数据流图的画法

单世民

概述

- 和其他众多的软件设计图一样，数据流图是软件设计中一个不可缺少的辅助工具。通过数据流图，软件设计者可以更有效地进行设计，更好地捕获用户需求。

- 数据流图用抽象模型的概念，按照软件内部数据传递、变换的关系，自顶向下逐层分解，找到满足功能要求的所有可实现的软件。

- 数据流图以分层的形式反映结构关系，清楚地表达和容易理解了整个系统。

数据流图的定义

- 数据流图：描绘系统的逻辑模型，只描绘数据流在系统中流动和处理的情况，是逻辑系统的图形表示。

  - 抽象：数据流图中去掉了具体的组织机构、工作场所、物质流等，只剩下信息和数据存储、流动、使用以及处理等抽象数据。

  - 概括：把系统对各种业务的处理过程联系起来考虑，形成一个总体，可反映出数据流之间的概括情况

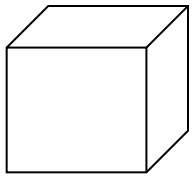
数据流图的作用

- 系统设计：
  - 自顶而下地分析系统的信息流程
  - 在图上确定需要计算机处理的部分
  - 向数据库设计过渡
  - 根据数据流向确定存取方式
  - 能够确定一个处理过程。

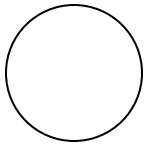
- 软件测试：
  - 帮助程序员查找到错误的发生位置。

数据流图符号

数据的源点/终点



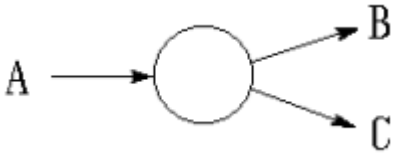
变换数据的处理



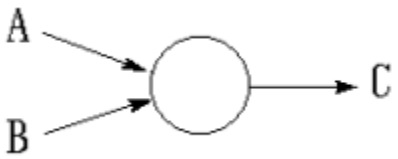
数据存储

数据流

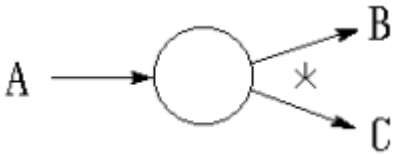
数据流图符号



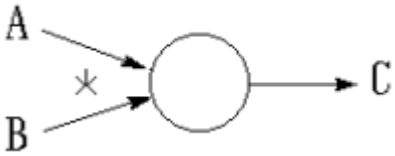
有A就有B或C，
或者两者都有



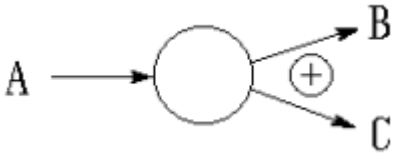
当A和B由一个
存在就有C



有A就有B和C，
两者同时有



当A与B都存在
就有C



有A就有B或C，但
不会同时有B和C

数据流图的四种组成成分

- 数据的源点或数据终点
  - 数据源和终点表示数据的外部来源和去处。它通常是系统之外的人员或组织，不受系统控制。
- 数据处理（加工）
  - 数据处理处理是对数据进行的操作，它把流入的数据流转换为流出的数据流。
- 数据存储（文件）
  - 数据存储是存贮数据的工具，数据存储名应与它的内容一致。
- 数据流
  - 数据流由一组确定的数据组成。
  - 数据流可以从处理流向处理，也可以从处理流进、流出数据存储，还可以从源点流向处理或从处理流向终点。

如何画数据流图

- 一般情况下，应该遵守“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，先画数据处理的输入和输出，再画数据处理内部。

1. 从问题描述中**取出 4 种基本组成成分**
2. 根据（1）的结果画出**系统的基本系统流图**（顶层图）
3. 把由（2）得到的基本系统模型细化为**系统的功能级数据流图**
4. 对功能级数据流图中的主要功能**进一步细化**，直至满意为止

如何画数据流图

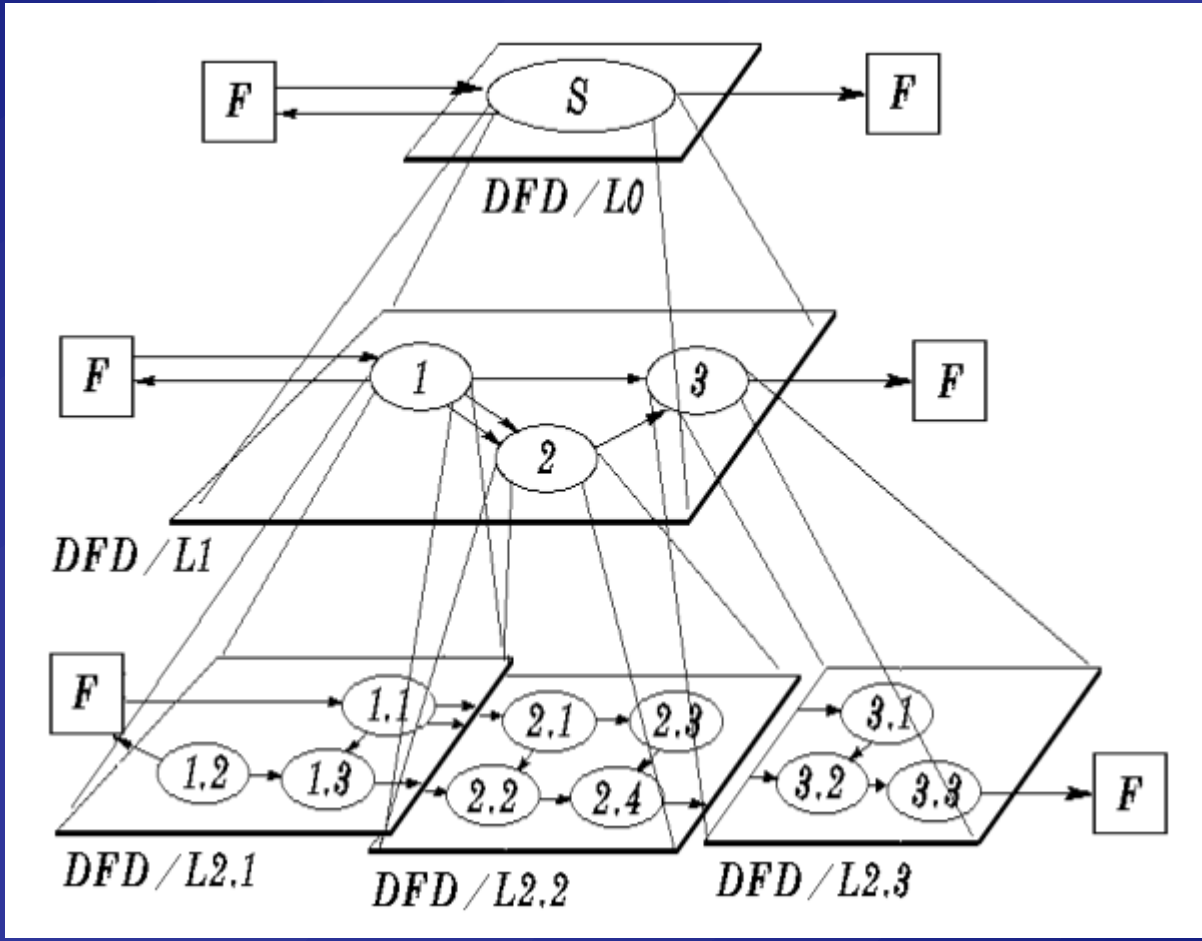
1. 从问题描述中**取出 4 种基本组成成分**
   - 不能混淆了数据流与数据处理、数据存储与数据源或终点

2. 根据（1）的结果画出**系统的基本系统流图**（顶层图）
   - **确定系统边界**，在系统分析初期，系统的功能需求等还不很明确，为了防止遗漏，不妨先将确定的系统边界范围定得大一些。
   - 系统边界确定后，那么越过边界的数据流就是系统的输入或输出，将输入与输出用数据处理符号连接起来,并加上输入数据来源和输出数据去向就形成了顶层图。

3. 把由（2）得到的基本系统模型细化为**系统的功能级数据流图**
   - 从系统输入端到输出端，逐步用数据流和数据处理连接起来，**当数据流的组成或值发生变化时**，就在该处画一个“数据处理”符号。
   - 数据流图时还应同时**画上数据存储**
   - 最后检查系统的边界，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。

4. 对功能级数据流图中的主要功能**进一步细化**，直至满意为止
   - 针对每一个数据处理进行分析，**如果在该数据处理内部还有数据流，则可将该数据处理分成若干个子数据处理**，并用一些数据流把子数据处理联接起来

如何画数据流图


F
S
F
DFD / L0
F
1
3
F
2
DFD / L1
F
1.1
2.1
2.3
3.1
1.2
1.3
2.2
2.4
3.2
3.3
F
DFD / L2.1
DFD / L2.2
DFD / L2.3

注意事项

- 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，先画数据处理的输入和输出，再画数据处理的内部。

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数据处理，并同时加上数据存储。
3. 数据处理的分解“由外向里”进行分解
4. 数据流、数据存储、数据处理的命名和文字说明要确切，能反映整体。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

注意事项

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。
7. 在具体画图过程中，按照从左至右画图，通常左侧、右侧分别是数据源和终点，中间是一系列数据处理和文件此外，数据流图中各种符号布置要合理，分布应均匀。
8. **是画数据流图而不是画程序框图！**
   - 程序框图是从对数据进行加工的角度描述系统的，其箭头是控制流，表示的是对数据进行加工的次序
   - 数据流图则是从数据的角度来描述系统的，其箭头是数据流，表示的是数据的流动方向

注意事项

- 父图与子图间平衡
  - 这里的平衡指的是子图的输入、输出数据流必须与父图中对应数据处理的输入、输出数据流相同。
  - 两种情况是允许的
    - 子图的输入/输出流比父图中相应数据处理的输入/输出流表达得更细。
    - 忽略枝节性的数据流。

注意事项

- 数据流图的分解
  - 经验表明，一般说来**一个数据处理每次分解量最多不要超过七个**为宜。
  - 分解时应遵循原则
    1. 分解应自然,概念上要合理、清晰。
    2. **上层可分解的快些**（即分解成的子数据处理个数多些），这是因为上层是综合性描述，对可读性的影响小。而**下层应分解得慢些**。
    3. 在不影响可读性的前提下,应适当地多分解成几部分,以**减少分解层数**。
    4. 一般说来，当数据处理可用一页纸明确地表述时，或数据处理只有单一输入/输出数据流时（出错处理不包括在内），就应停止对该数据处理的分解。另外，对数据流图中不再作分解的数据处理（即功能单元），必须作出详细的数据处理说明，并且每个数据处理说明的编号必须与功能单元的编号一致。

注意事项

- 数据流中的编号
  - 分层数据流图的顶层称为 0 层,称它是第 1 层的父图，而第 1 层既是0 层图的子图，又是第 2 层图的父图，依此类推。
  - 由于父图中有的数据处理可能就是功能单元，不能再分解，因此父图拥有的子图数少于或等于父图中的数据处理个数。
  - 数据流图中的数据处理编号规则
    - 子图中的编号为父图号和子数据处理的编号组成
    - 子图的父图号就是父图中相应数据处理的编号

注意事项

- 数据处理转换的方式
  - 改变数据的结构，例如将数组中各数据重新排序；
  - 产生新的数据，例如对原来的数据总计、求平均等值。

注意事项

- 数据存储的画法
  - 从数据存储流入或流出数据流时，数据流方向是很重要的。
    - 如果是读数据存储，则数据流的方向应从数据存储流出，写数据存储时则相反；
    - 如果是又读又写，则数据流是双向的。

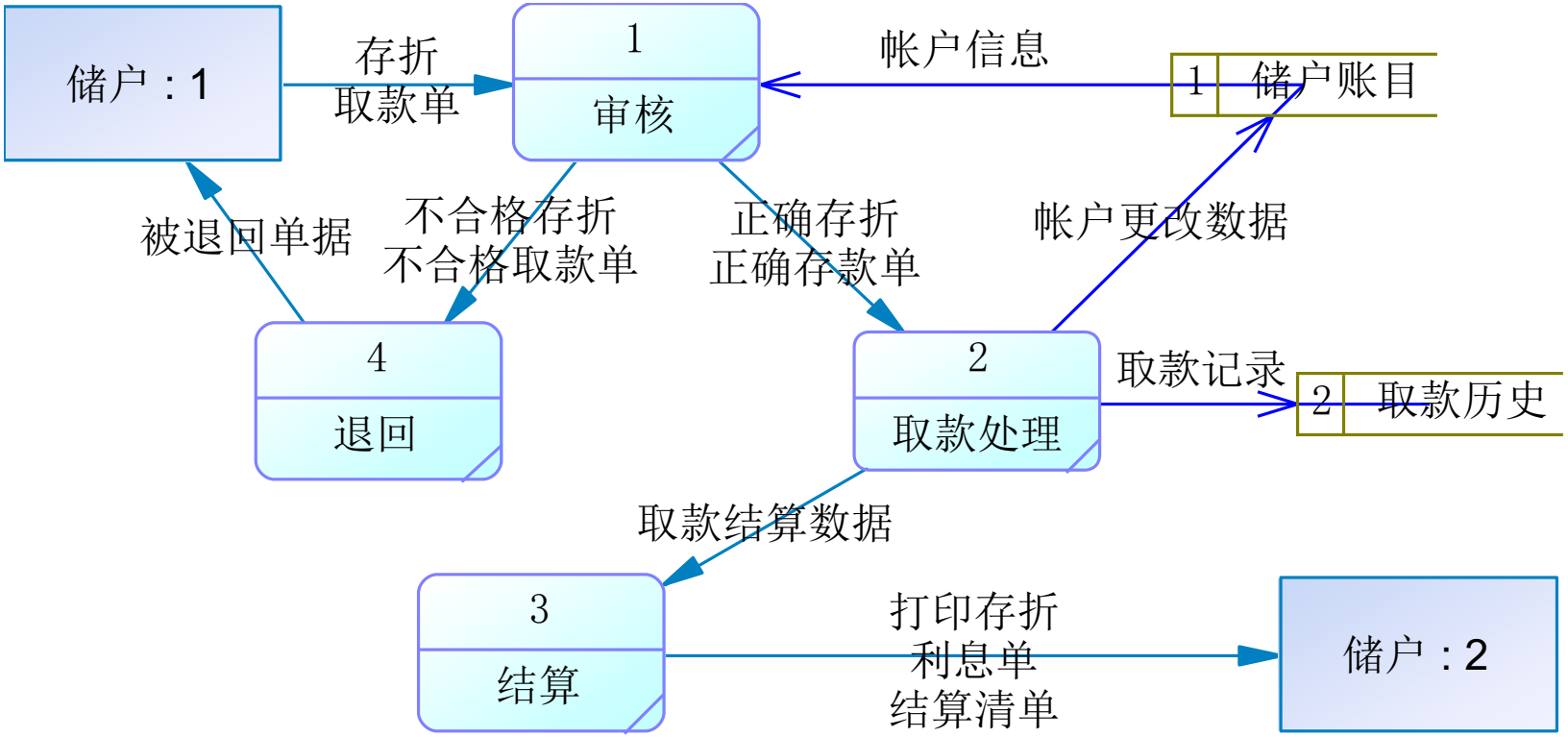
注意事项

- 数据源或终点的画法
  - 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。同一个源点、终点或文件均可在不同位置多次出现，这时要在源（终）点符号的右下方画小斜线，或在文件符号左边画竖线，以示重复。

练习

- 银行取款处理
  - ¤ 储户将填好的取款单、存折交银行
  - ¤ 银行做如下处理：
    1. 审核帐目，将不合格的存折、取款单退回储户，合格的存折、取款单送取款处理。
    2. 取款处理修改帐目，将存折、利息单、结算清单及现金交储户，同时将取款单存档。

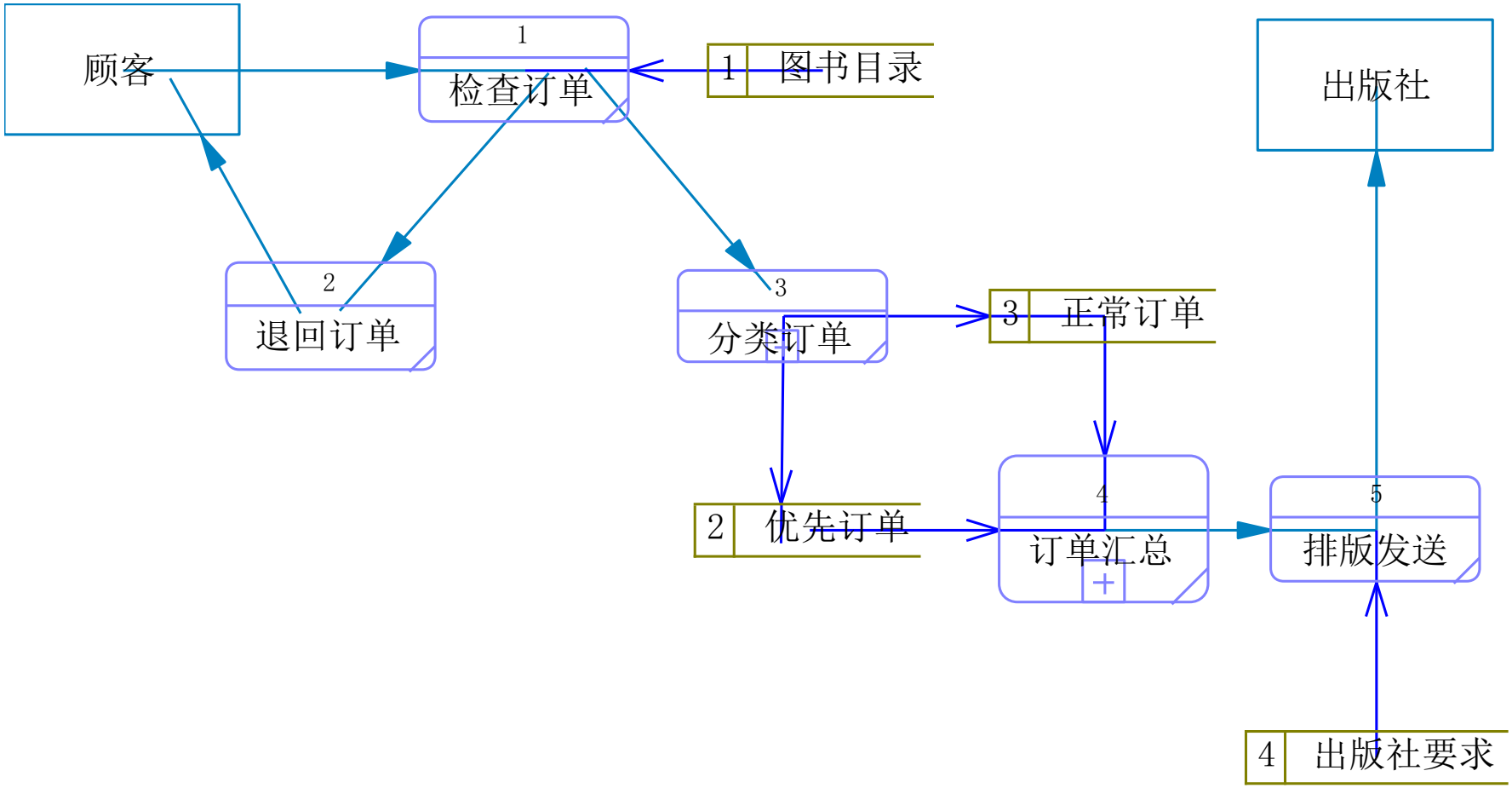
练习


储户：1
存折
取款单
1
审核
帐户信息
1
储户账目
被退回单据
不合格存折
不合格取款单
正确存折
正确存款单
帐户更改数据
4
退回
2
取款处理
取款记录
2
取款历史
取款结算数据
3
结算
打印存折
利息单
结算清单
储户：2

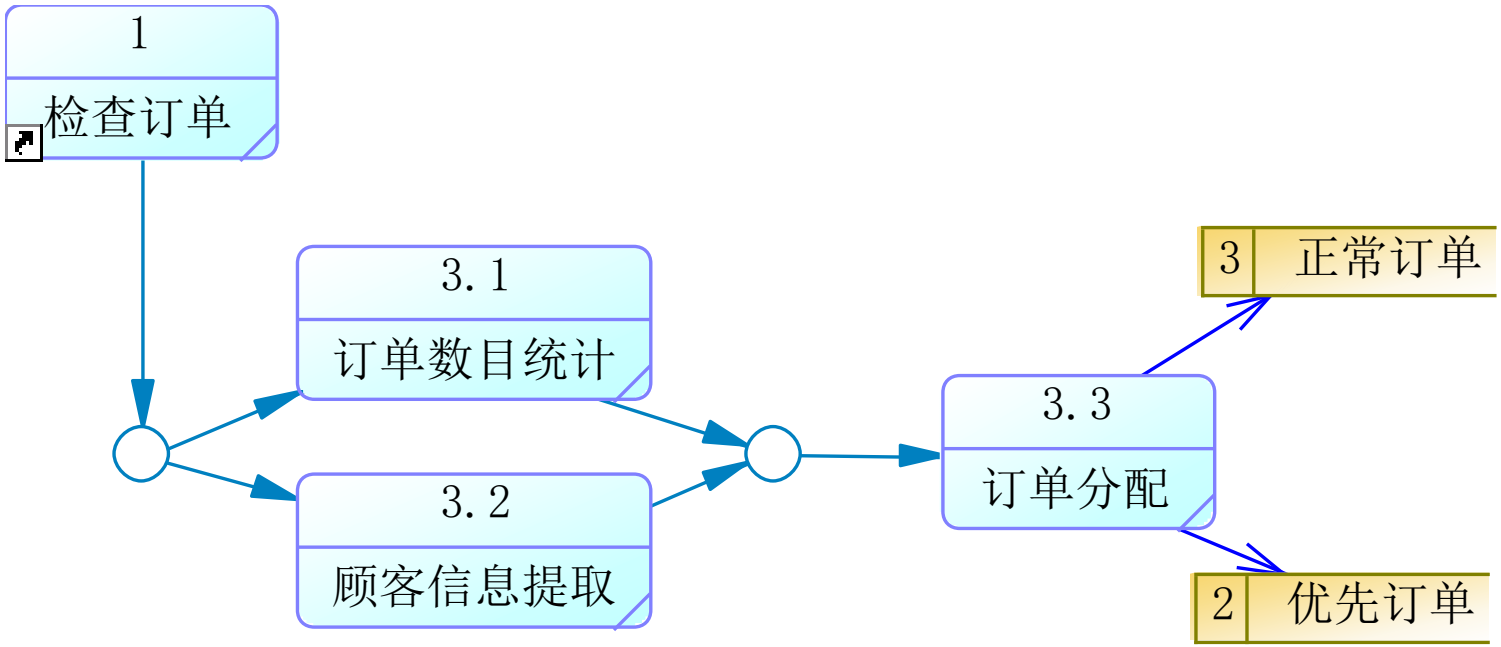
练习

- 图书预订系统
  - 书店向顾客发放订单，顾客将所填订单交由系统处理
  - 系统首先依据图书目录对订单进行检查并对合格订单进行处理，处理过程中根据顾客情况和订单数目将订单分为优先订单与正常订单两种，随时处理优先订单，定期处理正常订单。
  - 最后系统根据所处理的订单汇总，并按出版社要求发给出版社。

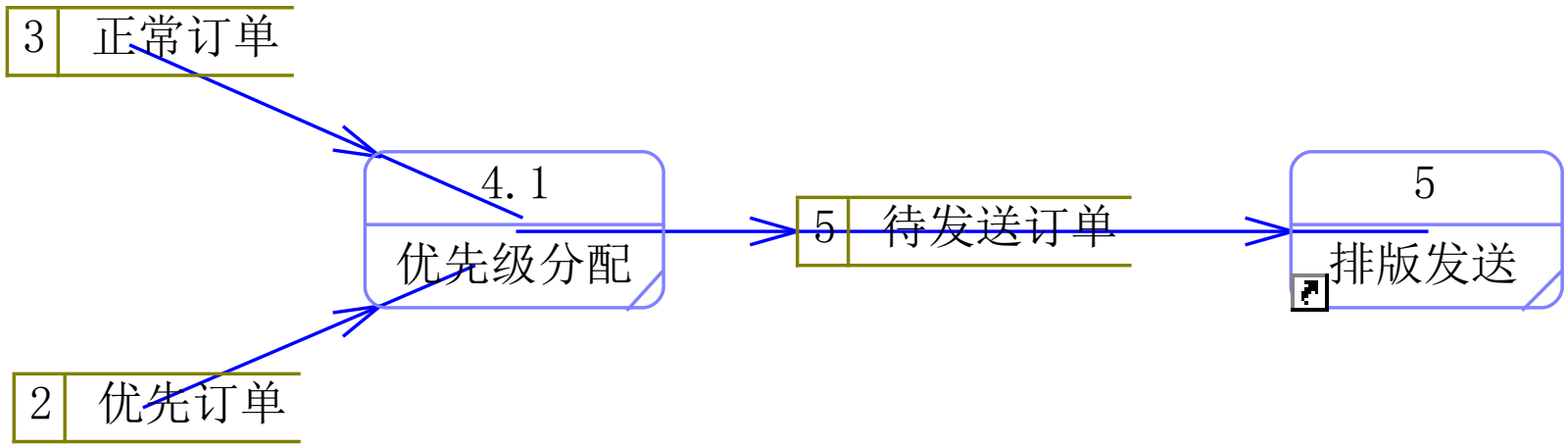
练习


顾客
1
检查订单
1 图书目录
出版社
2
退回订单
3
分类订单
3 正常订单
2 优先订单
4
订单汇总
5
排版发送
4 出版社要求

练习



练习



The End

Ssm

大连理工大学