

## 第3章 需求分析

**3.1 需求分析的任务**

**3.2 与用户沟通获取需求的方法**

**3.3 分析建模与规格说明**

**3.4 实体-联系图**

**3.5 数据规范化**

**3.6 状态转换图**

**3.7 其他图形工具**

**3.8 验证软件需求**

**3.9 小结**

**习题**

需求分析是软件定义时期的最后一个阶段，它的基本任务是准确地回答“系统必须做什么？”这个问题。

需求分析的任务还不是确定系统怎样完成它的工作，而仅仅是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。

在需求分析阶段结束之前，系统分析员应该写出软件需求规格说明书，以书面形式准确地描述软件需求。

在分析软件需求和书写软件需求规格说明书的过程中，分析员和用户都起着关键的、必不可少的作用。用户必须把他们对软件的需求尽量准确、具体地描述出来；分析员在需求分析开始时他们对用户的需求并不十分清楚，必须通过与用户沟通获取用户对软件的需求。

需求分析和规格说明是一项十分艰巨复杂的工作。用户与分析员之间需要沟通的内容非常多，在双方交流信息的过程中很容易出现误解或遗漏，也可能存在二义性。因此，不仅在整個需求分析过程中应该采用行之有效的方法，必须严格审查验证需求分析的结果。

目前有许多需求分析的结构化分析方法，这些分析方法都遵守下述准则：

- (1) 必须理解并描述问题的信息域，根据这条准则应该建立数据模型。
- (2) 必须定义软件应完成的功能，这条准则要求建立功能模型。
- (3) 必须描述作为外部事件结果的软件行为，这条准则要求建立行为模型。
- (4) 必须对描述信息、功能和行为的模型进行分解，用层次的方式展示细节。

## 3.1 需求分析的任务

### 3.1.1 确定对系统的综合要求

#### 1. 功能需求

这方面的需求指定系统必须提供的服务。通过需求分析应该划分出系统必须完成的所有功能。

#### 2. 性能需求

性能需求指定系统必须满足的定时约束或容量约束，通常包括速度(响应时间)、信息量速率、主存容量、磁盘容量、安全性等方面的需求。

### 3. 可靠性和可用性需求

可靠性需求定量地指定系统的可靠性。

可用性与可靠性密切相关，它量化了用户可以使用系统的程度。

### 4. 出错处理需求

这类需求说明系统对环境错误应该怎样响应。“出错处理”指的是当应用系统发现它自己犯下一个错误时所采取的行动。但是，应该有选择地提出这类出错处理需求，对应用系统本身错误的检测应该仅限于系统的关键部分，而且应该尽可能少。



## 5. 接口需求

接口需求描述应用系统与它的环境通信的格式。常见的接口需求有：用户接口需求；硬件接口需求；软件接口需求；通信接口需求。

## 6. 约束

设计约束或实现约束描述在设计或实现应用系统时应遵守的限制条件。常见的约束有：精度；工具和语言约束；设计约束；应该使用的标准；应该使用的硬件平台。

## 7. 逆向需求

## 8. 将来可能提出的要求

应该明确地列出那些虽然不属于当前系统开发范畴，但是将来很可能会提出来的要求。这样做的目的是，在设计过程中对系统将来可能的扩充和修改预做准备，以便一旦确实需要时能比较容易地进行这种扩充和修改。

### 3.1.2 分析系统的数据要求

任何一个软件系统本质上都是信息处理系统，系统必须处理的信息和系统应该产生的信息在很大程度上决定了系统的面貌。分析系统的数据要求通常采用建立数据模型的方法。

为了提高可理解性，常常利用图形工具辅助描绘数据结构。常用的图形工具有层次方框图和Warnier图。

### 3.1.3 导出系统的逻辑模型

综合上述两项分析的结果可以导出系统的详细的逻辑模型，通常用数据流图、实体-联系图、状态转换图、数据字典和主要的处理算法描述这个逻辑模型。

### 3.1.4 修正系统开发计划

根据在分析过程中获得的对系统的更深入更具体的了解，可以比较准确地估计系统的成本和进度，修正以前制定的开发计划。

## 3.2 与用户沟通获取需求的方法

### 3.2.1 访谈

访谈是最早开始使用的获取用户需求的技术，也是迄今为止仍然广泛使用的需求分析技术。

访谈有两种基本形式，分别是正式的和非正式的访谈。正式访谈时，系统分析员将提出一些事先准备好的具体问题。在非正式访谈中，分析员将提出一些用户可以自由回答的开放性问题，以鼓励被访问人员说出自己的想法。

当需要调查大量人员的意见时，向被调查人分发调查表是一个十分有效的做法。经过仔细考虑写出的书面回答可能比被访者对问题的口头回答更准确。分析员仔细阅读收回的调查表，然后再有针对性地访问一些用户，以便向他们询问在分析调查表时发现的新问题。

在访问用户的过程中使用情景分析技术往往非常有效。所谓情景分析就是对用户将来使用目标系统解决某个具体问题的方法和结果进行分析。



### 3.2.2 面向数据流自顶向下求精

结构化分析方法就是面向数据流自顶向下逐步求精进行需求分析的方法。通过可行性研究已经得出了目标系统的高层数据流图，需求分析的目标之一就是把数据流和数据存储定义到元素级。为了达到这个目标，通常从数据流图的输出端着手分析，这是因为系统的基本功能是产生这些输出，输出数据决定了系统必须具有的最基本的组成元素。

沿数据流图从输出端往输入端回溯，能够确定每个数据元素的来源，与此同时也就初步定义了有关的算法。

通常把分析过程中得到的有关数据元素的信息记录在数据字典中，把对算法的简明描述记录在IPO图中。通过分析而补充的数据流、数据存储和处理，应该添加到数据流图的适当位置上。

从输入端开始，分析员借助数据流图、数据字典和IPO图向用户解释输入数据是怎样一步一步地转变成输出数据的。这些解释集中反映了通过前面的分析工作分析员所获得的对目标系统的认识。

用户应该注意倾听分析员的报告，并及时纠正和补充分析员的认识。复查过程验证了已知的元素，补充了未知的元素，填补了文档中的空白。

随着分析过程的进展，经过问题和解答的反复循环，分析员越来越深入具体地定义了目标系统，最终得到对系统数据和功能要求的满意了解。图3.1粗略地概括了上述分析过程。

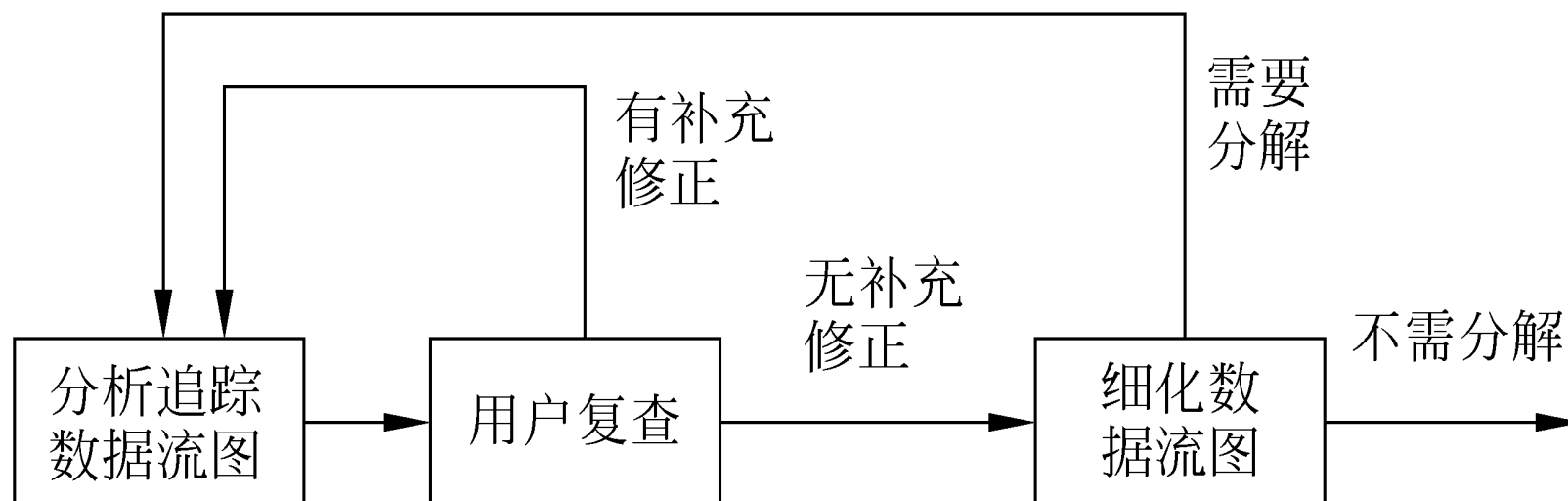


图3.1 面向数据流自顶向下求精过程

## 3.3 分析建模与规格说明

### 3.3.1 分析建模

所谓模型，就是为了理解事物而对事物做出的一种抽象，是对事物的一种无歧义的书面描述。通常，模型由一组图形符号和组织这些符号的规则组成。

结构化分析实质上是一种创建模型的活动。为了开发出复杂的软件系统，系统分析员应该从不同角度抽象出目标系统的特性，使用精确的表示方法构造系统的模型，验证模型是否满足用户对目标系统的需求，并在设计过程中逐渐把和实现有关的细节加进模型中，直至最终用程序实现模型。

根据结构化分析准则，需求分析过程应该建立3种模型，它们分别是数据模型、功能模型和行为模型。

### 3.3.2 软件需求规格说明

需求分析阶段还应该写出软件需求规格说明书，它是需求分析得出的最主要的文档。

通常用自然语言完整、准确、具体地描述系统的数据要求、功能需求、性能需求、可靠性和可用性要求、出错处理需求、接口需求、约束以及将来可能提出的要求。



## 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，它反映了用户的现实环境，而且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、数据对象的属性及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。所谓复合信息是指具有一系列不同性质或属性的事物。数据对象可以是外部实体、事物、行为、事件、角色、单位、地点或结构等。总之，可以由一组属性来定义的实体都可以被认为是数据对象。

### 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”（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)
- (2) 一对多联系(1 : N)
- (3) 多对多联系(M : N)

### 3.4.4 实体-联系图的符号

使用实体-联系图(entity-relationship diagram)来建立数据模型。用ER图描绘的数据模型称为ER模型。ER图中包含了实体(即数据对象)、关系和属性等3种基本成分，通常用矩形框代表实体，用连接相关实体的菱形框表示关系，用椭圆形或圆角矩形表示实体(或关系)的属性，并用直线把实体(或关系)与其属性连接起来。例如，图3.2是某学校教学管理的ER图。

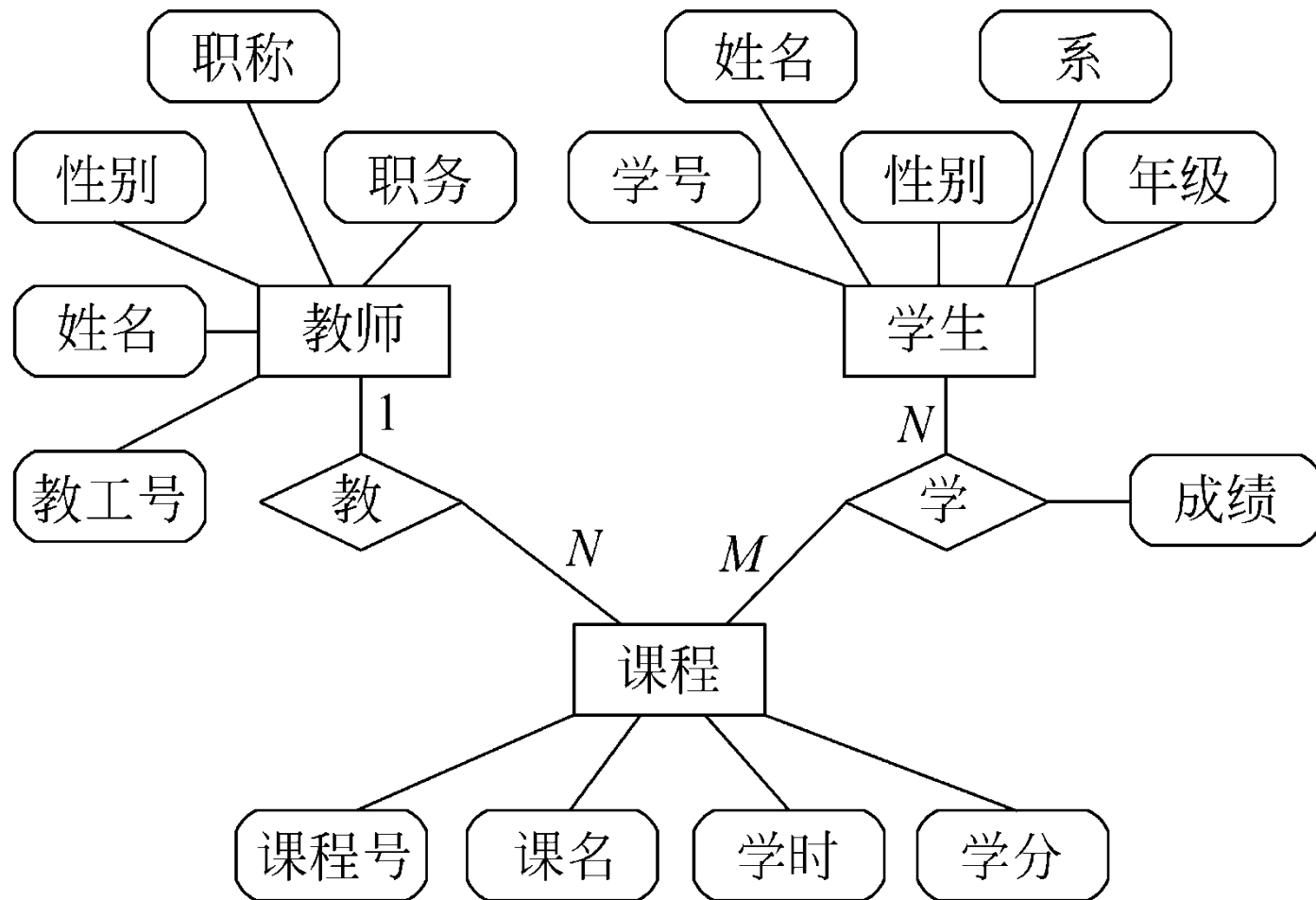


图3.2 某校教学管理ER图

## 3.5 数据规范化

软件系统经常使用各种长期保存的信息，这些信息通常以一定方式组织并存储在数据库或文件中，为减少数据冗余，避免出现插入异常或删除异常，简化修改数据的过程，通常需要把数据结构规范化。

通常用“范式(normal forms)”定义消除数据冗余的程度。第一范式(1 NF)数据冗余程度最大，第五范式(5 NF)数据冗余程度最小。从实用角度来看，在大多数场合选用第三范式都比较恰当。

- (1) 第一范式每个属性值都必须是原子值，即仅仅是一个简单值而不含内部结构。
- (2) 第二范式满足第一范式条件，而且每个非关键字属性都由整个关键字决定(而不是由关键字的一部分来决定)。
- (3) 第三范式符合第二范式条件，每个非关键字属性都仅由关键字决定，而且一个非关键字属性不能仅仅是对另一个非关键字属性的进一步描述(即一个非关键字属性值不依赖于另一个非关键字属性值)。



## 3.6 状态转换图

状态转换图(简称为状态图)通过描绘系统的状态及引起系统状态转换的事件，来表示系统的行为。

此外，状态图还指明了作为特定事件的结果系统将做哪些动作(例如，处理数据)。因此，状态图提供了行为建模机制，可以满足分析准则的要求。

### 3.6.1 状态

状态是任何可以被观察到的系统行为模式，一个状态代表系统的一种行为模式。状态规定了系统对事件的响应方式。系统对事件的响应，既可以是做一个(或一系列)动作，也可以是仅仅改变系统本身的状态，还可以是既改变状态又做动作。

在状态图中定义的状态主要有：初态(即初始状态)、终态(即最终状态)和中间状态。在一张状态图中只能有一个初态，而终态则可以有0至多个。

状态图既可以表示系统循环运行过程，也可以表示系统单程生命期。当描绘循环运行过程时，通常并不关心循环是怎样启动的。当描绘单程生命期时，需要标明初始状态(系统启动时进入初始状态)和最终状态(系统运行结束时到达最终状态)。

### 3.6.2 事件

事件是在某个特定时刻发生的事情，它是对引起系统做动作或(和)从一个状态转换到另一个状态的外界事件的抽象。例如，内部时钟表明某个规定的时间段已经过去，用户移动或点击鼠标等都是事件。简而言之，事件就是引起系统做动作或(和)转换状态的控制信息。

### 3.6.3 符号

在状态图中，初态用实心圆表示，终态用一对同心圆(内圆为实心圆)表示。

中间状态用圆角矩形表示，可以用两条水平横线把它分成上、中、下3个部分。上面部分为状态的名称，这部分是必须有的；中间部分为状态变量的名字和值，这部分是可选的；下面部分是活动表，这部分也是可选的。

活动表的语法格式如下：

事件名(参数表)/动作表达式

其中，“事件名”可以是任何事件的名称。在活动表中经常使用下述3种标准事件：**entry**，**exit**和**do**。**entry**事件指定进入该状态的动作，**exit**事件指定退出该状态的动作，而**do**事件则指定在该状态下的动作。需要时可以为事件指定参数表。活动表中的动作表达式描述应做的具体动作。

状态图中两个状态之间带箭头的连线称为状态转换，箭头指明了转换方向。状态变迁通常是由事件触发的，在这种情况下应在表示状态转换的箭头线上标出触发转换的事件表达式；如果在箭头线上未标明事件，则表示在源状态的内部活动执行完之后自动触发转换。

事件表达式的语法如下：

事件说明 [ 守卫条件 ] / 动作表达式

其中，

事件说明的语法为：事件名(参数表)。

守卫条件是一个布尔表达式。如果同时使用事件说明和守卫条件，则当且仅当事件发生且布尔表达式为真时，状态转换才发生。如果只有守卫条件没有事件说明，则只要守卫条件为真状态转换就发生。

动作表达式是一个过程表达式，当状态转换开始时执行该表达式。

图3.3给出了状态图中使用的主要符号。

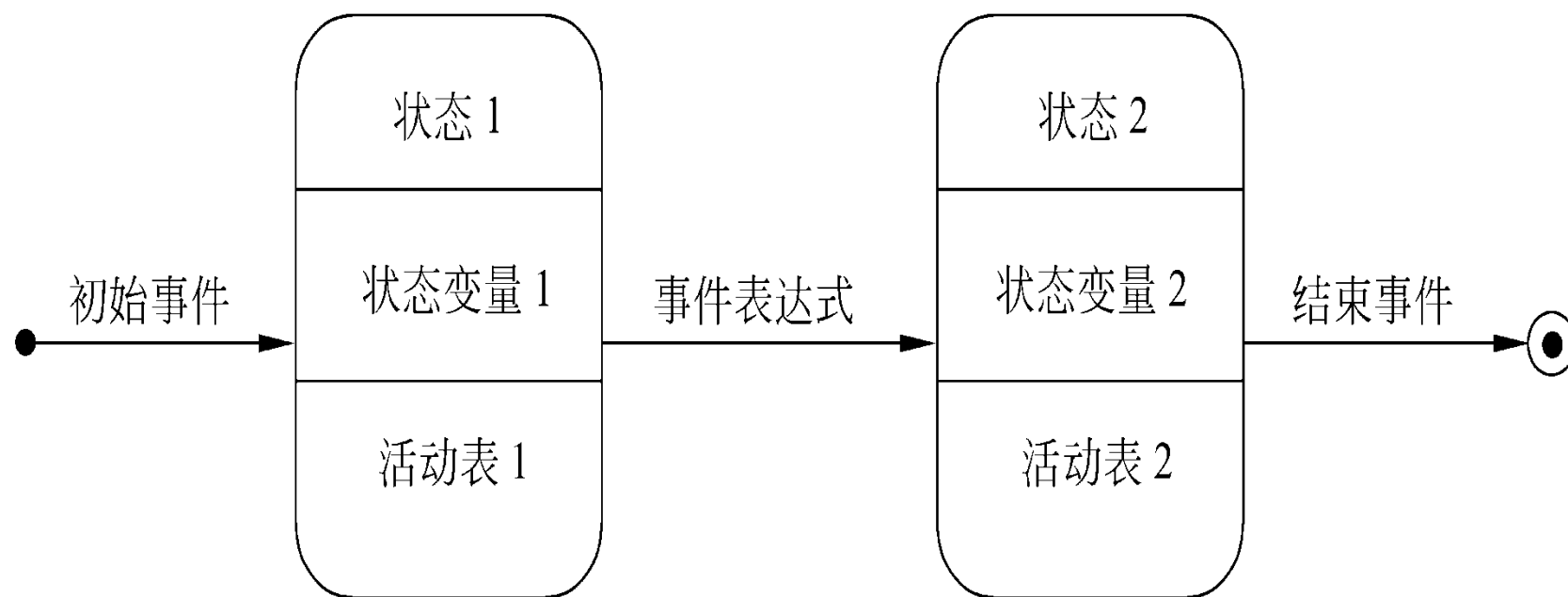


图3.3 状态图中使用的主要符号

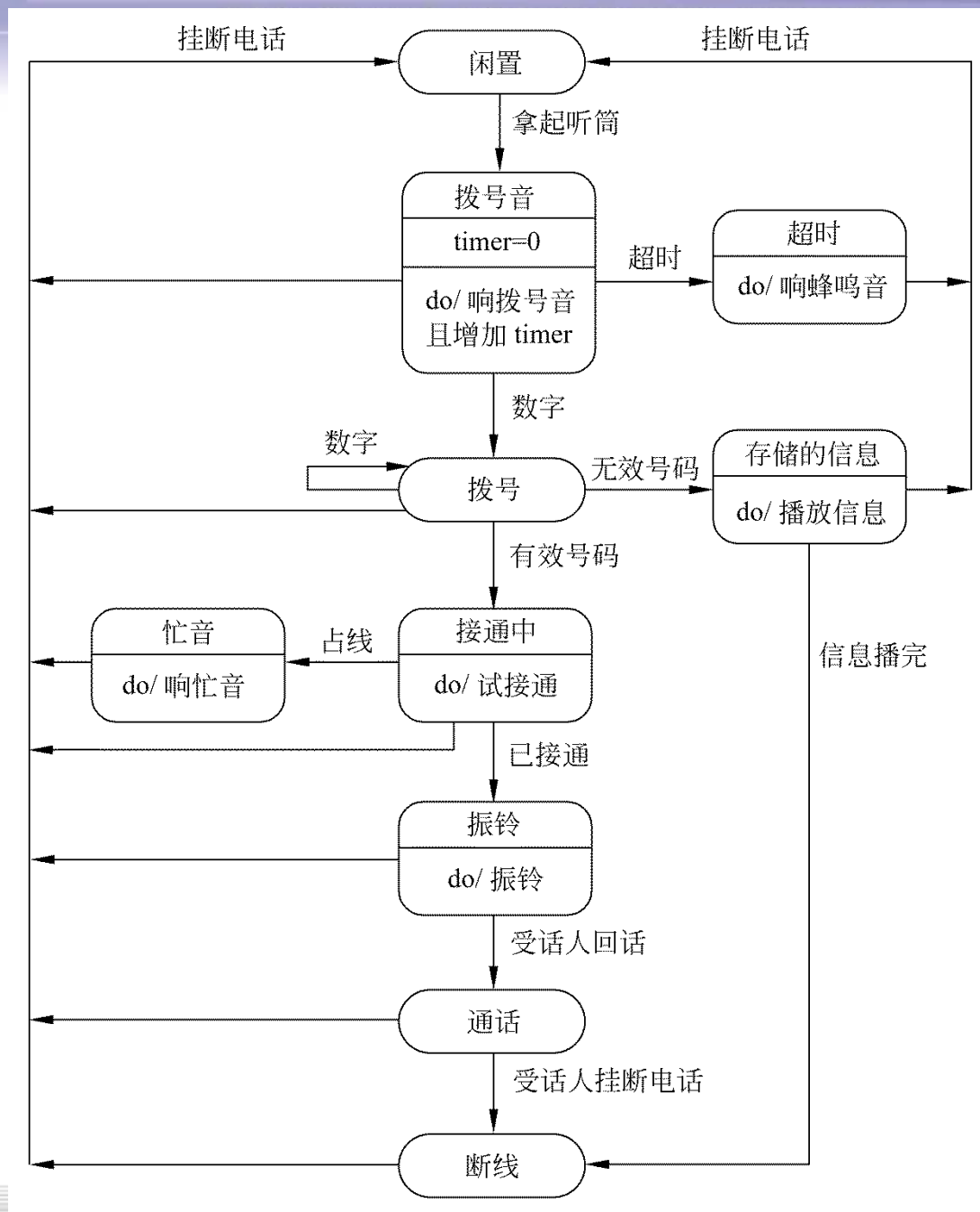


### 3.6.4 例子

图3.4是电话系统的状态图。

图中表明，没有人打电话时电话处于闲置状态；有人拿起听筒则进入拨号音状态，到达这个状态后，电话的行为是响起拨号音并计时；这时如果拿起听筒的人改变主意不想打了，他把听筒放下(挂断)，电话重又回到闲置状态；如果拿起听筒很长时间不拨号(超时)，则进入超时状态；.....。

### 3.6.4 例子



## 3.7 其他图形工具

### 3.7.1 层次方框图

层次方框图用树形结构的一系列多层次的矩形框描绘数据的层次结构。

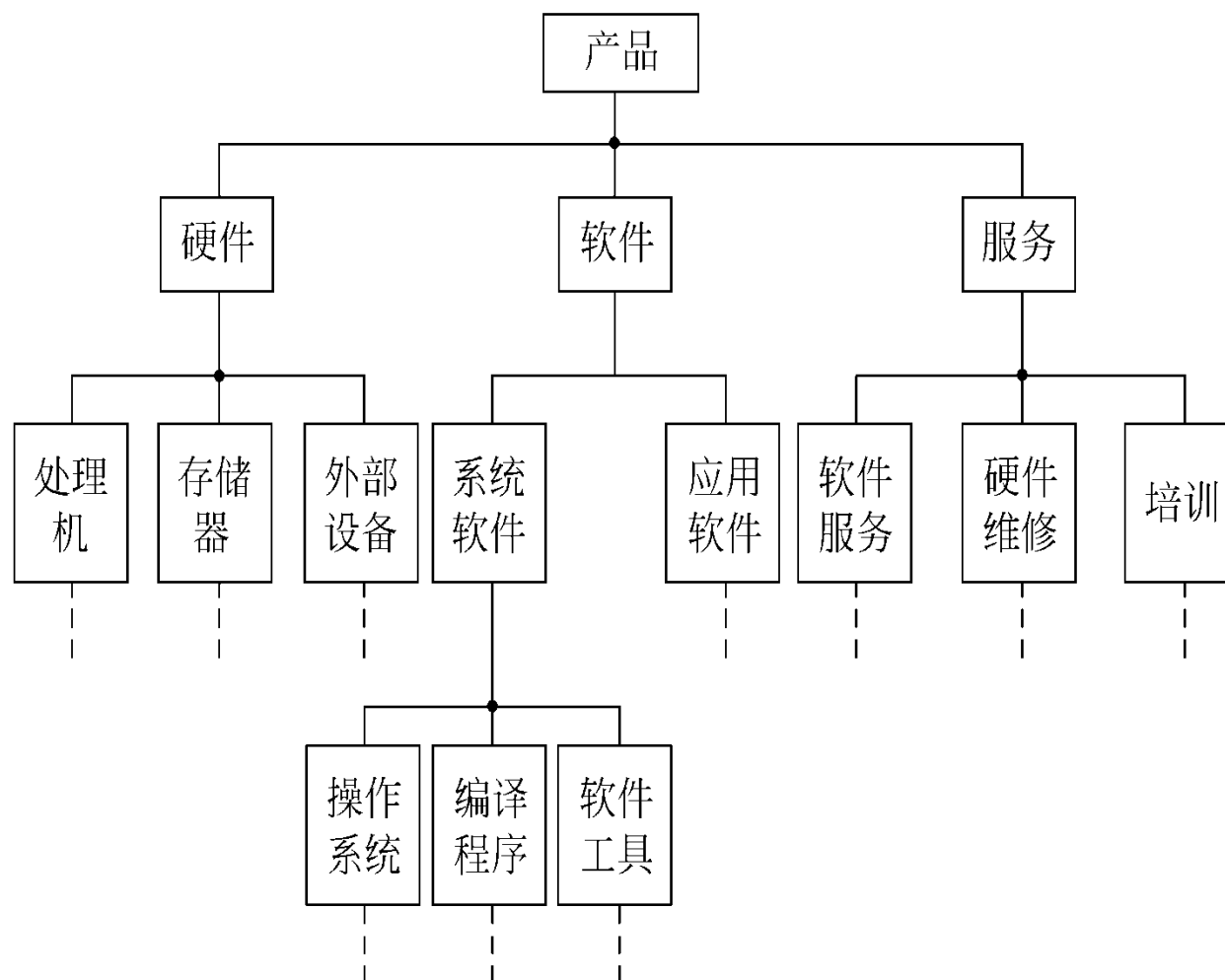


图3.5 层次方框图的一个例子

### 3.7.2 Warnier图

法国计算机科学家Warnier提出了表示信息层次结构的另外一种图形工具。Warnier图也用树形结构描绘信息。

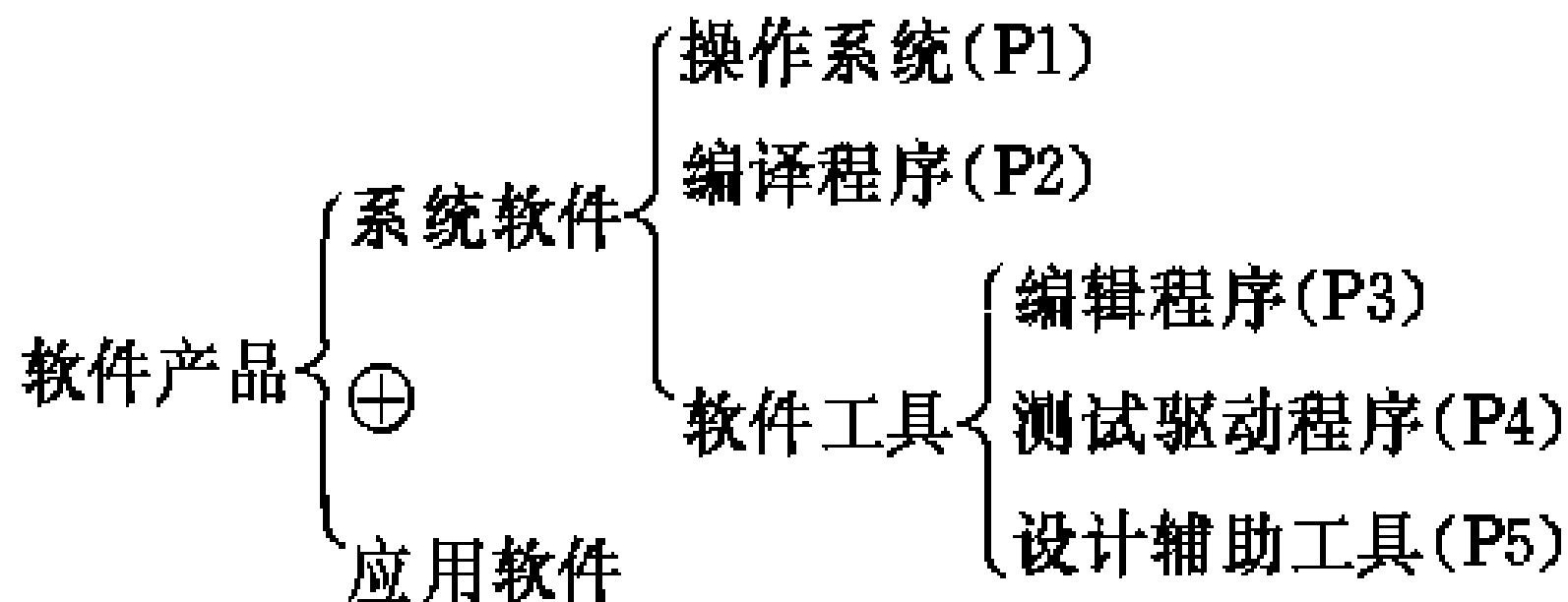


图3.6 Warnier图的一个例子

### 3.7.3 IPO图

**IPO图**是输入、处理、输出图的简称，它是**IBM**公司发展完善起来的一种图形工具，能够方便地描绘输入数据、对数据的处理和输出数据之间的关系。

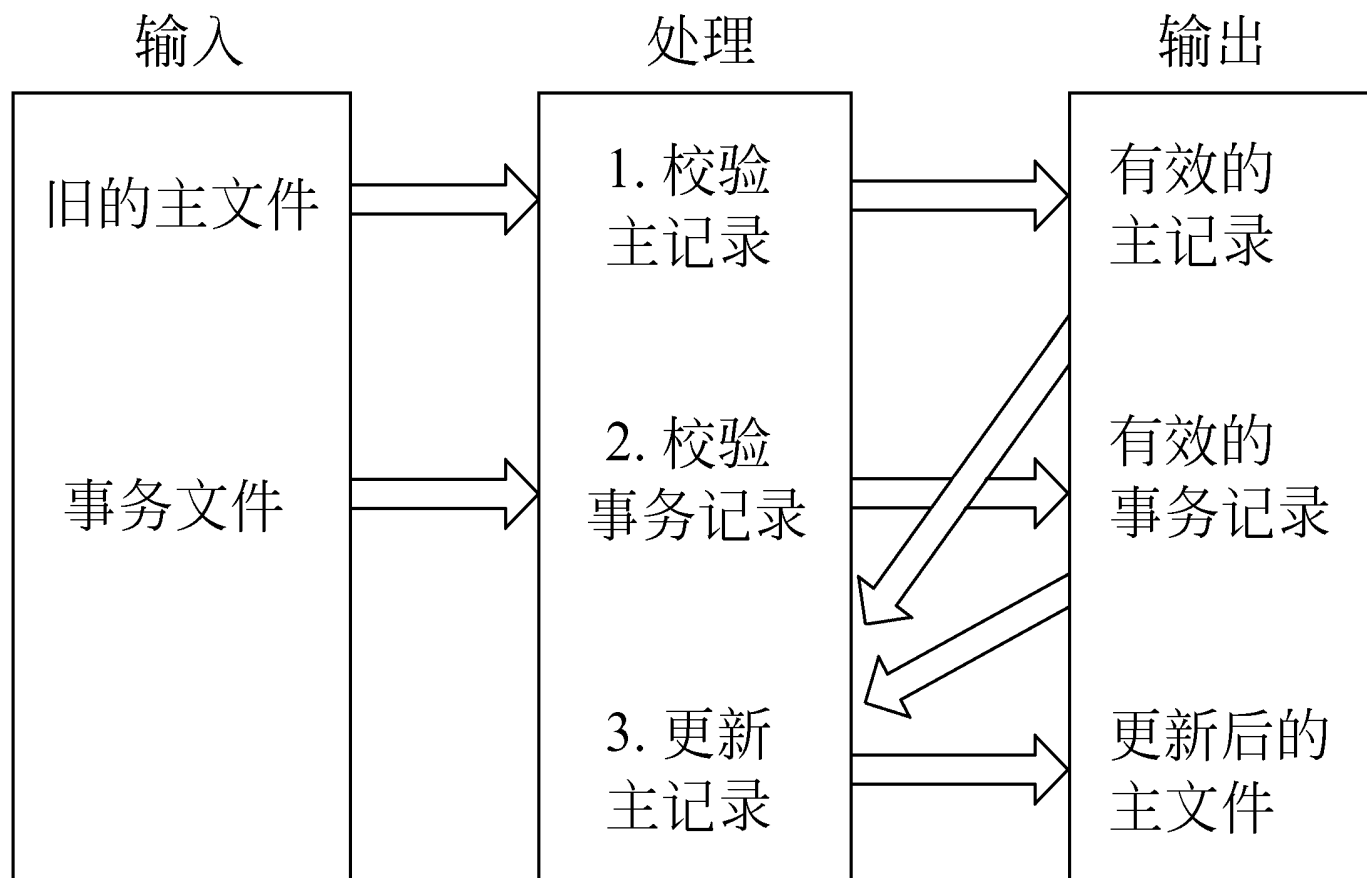


图3.7 IPO图的一个例子图



**IPO图的基本形式是在左边的框中列出有关的输入数据，在中间的框内列出主要的处理，在右边的框内列出产生的输出数据。处理框中列出处理的次序暗示了执行的顺序，在IPO图中用箭头清楚地指出数据通信的情况。图3.7是一个主文件更新的例子。**

一种改进的**IPO**图(也称为**IPO**表), 这种图中包含某些附加的信息, 在软件设计过程中将比原始的**IPO**图更有用。在需求分析阶段可以使用**IPO**图简略地描述系统的主要算法(即数据流图中各个处理的基本算法)。

IPO 表	
系统：_____	作者：_____
模块：_____	日期：_____
编号：_____	
被调用：	调用：
输入：	输出：
处理：	
局部数据元素：	注释：

图3.8 改进的IPO图的形式

## 3.8 验证软件需求

### 3.8.1 从哪些方面验证软件需求的正确性

需求分析阶段的工作结果是开发软件系统的重要基础。为了提高软件质量，确保软件开发成功，降低软件开发成本，必须严格验证需求的正确性。一般说来，应该从下述4个方面进行验证：

- (1) 一致性：所有需求必须是一致的，任何一条需求不能和其他需求互相矛盾。
- (2) 完整性：需求必须是完整的，规格说明书应该包括用户需要的每一个功能或性能。
- (3) 现实性：指定的需求应该用现有的硬件技术和软件技术基本上可以实现的。
- (4) 有效性：必须证明需求是正确有效的，确实能解决用户面对的问题。

## 3.8.2 验证软件需求的方法

### 1. 验证需求的一致性

当需求分析的结果是用自然语言书写的时候，除了靠人工技术审查验证软件系统规格说明书的正确性之外，目前还没有其他更好的“测试”方法。但是，这种非形式化的规格说明书是难于验证的，人工审查的效果是没有保证的，冗余、遗漏和不一致等问题可能没被发现而继续保留下来，以致软件开发工作不能在正确的基础上顺利进行。

为了克服上述困难，人们提出了形式化的描述软件需求的方法。当软件需求规格说明书是用形式化的需求陈述语言书写的时候，可以用软件工具验证需求的一致性，从而能有效地保证软件需求的一致性。

## 2. 验证需求的现实性

为了验证需求的现实性，分析员应该参照以往开发类似系统的经验，分析用现有的软、硬件技术实现目标系统的可能性。必要的时候应该采用仿真或性能模拟技术，辅助分析软件需求规格说明书的现实性。

### 3. 验证需求的完整性和有效性

只有目标系统的用户才真正知道软件需求规格说明书是否完整、准确地描述了他们的需求。因此，检验需求的完整性，特别是证明系统确实满足用户的实际需要，只有在用户的密切合作下才能完成。



### 3.8.3 用于需求分析的软件工具

为了更有效地保证软件需求的正确性，特别是为了保证需求的一致性，需要有适当的软件工具支持需求分析工作。这类软件工具应该满足下列要求：

- (1) 必须有形式化的语法(或表)，因此可以用计算机自动处理使用这种语法说明的内容；
- (2) 使用这个软件工具能够导出详细的文档；
- (3) 必须提供分析(测试)规格说明书的不一致性和冗余性的手段，并且应该能够产生一组报告指明对完整性分析的结果；

(4) 使用这个软件工具之后，应该能够改进通信状况。

作为需求工程方法学的一部分，在1977年设计完成了RSL(需求陈述语言)。RSL中的语句是计算机可以处理的，处理以后把从这些语句中得到的信息集中存放在一个称为ASSM（抽象系统语义模型）的数据库中。有一组软件工具处理ASSM数据库中的信息以产生出用PASCAL语言书写的模拟程序，从而可以检验需求的一致性、完整性和现实性。

**1977年美国密执安大学开发了PSL/PSA(问题陈述语言/问题陈述分析程序)系统。这个系统是CADSAT（计算机辅助设计和规格说明分析工具）的一部分，它的基本结构类似于RSL。**

## 3.9 小结

传统软件工程方法学使用结构化分析技术，完成分析用户需求的工作。需求分析是发现、求精、建模、规格说明和复审的过程。需求分析的第一步是进一步了解用户当前所处的情况，发现用户所面临的问题和对目标系统的基本需求；接下来应该与用户深入交流，对用户的基本需求反复细化逐步求精，以得出对目标系统的完整、准确和具体的需求。具体地说，应该确定系统必须具有的功能、性能、可靠性和可用性，必须实现的出错处理需求、接口需求，必须满足的约束条件，并且预测系统的发展前景。

为了详细地了解并正确地理解用户的需求，必须使用适当方法与用户沟通。访谈是与用户通信的历史悠久的技术，至今仍被许多系统分析员采用。从可行性研究阶段得到的数据流图出发，在用户的协助下面向数据流自顶向下逐步求精，也是与用户沟通获取需求的一个有效的方法。为了促使用户与分析员齐心协力共同分析需求，人们研究出一种面向团队的需求收集法，称为简易的应用规格说明技术，现在这种技术已经成为信息系统领域使用的主流技术。实践表明，快速建立软件原型是最准确、最有效和最强大的需求分析技术。

为了更好地理解问题，人们常常采用建立模型的方法，结构化分析实质上就是一种建模活动，在需求分析阶段通常建立数据模型、功能模型和行为模型。除了创建分析模型之外，在需求分析阶段还应该写出软件需求规格说明书，经过严格评审并得到用户确认之后，作为这个阶段的最终成果。通常主要从一致性、完整性、现实性和有效性等4个方面复审软件需求规格说明书。

多数人习惯于使用实体-联系图建立数据模型，使用数据流图建立功能模型，使用状态图建立行为模型。读者应该掌握这些图形的基本符号，并能正确地使用这些符号建立软件系统的模型。

数据字典描述在数据模型、功能模型和行为模型中出现的数据对象及控制信息的特性，给出它们的准确定义。因此，数据字典成为把3种分析模型粘合在一起的“粘合剂”，是分析模型的“核心”。



为了提高可理解性，还可以用层次方框图或Warnier图等图形工具辅助描绘系统中的数据结构。为了减少冗余、简化修改步骤，往往需要规范数据的存储结构。

算法也是重要的，分析的基本目的是确定系统必须做什么。概括地说，任何一个计算机系统的基本功能都是把输入数据转变成输出信息，算法定义了转变的规则。因此，没有对算法的了解就不能确切知道系统的功能。**IPO**图是描述算法的有效工具。



## 习题

**3-1** 为什么要进行需求分析?通常对软件系统有哪些需求?

**3-2** 怎样与用户有效地沟通以获取用户的真实需求?

**3-3** 银行计算机储蓄系统的工作过程大致如下: 储户填写的存款单或取款单由业务员键入系统, 如果是存款则系统记录存款人姓名、住址(或电话号码)、身份证号码、存款类型、存款日期、到期日期、利率及密码(可选)等信息, 并印出存单给储户; 如果是取款而且存款时留有密码, 则系统首先核对储户密码, 若密码正确或存款时未留密码, 则系统计算利息并印出利息清单给储户。

请用数据流图描绘本系统的功能，并用实体-联系图描绘系统中的数据对象。

**3-4** 分析习题2第3题所述的机票预订系统。请用实体-联系图描绘本系统中的数据对象并用数据流图描绘本系统的功能。

**3-5** 分析习题2第4题所述的患者监护系统。请用实体-联系图描绘本系统中的数据对象并用数据流图描绘本系统的功能，画出本系统的顶层IPO图。

**3-6 复印机的工作过程大致如下：**未接到复印命令时处于闲置状态，一旦接到复印命令则进入复印状态，完成一个复印命令规定的工作后又回到闲置状态，等待下一个复印命令；如果执行复印命令时发现没纸，则进入缺纸状态，发出警告，等待装纸，装满纸后进入闲置状态，准备接收复印命令；如果复印时发生卡纸故障，则进入卡纸状态，发出警告等待维修人员来排除故障，故障排除后回到闲置状态。

请用状态转换图描绘复印机的行为。