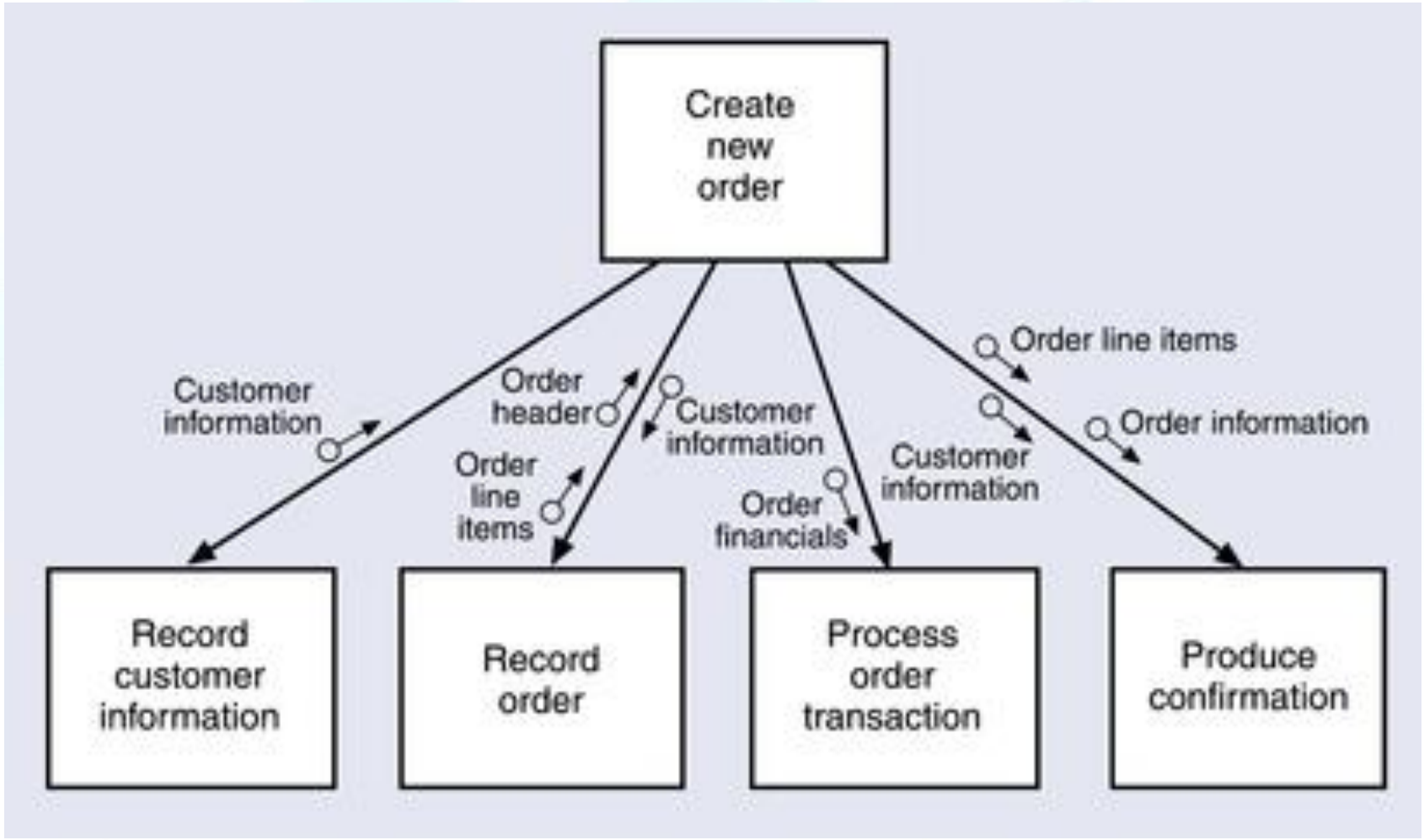
结构化的系统设计

结构图的创建方法



“创建新订单”DFD的重组

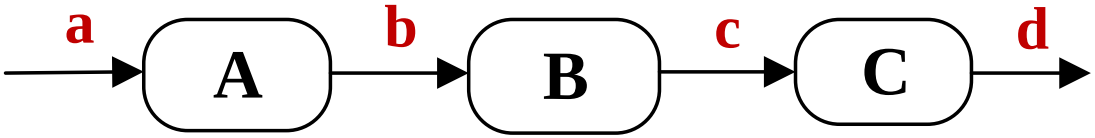
结构图的创建方法



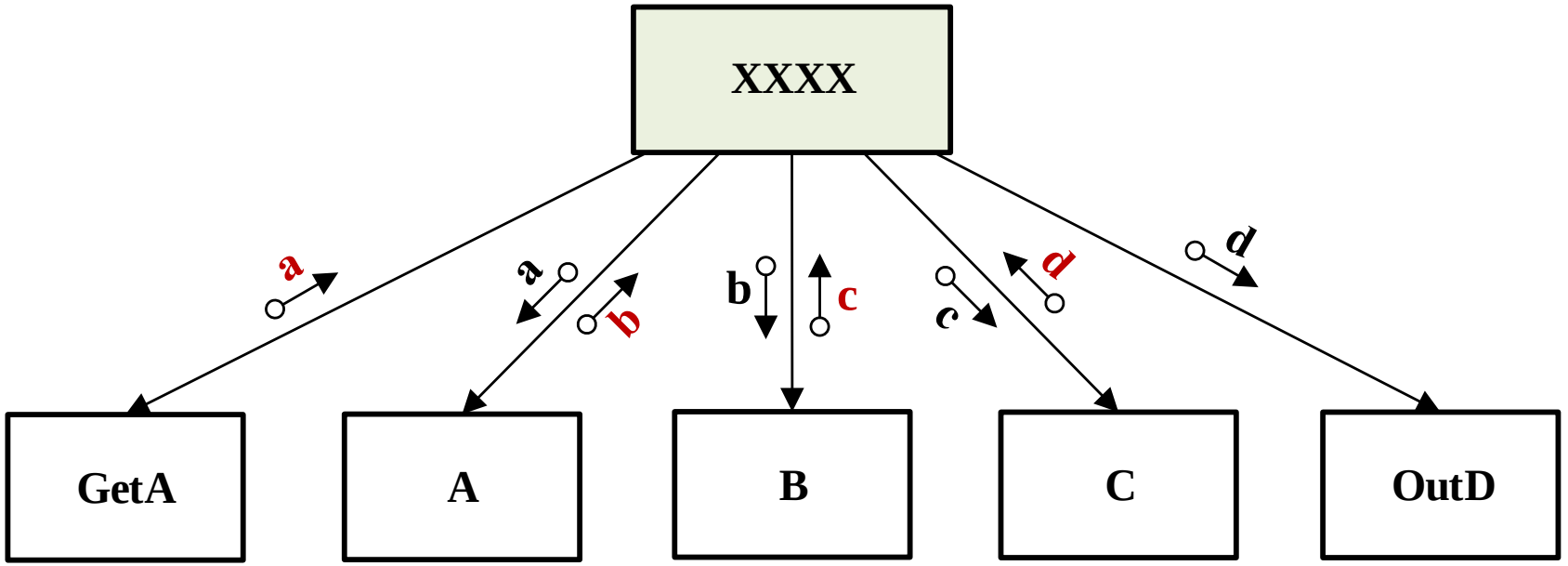
“创建新订单”模块的结构图

DFD到系统结构图转换的基本模式

1.简单变换型DFD→系统结构图转换



第1种情况：不考虑扇入扇出度的问题，得到系统结构图

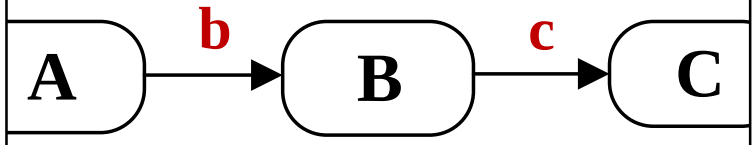


结构化的系统设计

DFD到系统结构图转换的基本模式

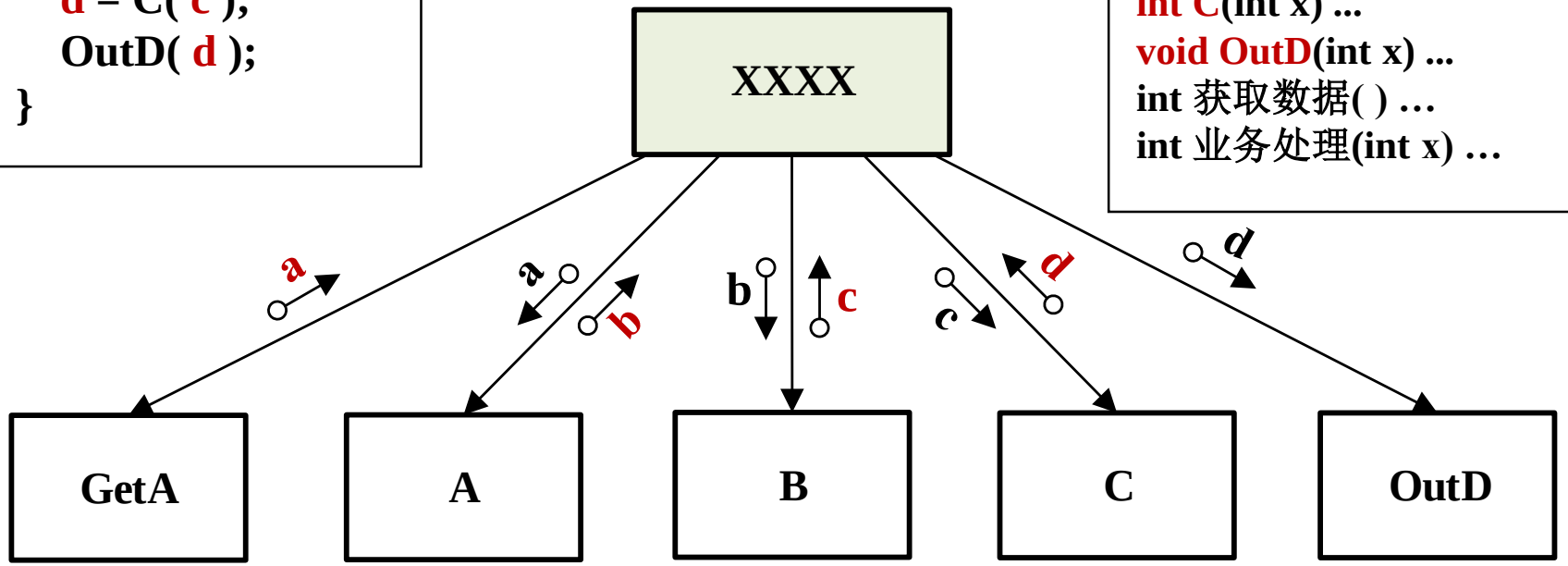
1.简单变换型DFD→系统结构图转换

```
/*系统控制主结构*/
main()
 { int a, b, c, d;
    a = GetA( );
    b = A( a );
    c = B( b );
    d = C( c );
    OutD( d );
 }
```



考虑扇入扇出度的问题，

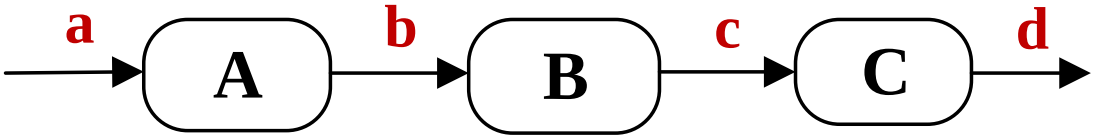
```
/*其他函数定义*/
int GetA( );
 {   int x;
     x = 获取数据( );
     return x;
 }
int A(int x);
 {  inty;
     y = 业务处理(x);
    return y;
 }
int B(int x) ...
int C(int x) ...
void OutD(int x) ...
int 获取数据( ) …
int 业务处理(int x) …
```



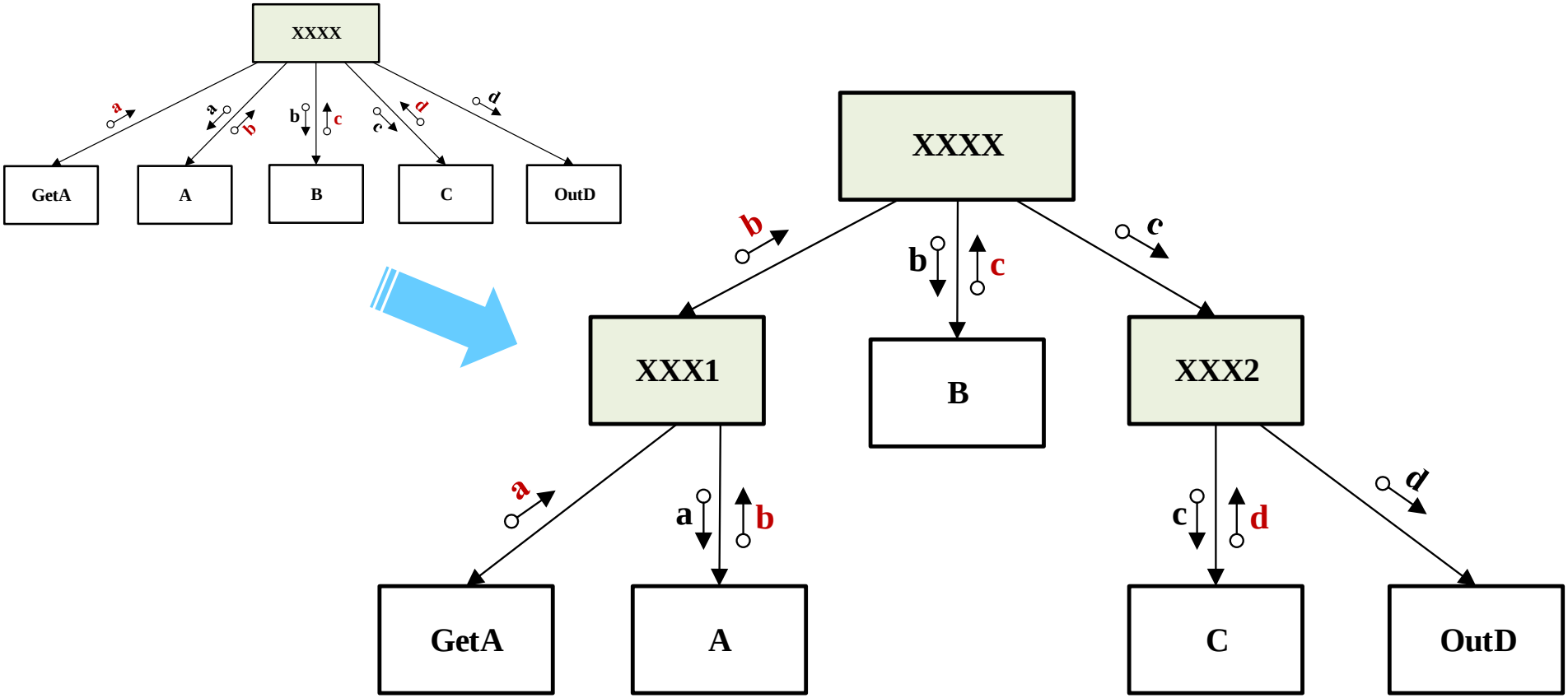
结构化的系统设计

DFD到系统结构图转换的基本模式

1.简单变换型DFD→系统结构图转换



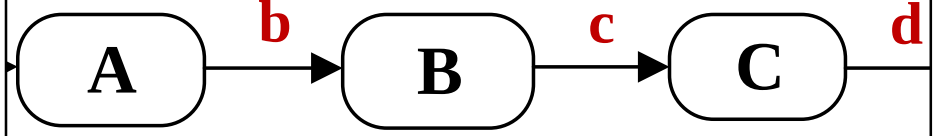
第2种情况：虑扇出度≤3时，得到系统结构图



结构化的系统设计

结构图转换的基本模式

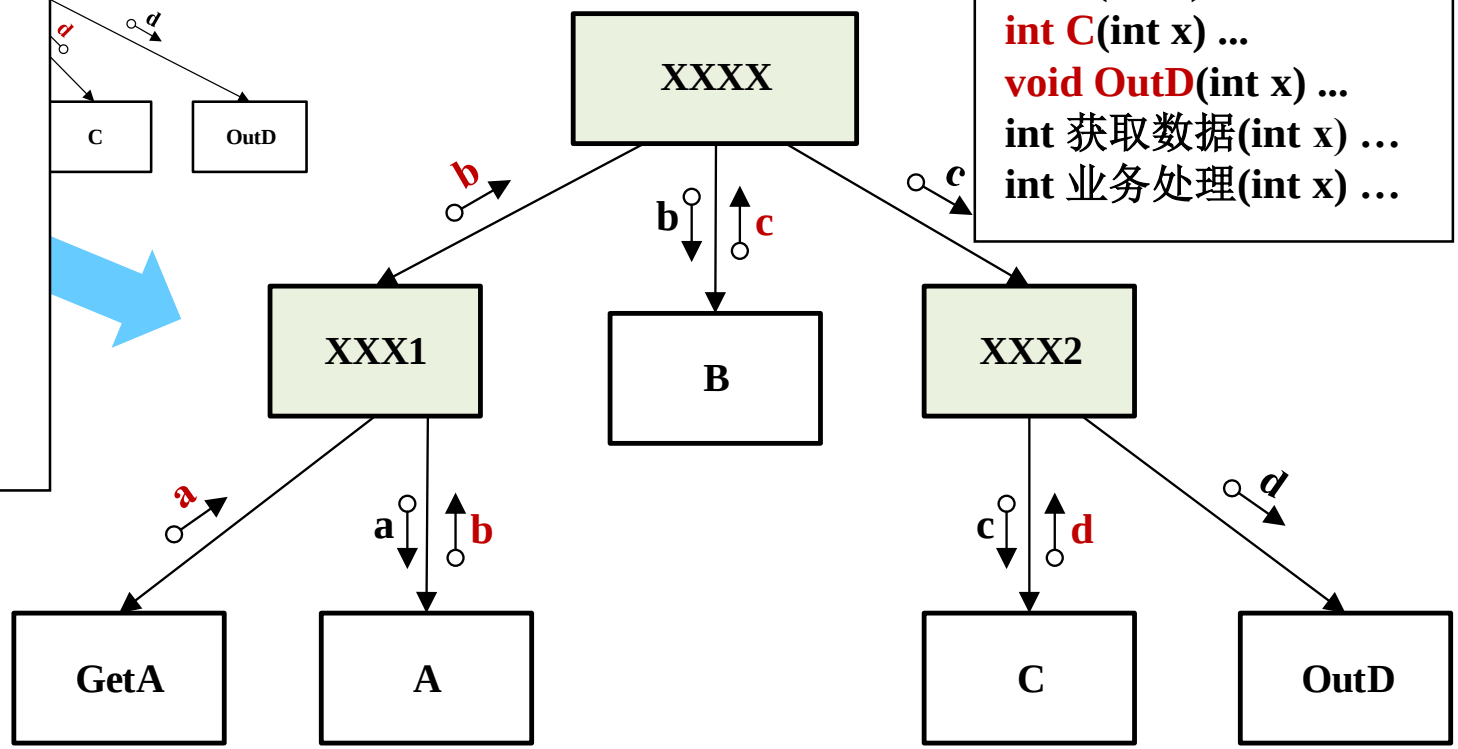
DFD→系统结构图转换



考虑扇出度≤3时，得到系统结构

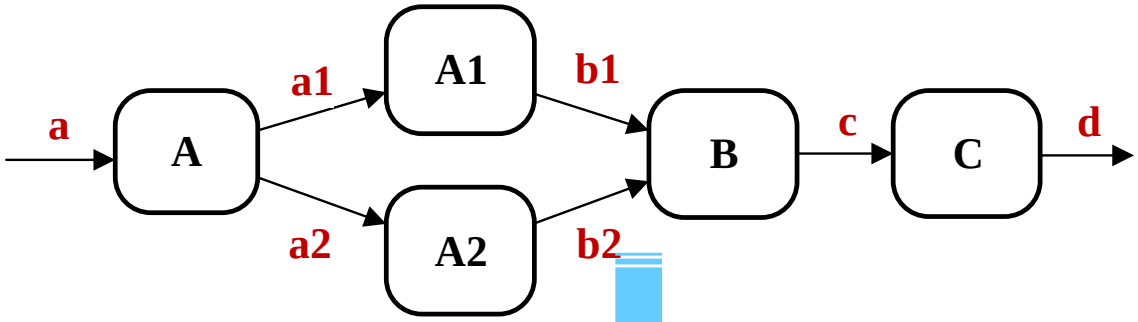
```
/*系统控制主结构*/
main()
 {  int b, c;
    b = XXX1( );
    c = B( b );
    XXX2( c );
 }
/*系统控制子结构*/
int XXX1( )
 {  int a, b;
    a = GetA( );
    b = A( a );
    return b;
 }
void XXX2( int x )
 {  int d;
    d = C( x );
    OutD( d );
 }
```

```
/*其他函数定义*/
int GetA( )
  {  int x;
     x = 获取数据( );
     return x;
  }
int A(int x)
  {  inty;
     y = 业务处理( x );
     return y;
  }
int B(int x) ...
int C(int x) ...
void OutD(int x) ...
int 获取数据(int x) …
int 业务处理(int x) …
```

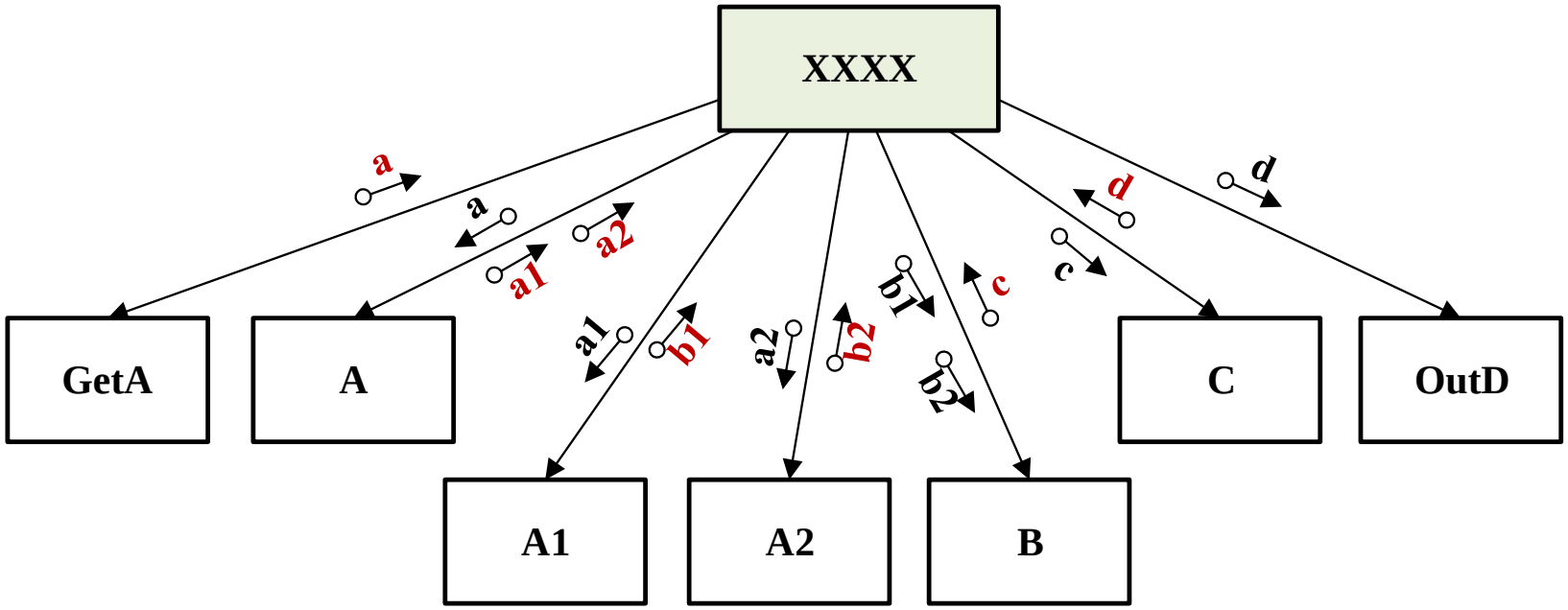


DFD到系统结构图转换的基本模式

2.复杂变换型DFD→系统结构图转换



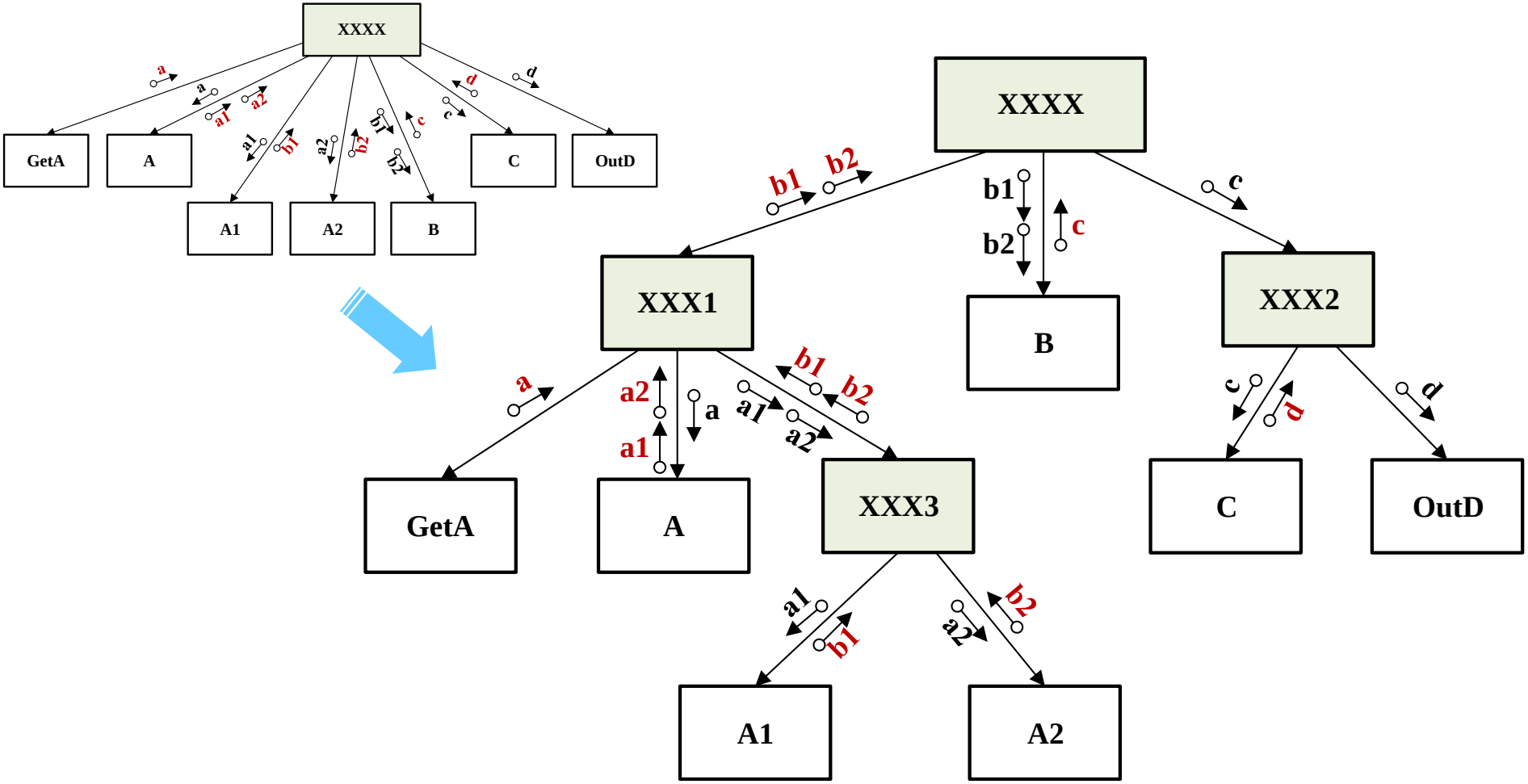
第1种情况：不考虑扇出度得到系统结构图



DFD到系统结构图转换的基本模式

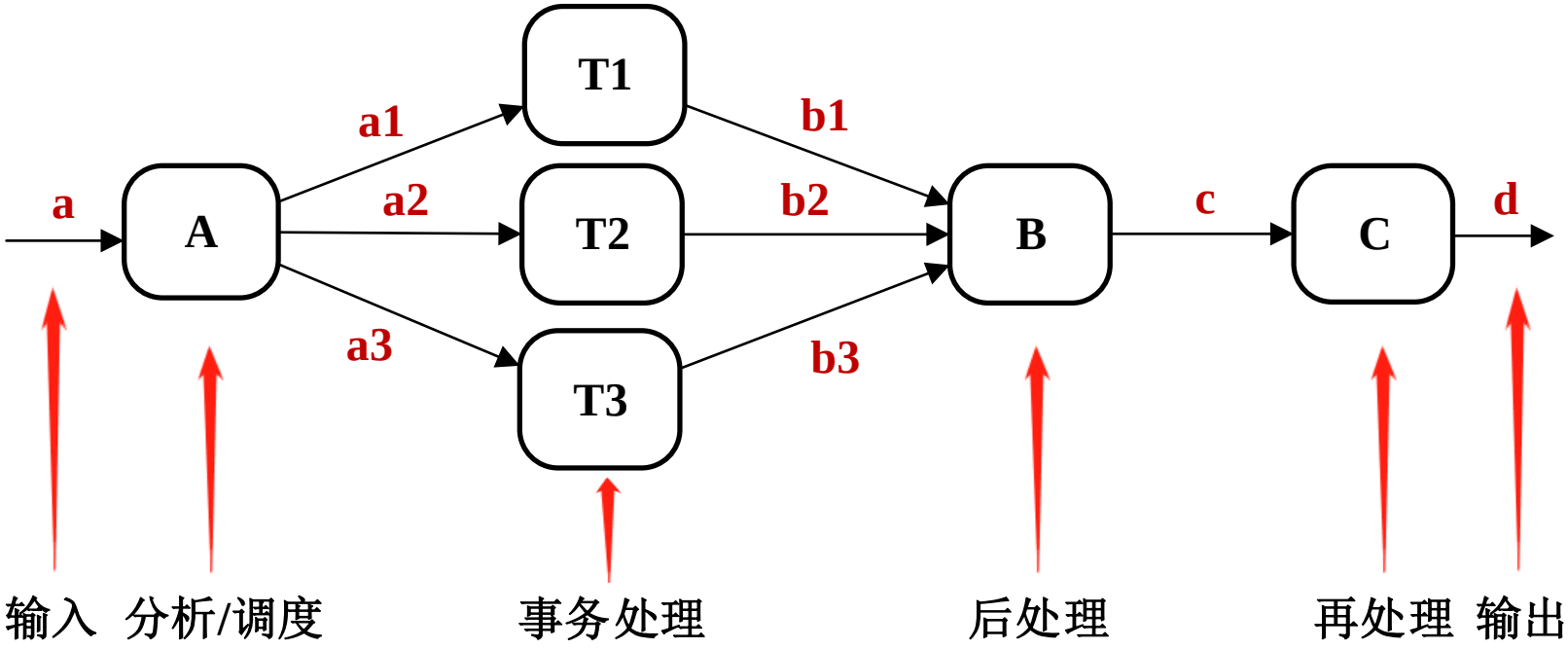
2.复杂变换型DFD→系统结构图转换

第2种情况：考虑扇出度≤3时得到系统结构图



DFD到系统结构图转换的基本模式

3.事务型DFD→系统结构图转换

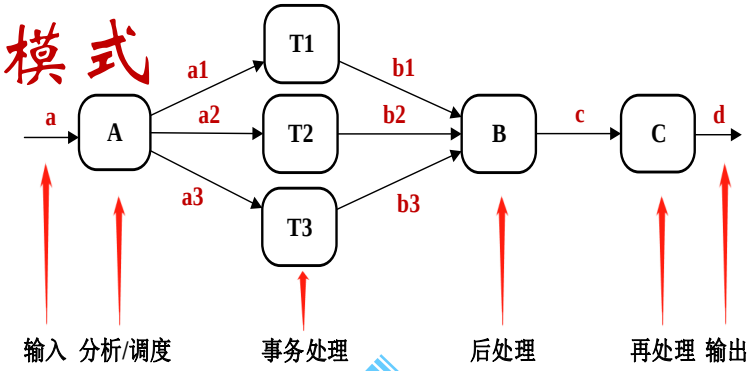


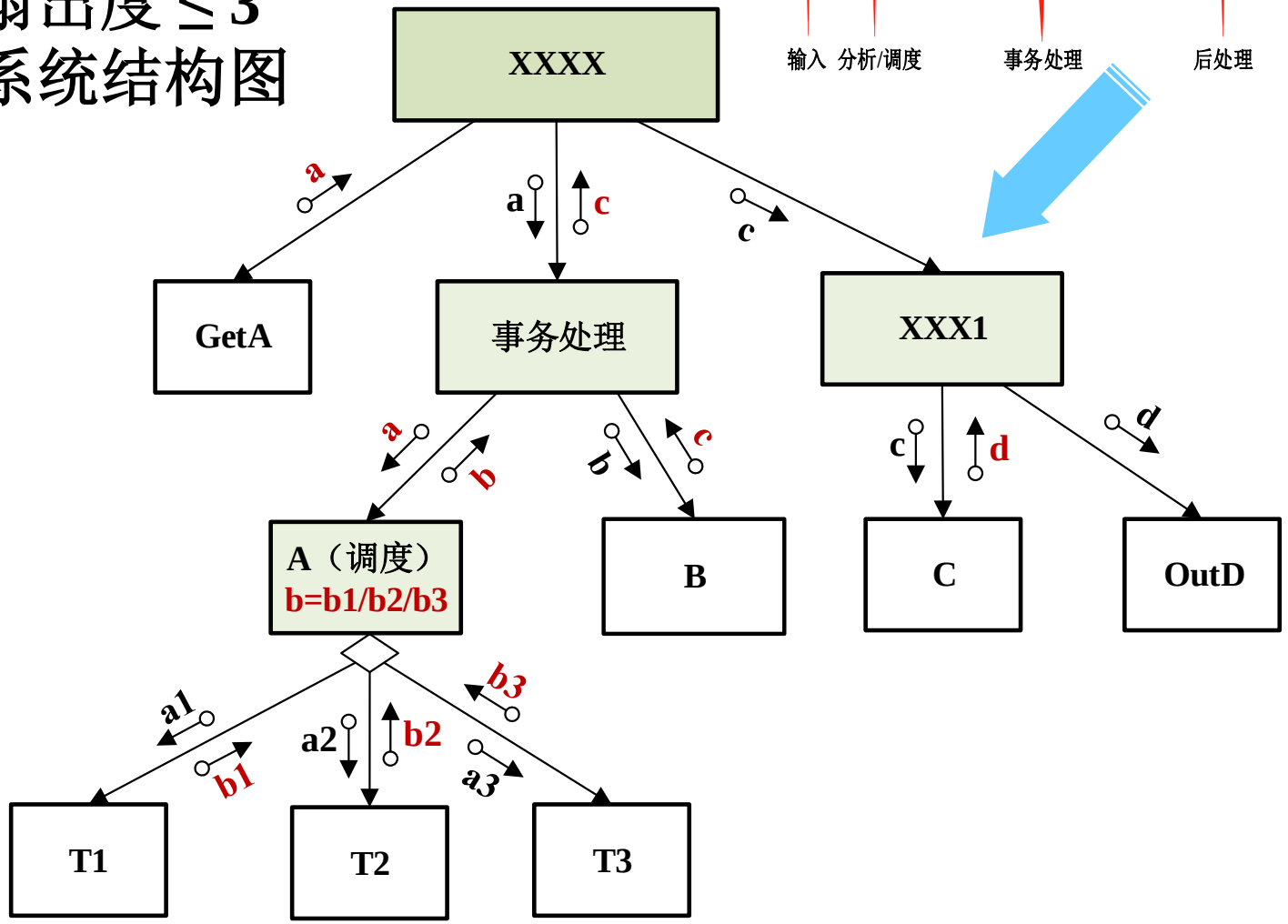
结构化的系统设计

DFD到系统结构图转换的基本模式

3.事务型DFD→系统结构图转换

考虑扇出度≤3
得到系统结构图

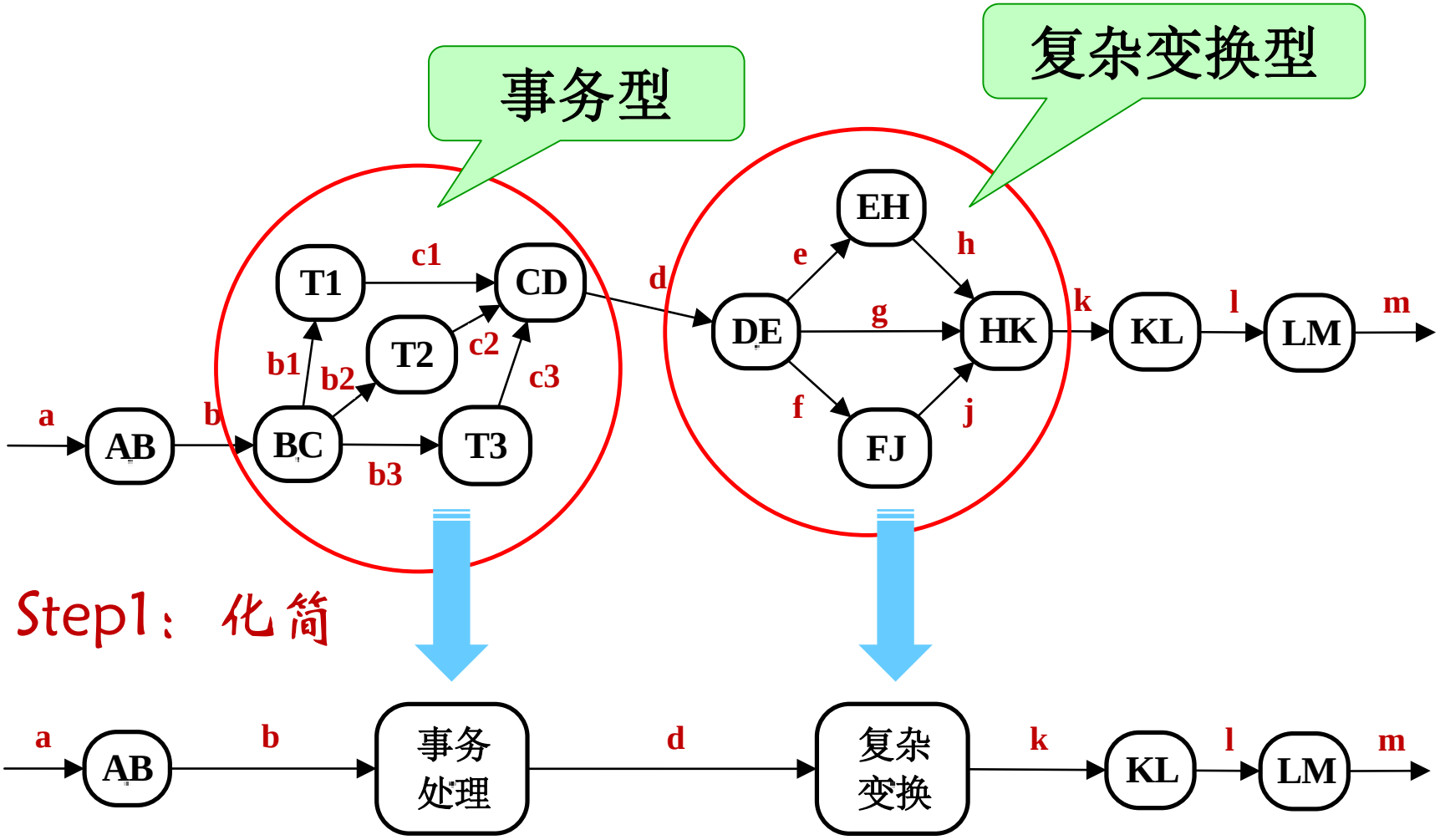




结构化的系统设计

DFD到系统结构图转换的基本模式

4.事务型与变换型混合复杂DFD→结构图转换

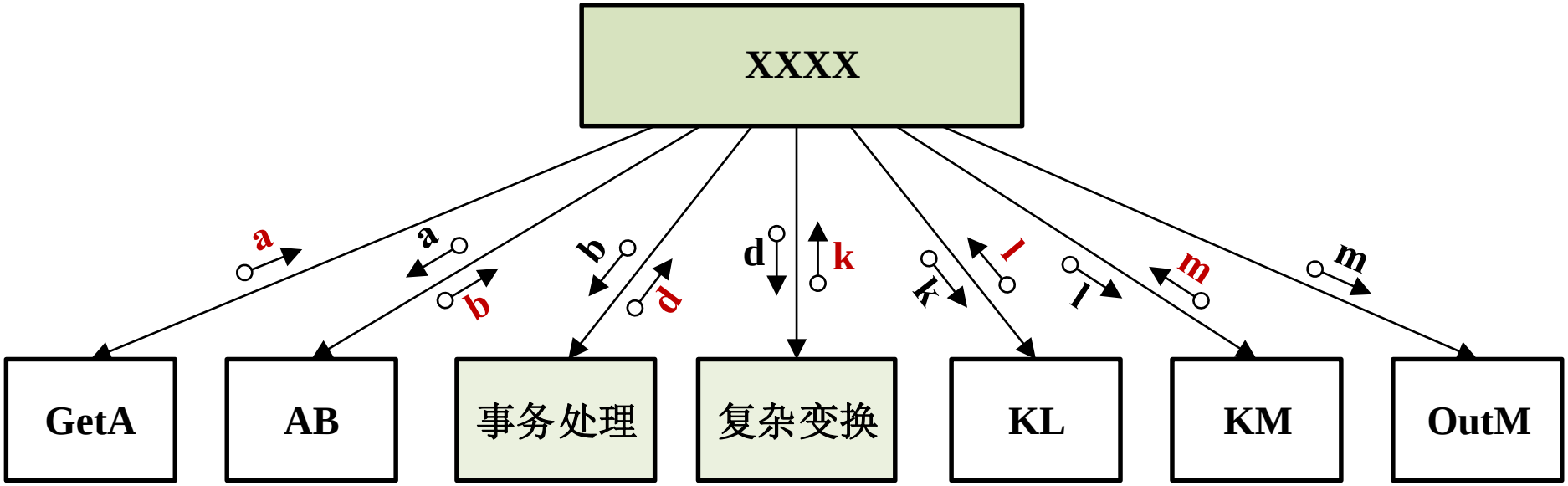


结构化的系统设计

DFD到系统结构图转换的基本模式

4.事务型与变换型混合复杂DFD→结构图转换

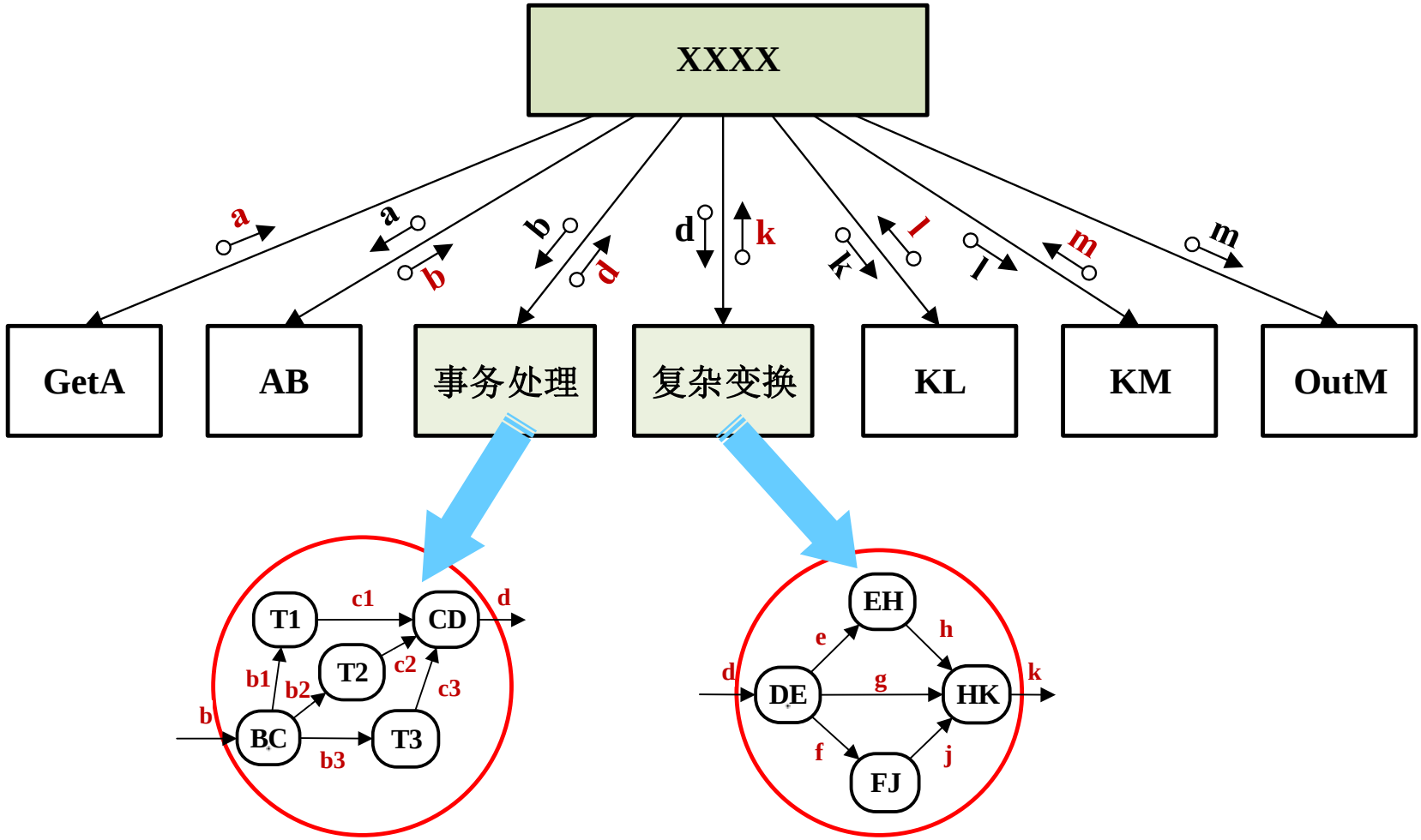
Step2：将化简的DFD→结构图转换



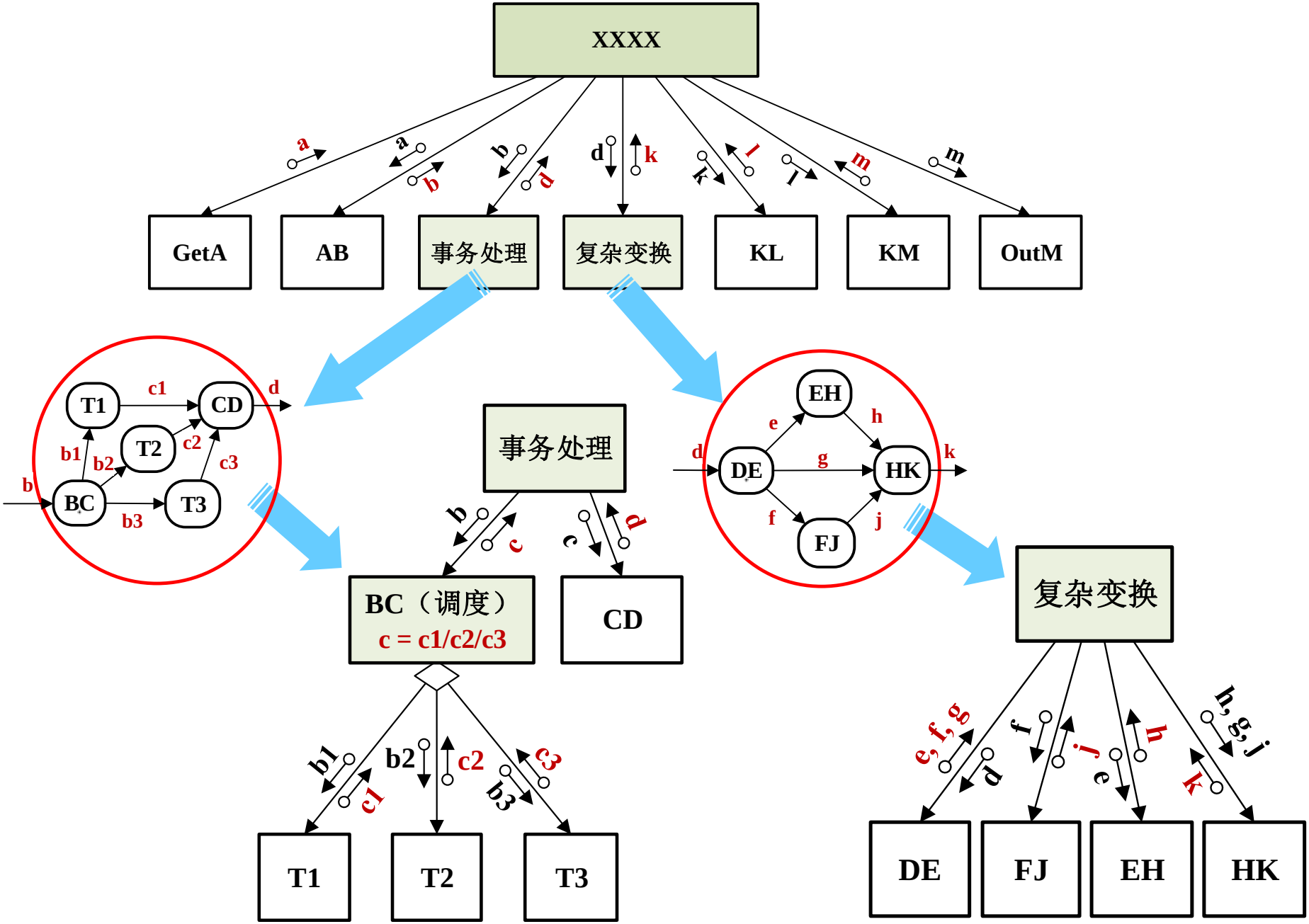
DFD到系统结构图转换的基本模式

4.事务型与变换型混合复杂DFD→结构图转换

Step3：拓展事务处理和复杂变换模块的结构图



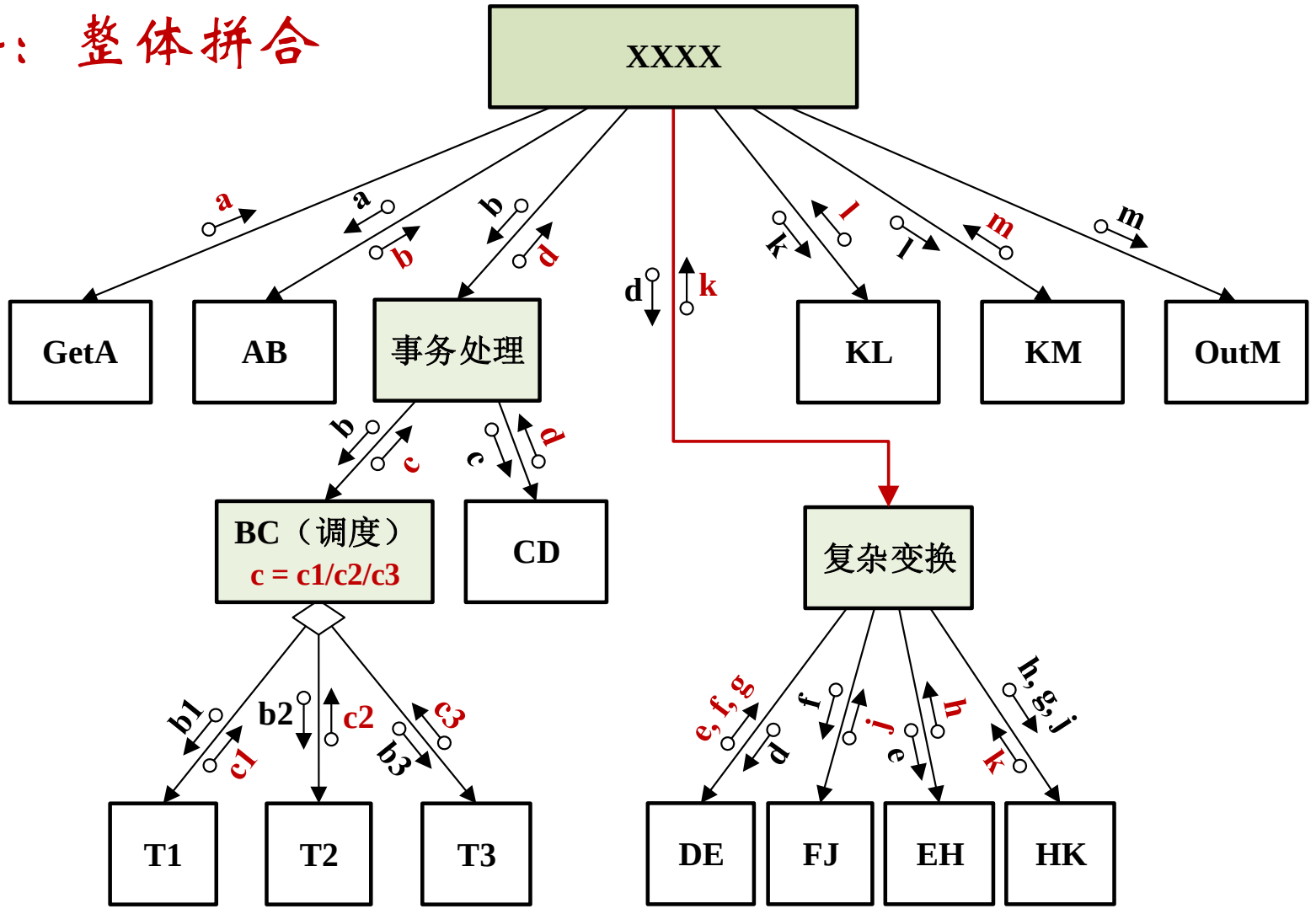
Step3：拓展事务处理和复杂变换模块的结构图



DFD到系统结构图转换的基本模式

4.事务型与变换型混合复杂DFD→结构图转换

Step4：整体拼合



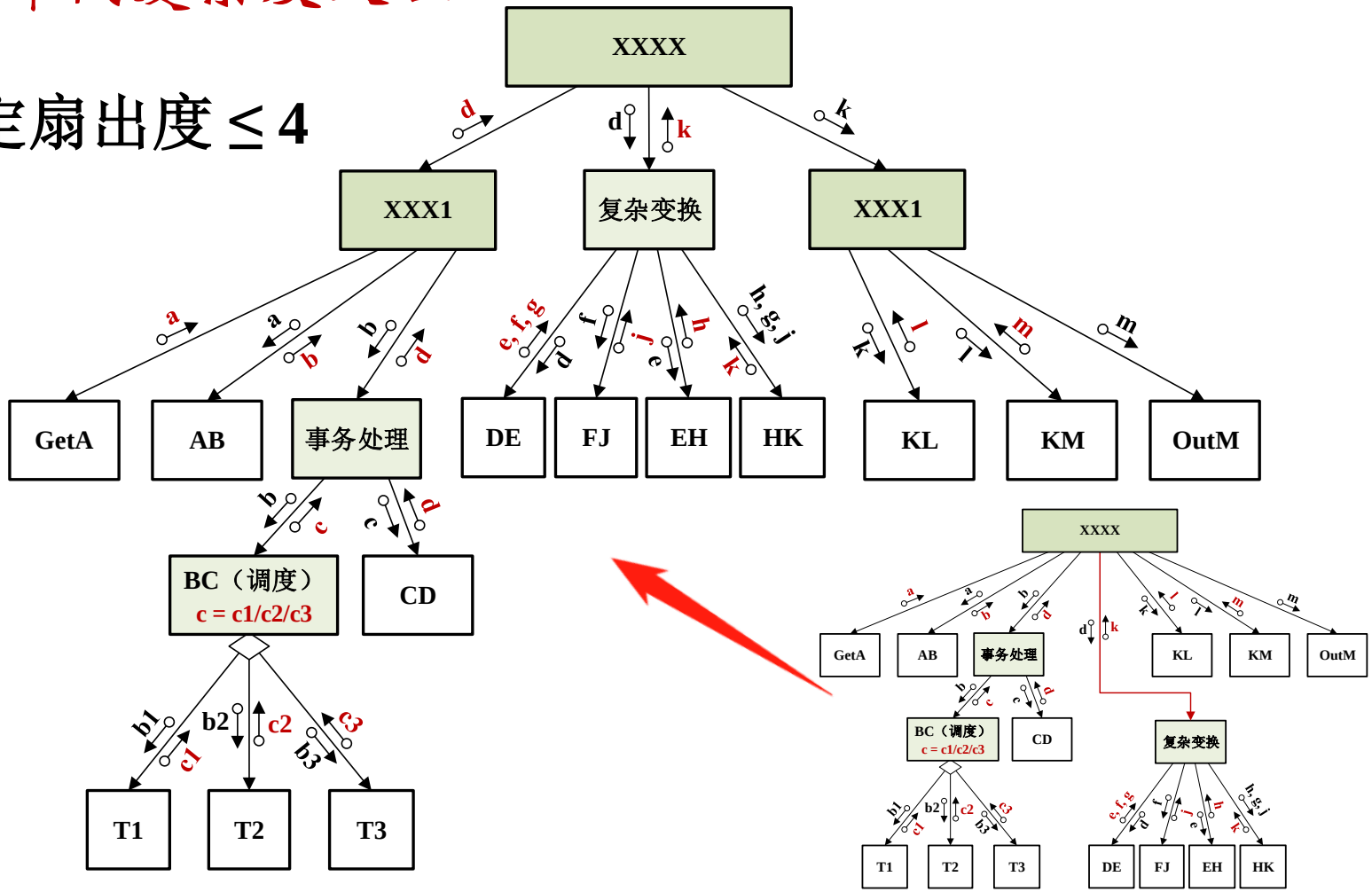
结构化的系统设计

DFD到系统结构图转换的基本模式

4.事务型与变换型混合复杂DFD→结构图转换

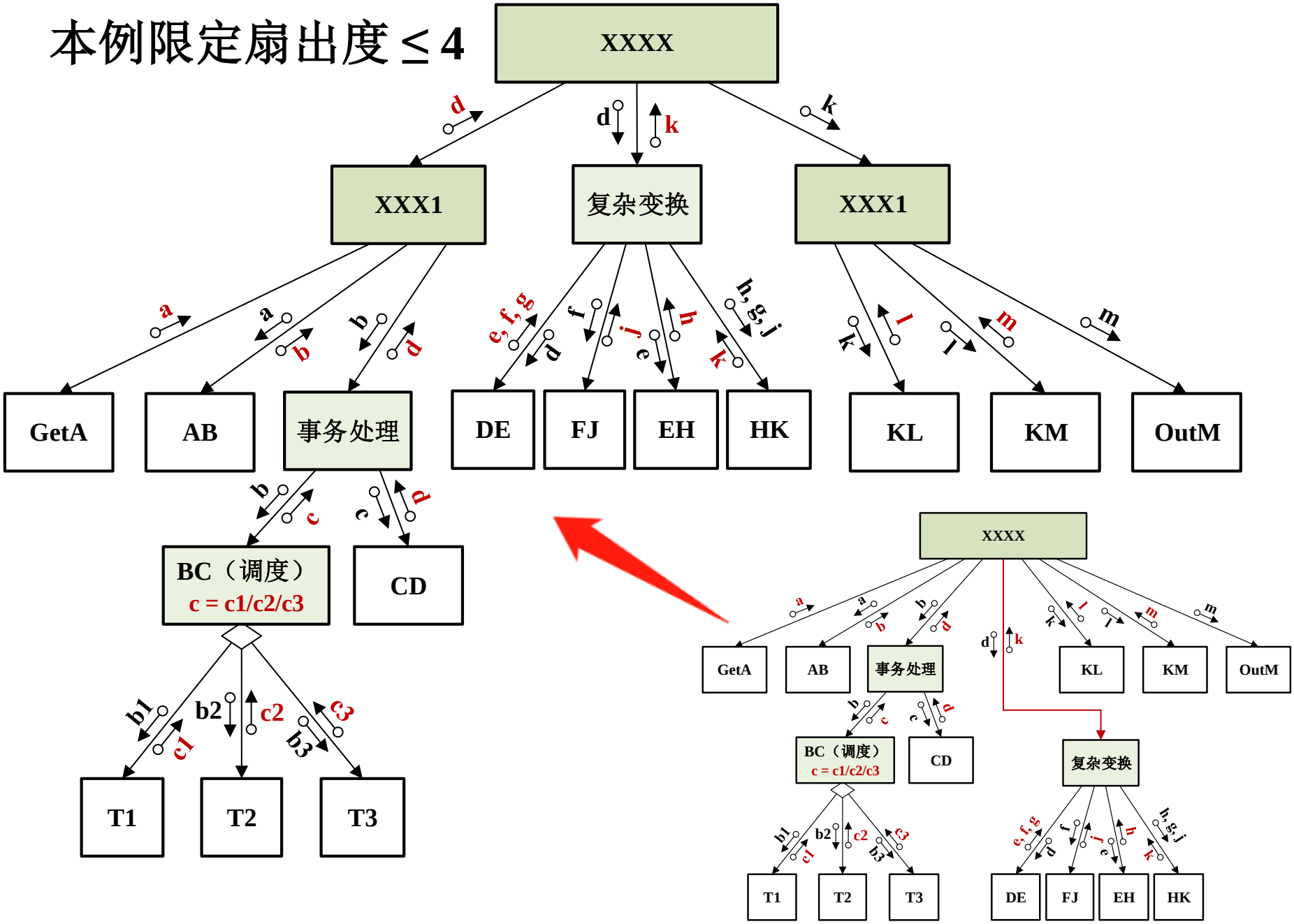
Step5：降低复杂度处理

本例限定扇出度≤4



Step5：降低复杂度处理

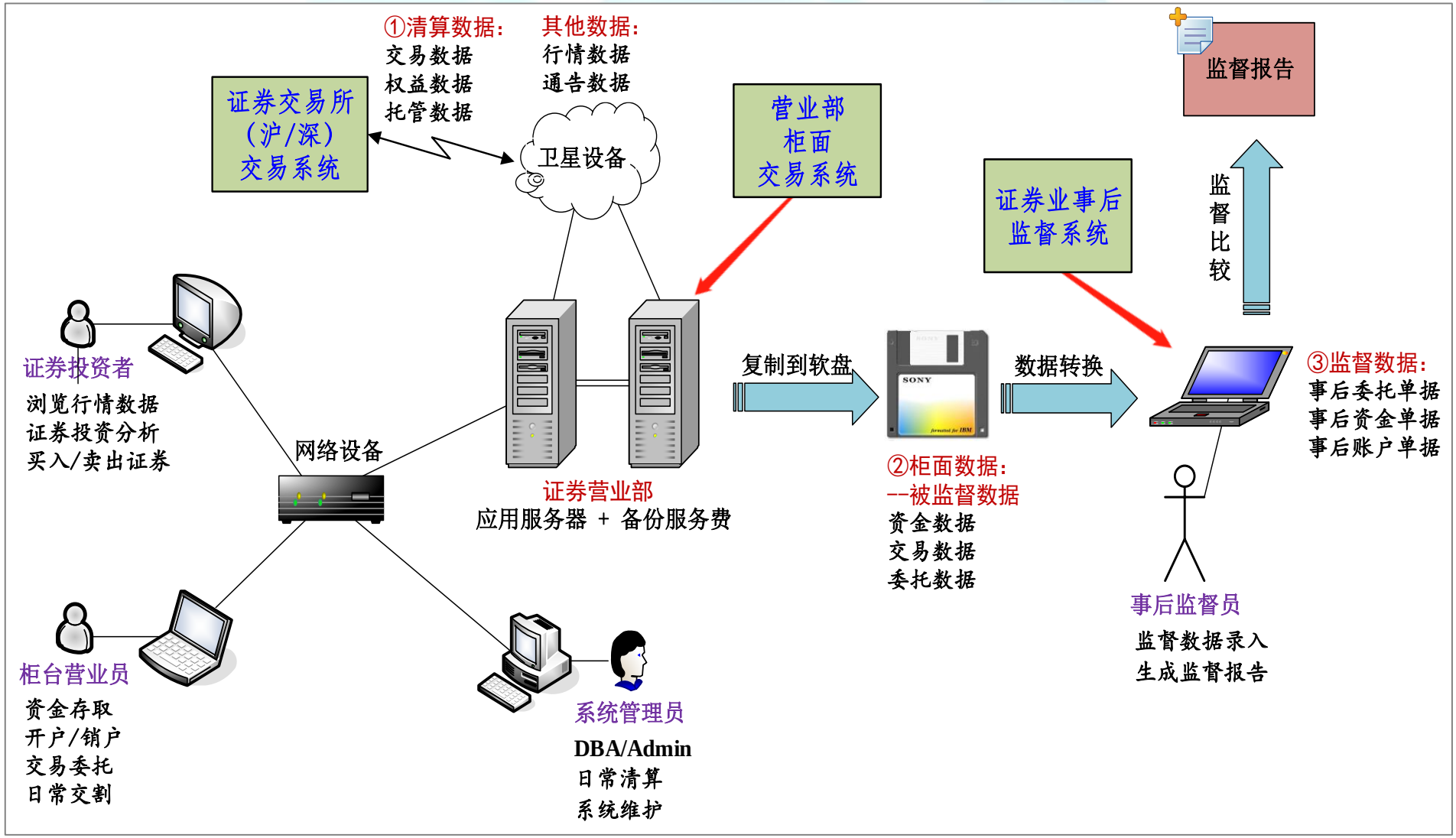
本例限定扇出度≤4



结构化的系统设计

结构化分析与设计实例

《证券业事后监督系统》需求介绍



结构化分析与设计实例

公司负责人愿景 即通过事后监督系统能够：

① 及时发现日常工作差错

② 防止营业部内部违法犯罪

事后监督系统目标（愿景分解）：

① 监督营业部日常资金存取安全

② 监督证券交易环节准确无误、无违规操作

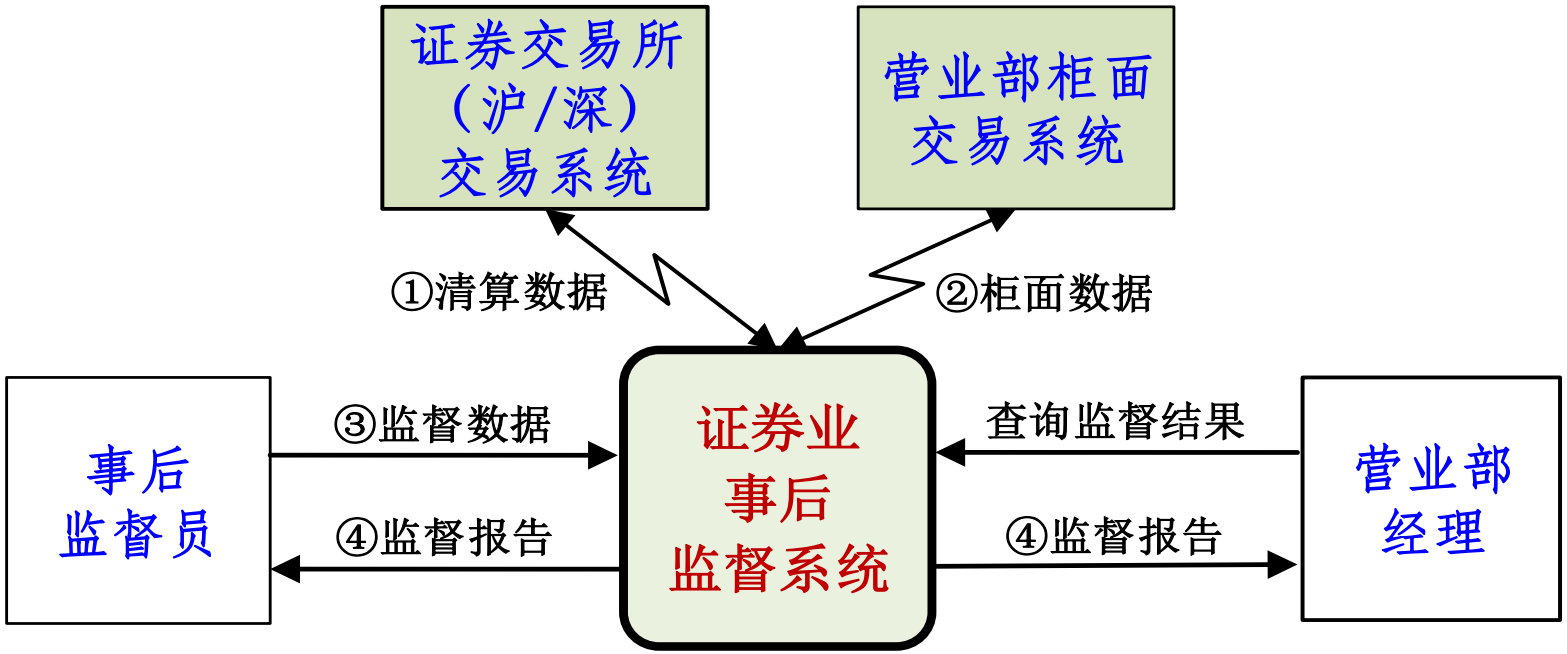
③ 监督营业部清算环节无差错、无违法犯罪

差　　错： 资金存取、交易委托、清算数据等

违规违法： 资金透支、资金转移、透支交易、中签转移、计息克扣、证券余额转移、权益转移等

结构化分析与设计实例

《证券业事后监督系统》顶层DFD



数据流说明：

①清算数据：
- 交易数据
- 权益数据
- 托管数据

②柜面数据：
- 资金数据
- 交易数据
- 委托数据

③监督数据：
- 事后委托单据
- 事后资金单据
- 事后账户单据

④监督报告：
- 比较类报告
- 自查类报告
- 清算类报告

结构化分析与设计实例

《证券业事后监督系统》分析与设计

01-SHJD数据逻辑流程 数据流图DFD建立

02-SHJD数据结构说明 数据字典DD建立

03-SHJD功能模块分布表 结构图建立

04-SHJD数据库关系表 实体关系图ERD建立

05-SHJD文件-目录分布图 部署设计