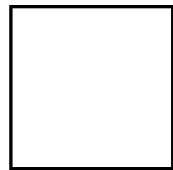


2.4 数据流图

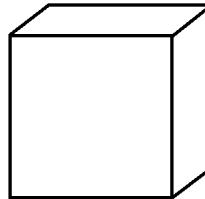
数据流图(DFD)是一种图形化技术，它描绘信息流和数据从输入移动到输出的过程中所经受的变换。

数据流图是系统逻辑功能的图形表示，完全不需要考虑怎样具体地实现这些功能，即使不是专业的计算机技术人员也容易理解它，因此是分析员与用户之间极好的通信工具。

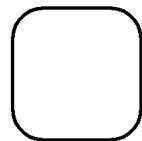
2.4.1 符号



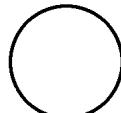
或



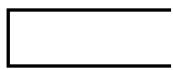
数据的源点 / 终点



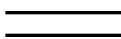
或



变换数据的处理



或



数据存储



数据流

(a)

处理并不一定是一个程序。一个处理框可以代表一系列程序、单个程序或者程序的一个模块；它甚至可以代表用穿孔机穿孔或目视检查数据正确性等人工处理过程。一个数据存储也并不等同于一个文件，它可以表示一个文件、文件的一部分、数据库的元素或记录的一部分等；数据可以存储在磁盘、磁带、磁鼓、主存、微缩胶片、穿孔卡片及其他任何介质上(包括人脑)。

数据存储和数据流都是数据，仅仅所处的状态不同。数据存储是处于静止状态的数据，数据流是处于运动中的数据。

通常在数据流图中忽略出错处理，也不包括诸如打开或关闭文件之类的内务处理。数据流图的基本要点是描绘“做什么”而不考虑“怎样做”。

2.4.2 例子

假设一家工厂的采购部每天需要一张定货报表，报表按零件编号排序，表中列出所有需要再次定货的零件。对于每个需要再次定货的零件应该列出下述数据：零件编号，零件名称，定货数量，目前价格，主要供应者，次要供应者。零件入库或出库称为事务，通过放在仓库中的CRT终端把事务报告给定货系统。当某种零件的库存数量少于库存量临界值时就应该再次定货。

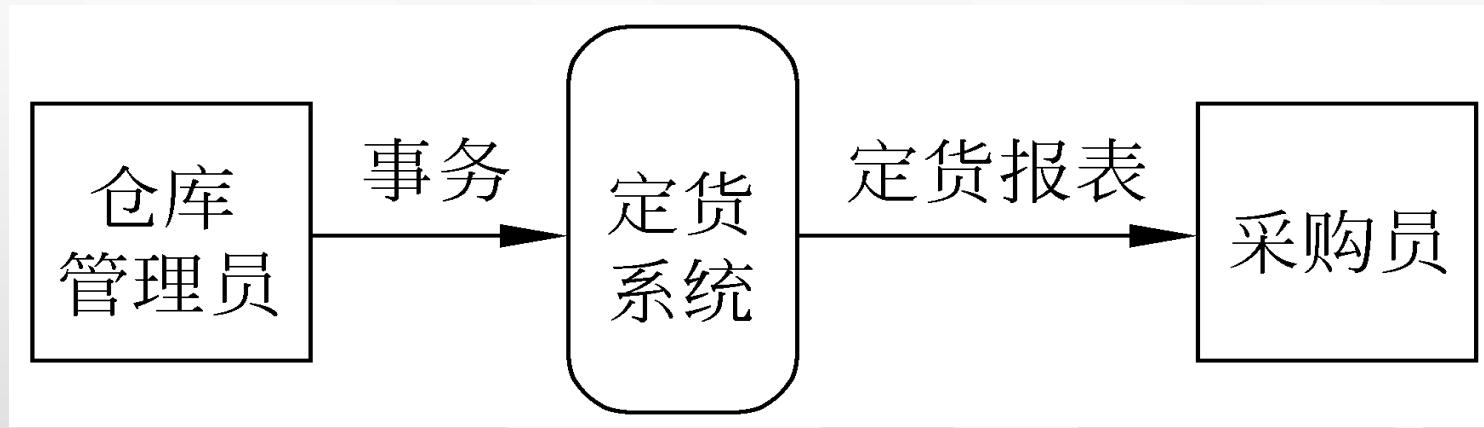


图2.5 定货系统的基本系统模型

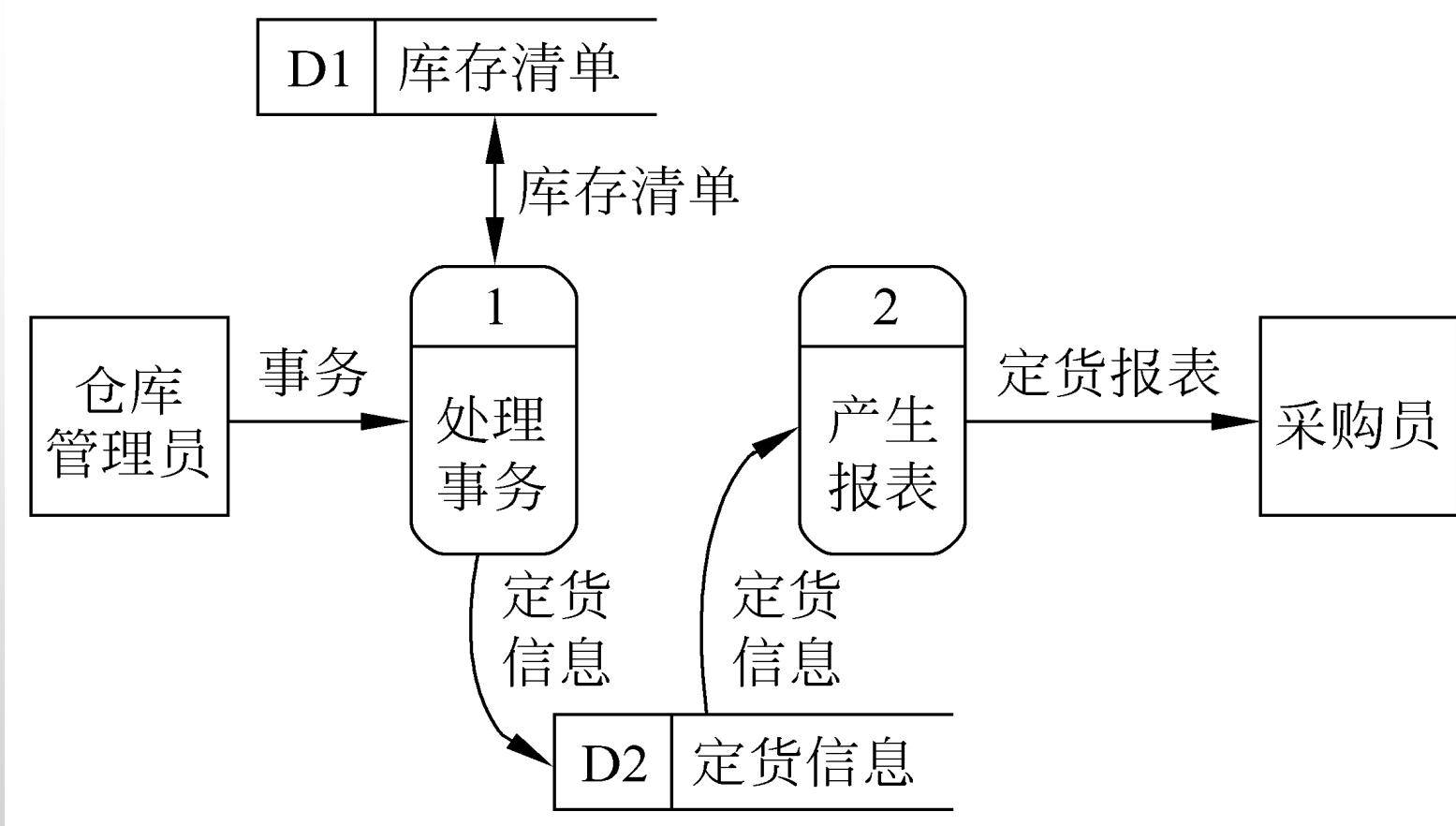


图2.6 定货系统的功能级数据流图

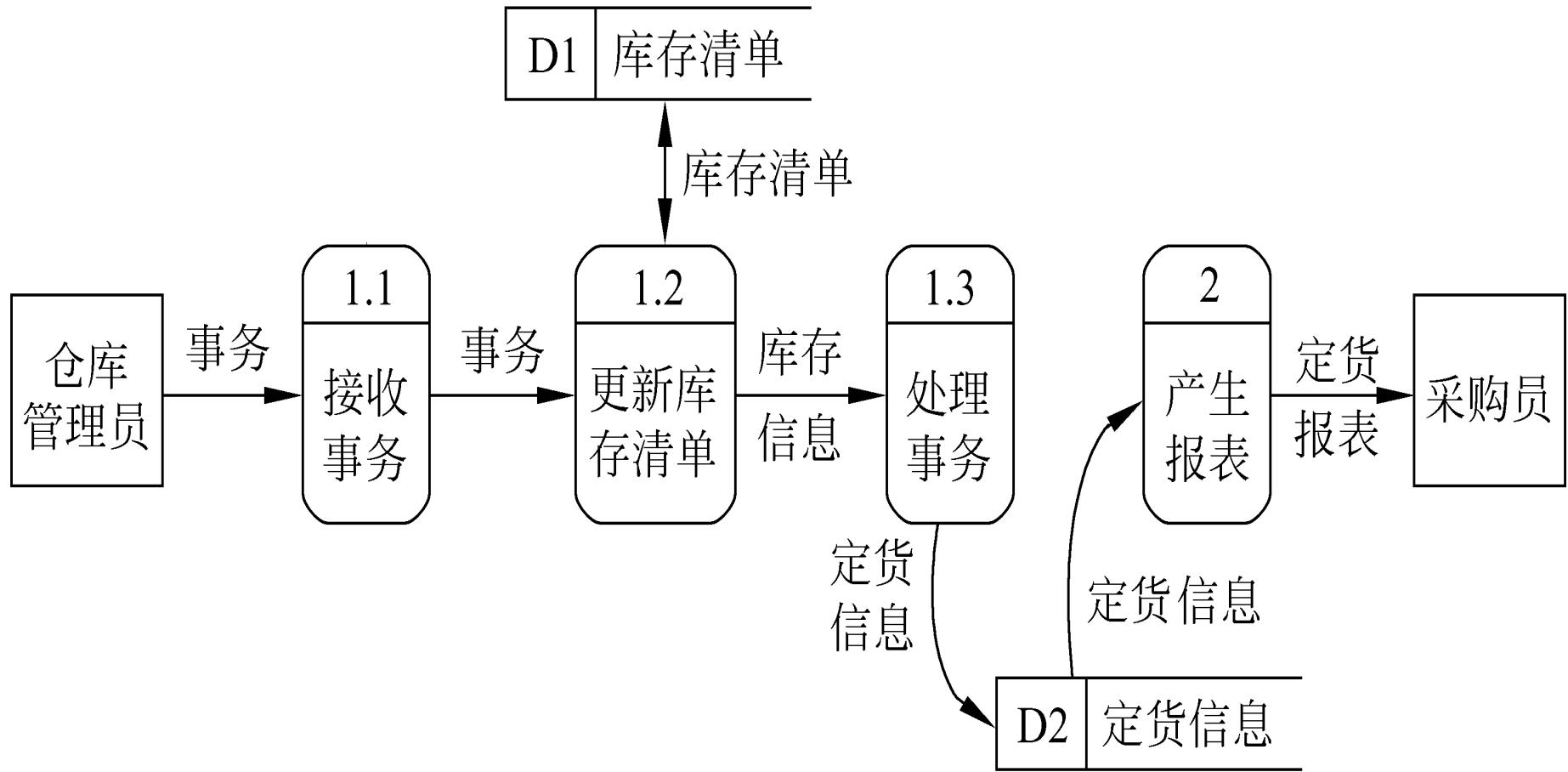


图2.7 把处理事务的功能进一步分解后的数据流图

2.4.3 命名

下面讲述在命名时应注意的问题：

1. 为数据流(或数据存储)命名

- (1) 有代表性
- (2) 有具体含义
- (3) 命名时遇到了困难，应该试试重新分解，看是否能克服这个困难。

2. 为处理命名

- (1) 通常先为数据流命名，然后再为与之相关联的处理命名。这样命名比较容易，而且体现了人类习惯的“由表及里”的思考过程。
- (2) 名字应该反映整个处理的功能，而不是它的一部分功能。
- (3) 名字最好由一个具体的及物动词加上一个具体的宾语组成。应该尽量避免使用“加工”、“处理”等空洞笼统的动词作名字。

- (4) 通常名字中仅包括一个动词，如果必须用两个动词才能描述整个处理的功能，则把这个处理再分解成两个处理可能更恰当些。
- (5) 如果在为某个处理命名时遇到困难，则很可能是发现了分解不当的迹象，应考虑重新分解。

3. 为数据源点/终点命名

数据源点/终点并不需要在开发目标系统的过
程中设计和实现，它并不属于数据流图的核心
内容，只不过是目标系统的外围环境部分(可
能是人员、计算机外部设备或传感器装置)。
通常，为数据源点/终点命名时采用它们在问
题域中习惯使用的名字(如“采购员”、“仓
库管理员”等)。

2.4.4 用途

1、画数据流图的基本目的是利用它作为**交
流**信息的工具。

数据流图应该分层，并且在把功能级数据流图细化后得到的处理超过9个时，应该采用画分图的办法，也就是把每个主要功能都细化为一张数据流分图，而原有的功能级数据流图用来描绘系统的整体逻辑概貌。

2、数据流图的另一个主要用途是作为分析和设计的工具。

当用数据流图辅助物理系统的设计时，以图中不同处理的定时要求为指南，能够在数据流图上画出许多组自动化边界，每组自动化边界可能意味着一个不同的物理系统，因此可以根据系统的逻辑模型考虑系统的物理实现。

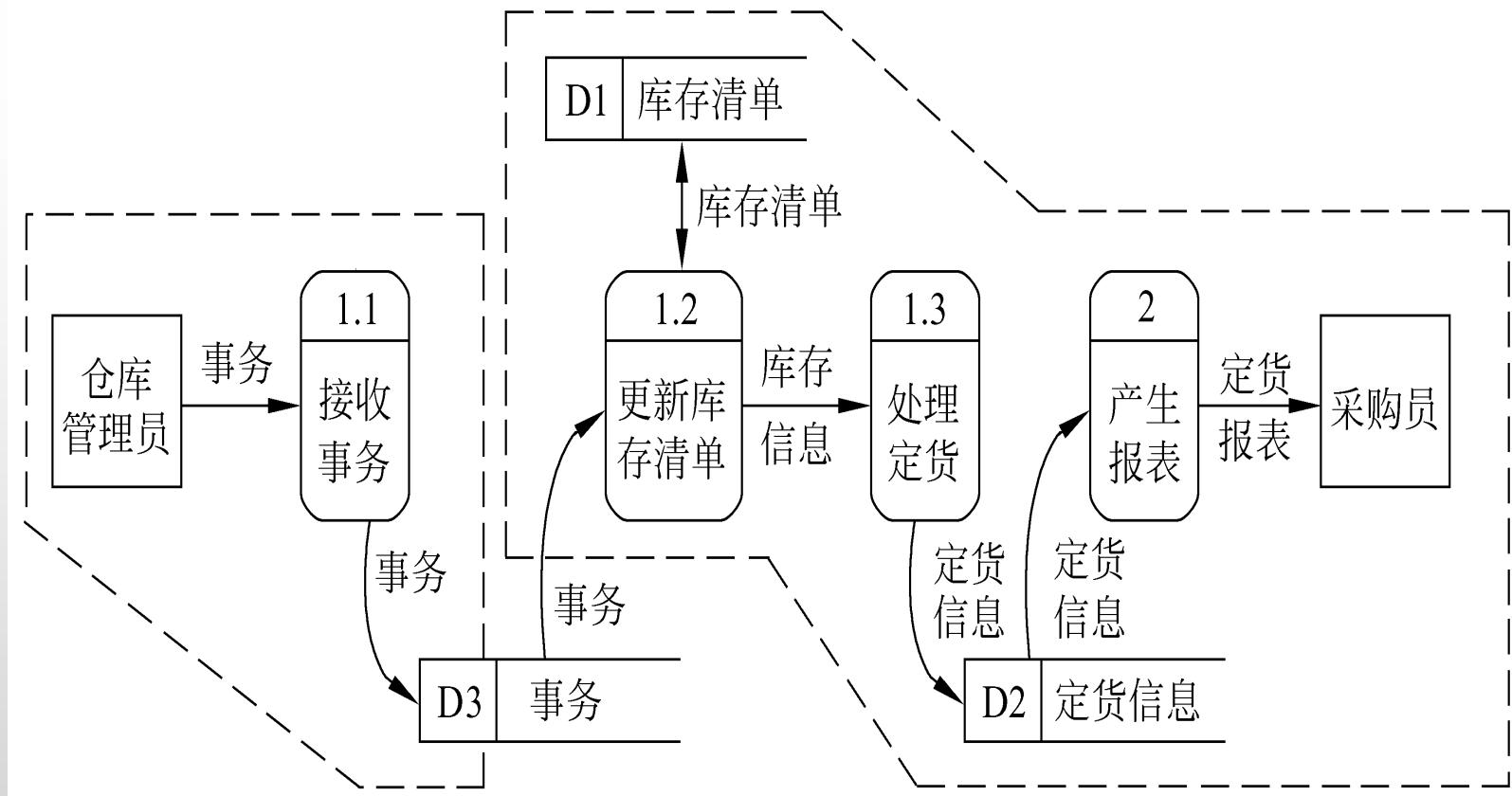


图2.8 这种划分自动化边界的方法暗示
以批量方式更新库存清单

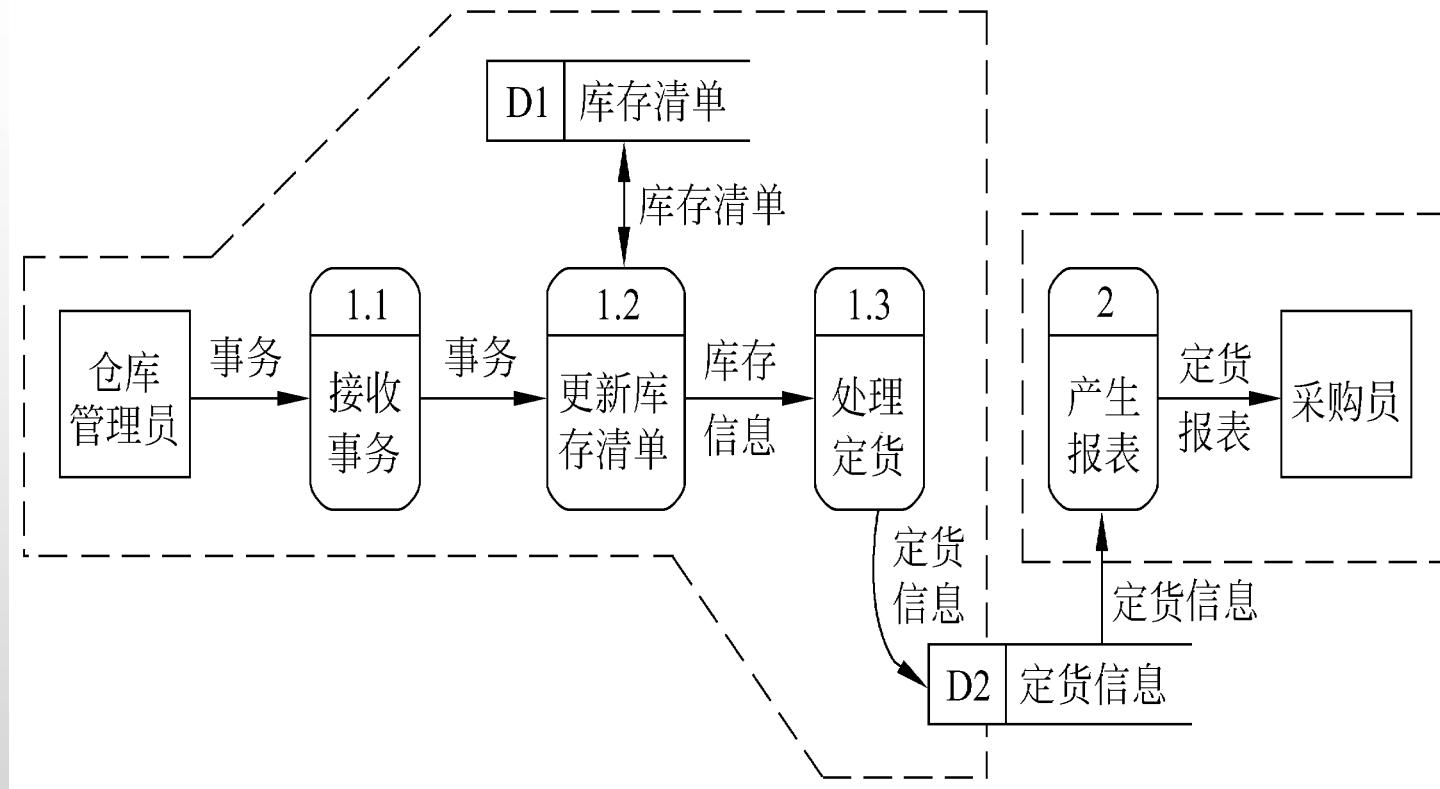


图2.9 另一种划分自动化边界的方法建议
以联机方式更新库存清单

3.4 实体-联系图

为了把用户的数据要求清楚、准确地描述出来，系统分析员通常建立一个概念性的数据模型（也称为信息模型）。概念性数据模型是一种面向问题的数据模型，是按照用户的观点对数据建立的模型。它描述了从用户角度看到的数据，与在软件系统中的实现方法无关。

- 数据模型中包含3种相互关联的信息：
- 数据对象、
 - 数据对象的属性
 - 数据对象彼此间相互连接的关系

3.4.1 数据对象

数据对象是对软件必须理解的复合信息的抽象。复合信息是指具有一系列不同性质或属性的事物，仅有单个值的事物(例如，宽度)不是数据对象。

数据对象，由一组属性来定义的实体。

数据对象彼此间是有关联的，例如，教师“教”课程，学生“学”课程，教或学的关系表示教师和课程或学生和课程之间的一种特定的连接。

数据对象只封装了数据而没有对施加于数据上的操作的引用，这是数据对象与面向对象范型(参见本书第9章)中的“类”或“对象”的显著区别。

3.4.2 属性

属性定义了数据对象的性质。必须把一个或多个属性定义为“标识符”，也就是说，当我们希望找到数据对象的一个实例时，用标识符属性作为“关键字”（通常简称为“键”）。

应该根据对所要解决的问题的理解，来确定特定数据对象的一组合适的属性。

3.4.3 联系

数据对象彼此之间相互连接的方式称为联系，也称为关系。联系可分为以下3种类型：

(1) 一对一联系(1 : 1)

例如，一个部门有一个经理，而每个经理只在一个部门任职，则部门与经理的联系是一对一的。

(2) 一对多联系(1 : N)

例如，某校教师与课程之间存在一对多的联系“教”，即每位教师可以教多门课程，但是每门课程只能由一位教师来教(见图3.2)。

(3) 多对多联系(M:N)

例如，一个学生可以学多门课程，而每门课程可以有多个学生来学。

联系也可能有属性。例如，学生“学”某门课程所取得的成绩，既不是学生的属性也不是课程的属性。由于“成绩”既依赖于某名特定的学生又依赖于某门特定的课程，所以它是学生与课程之间的联系“学”的属性(见图3.2)。

3.4.4 实体-联系图的符号

通常，使用实体-联系图(entity-relationship diagram)来建立数据模型。可以把实体-联系图简称为ER图，相应地可把用ER图描绘的数据模型称为ER模型。

ER图的符号：

实体

属性

关系

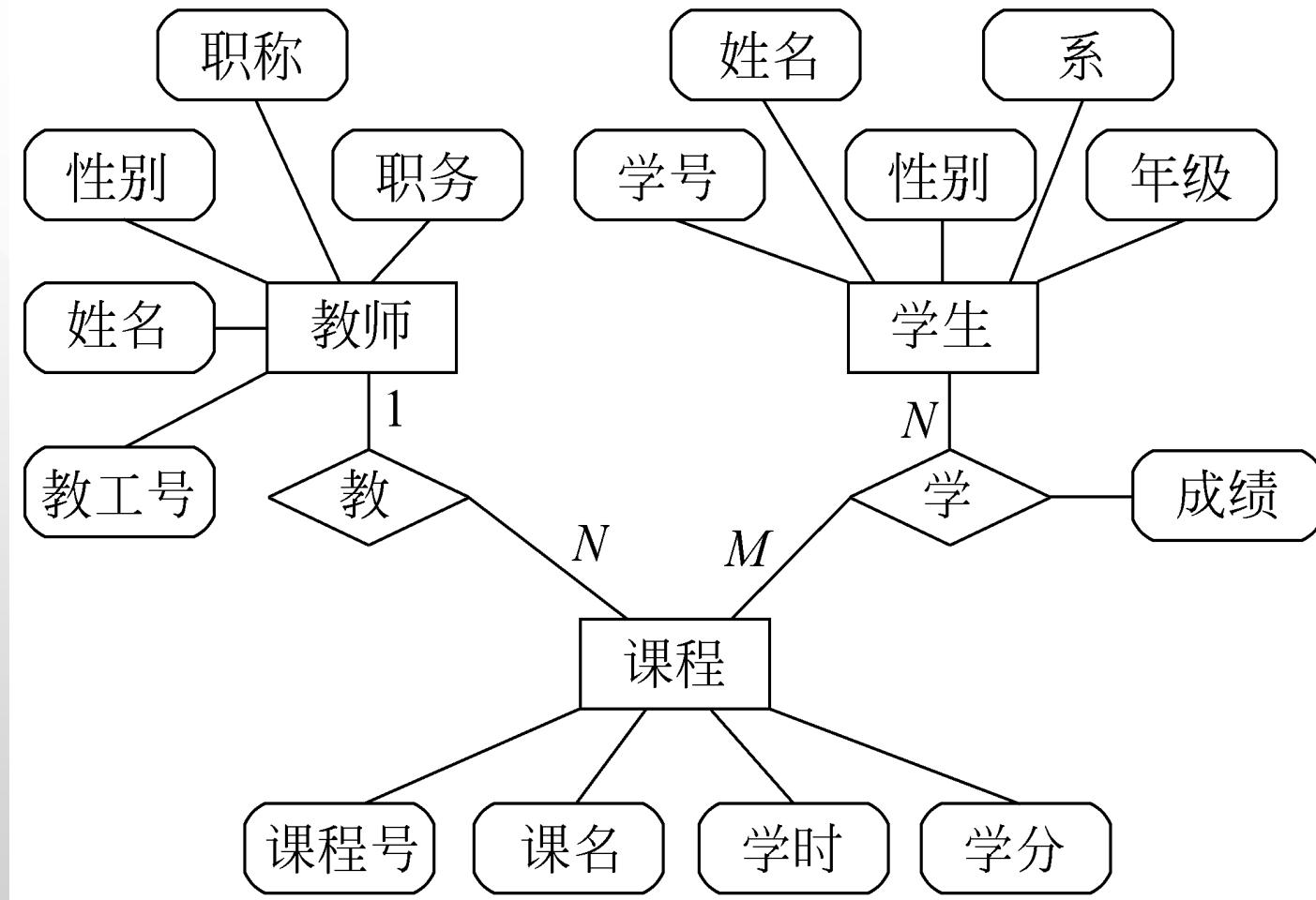


图3.2 某校教学管理ER图

人们通常就是用实体、联系和属性这3个概念来理解现实问题的，因此，ER模型比较接近人的习惯思维方式。此外，ER模型使用简单的图形符号表达系统分析员对问题域的理解，不熟悉计算机技术的用户也能理解它，因此，ER模型可以作为用户与分析员之间有效的交流工具。