

“十二五”普通高等教育本科国家级规划教材  
北京高等教育精品教材

21世纪软件工程专业规划教材

# 软件工程导论（第6版）



## 第3章 需求分析



# 第3章需求分析

为了开发出真正满足用户需求的软件产品，首先必须知道用户的需求。对软件需求的深入理解是软件开发工作获得成功的前提条件，不论人们把设计和编码工作做得如何出色，不能真正满足用户需求的程序只会令用户失望，给开发者带来烦恼。

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。



# 第3章需求分析

尽管目前有许多不同的用于需求分析的结构化分析方法，但是，所有这些分析方法都遵守下述准则。

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。

必须定义软件应完成的功能，这条准则要求建立功能模型。

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

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。



# 主要内容

- 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



# 主要内容



## 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



# 3.1 需求分析的任务

## 3.1.1 确定对系统的综合要求

虽然功能需求是对软件系统的一项基本需求，但却并不是唯一的需求。通常对软件系统有下述几方面的综合要求。

- 功能需求
- 性能需求
- 可靠性和可用性需求
- 出错处理需求
- 接口需求
- 约束
- 逆向需求
- 将来可能提出的要求



# 3.1 需求分析的任务

## 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

## 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。



# 3.1 需求分析的任务

## 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。

## 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。例如，如果它接收到从另一个系统发来的违反协议格式的消息，应该做什么？注意，上述这类错误并不是由该应用系统本身造成的。



# 3.1 需求分析的任务

## 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。

## 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。



# 3.1 需求分析的任务

## 逆向需求

逆向需求说明软件系统不应该做什么。理论上有限多个逆向需求，人们应该仅选取能澄清真实需求且可消除可能发生的误解的那些逆向需求。

## 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，但是据分析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 需求分析的任务

## 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，对软件设计有深远影响，因此，必须分析系统的数据要求，这是软件需求分析的一个重要任务。

复杂的数据由许多基本的数据元素组成，数据结构表示数据元素之间的逻辑关系。利用数据字典可以全面准确地定义数据，但是数据字典的缺点是不够形象直观。为了提高可理解性，常常利用图形工具辅助描绘数据结构。



# 3.1 需求分析的任务

## 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 需求分析的任务

## 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。



# 主要内容

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



## 3.2 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，例如，询问客户公司销售的商品种类、雇用的销售人员数目以及信息反馈时间应该多快等。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，例如，询问用户对目前正在使用的系统有哪些不满意的地方。



## 3.2 与用户沟通获取需求的方法

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，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

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 与用户沟通获取需求的方法

情景分析技术的用处主要体现在下述两个方面。

它能在某种程度上演示目标系统的行为，从而便于用户理解，而且还可能进一步揭示出一些分析员目前还不知道的需求。

由于情景分析较易为用户所理解，使用这种技术能保证用户在需求分析过程中始终扮演一个积极主动的角色。



## 3.2 与用户沟通获取需求的方法

### 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。

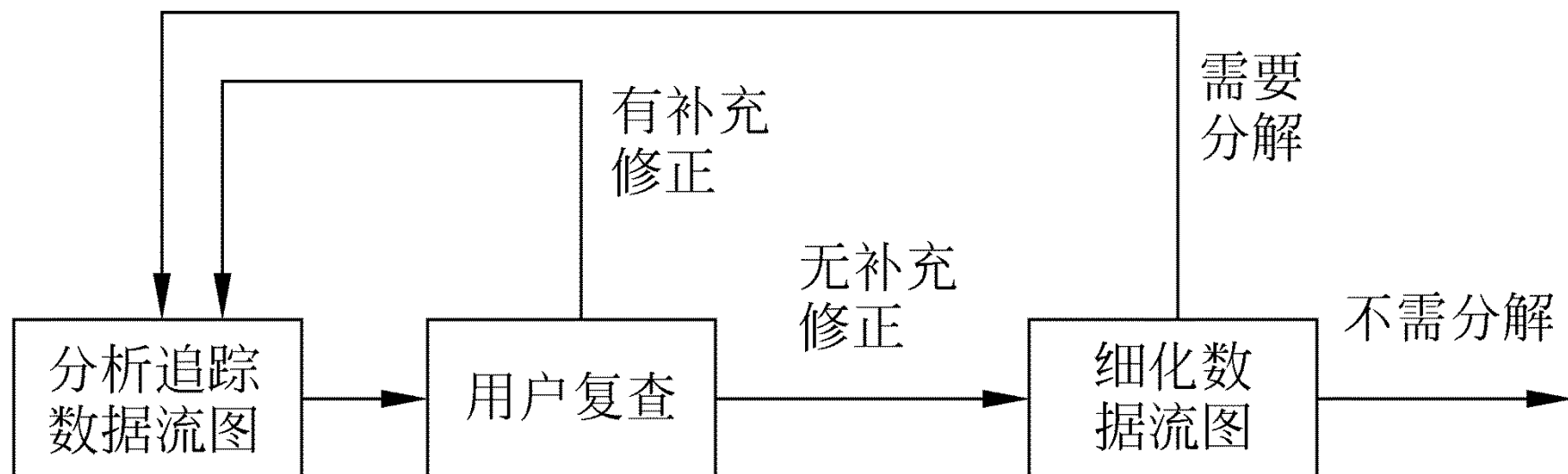
为了达到这个目标，通常从数据流图的输出端着手分析，这是因为系统的基本功能是产生这些输出，输出数据决定了系统必须具有的最基本的组成元素。

数据流图是帮助复查的极好工具，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。



## 3.2 与用户沟通获取需求的方法

随着分析过程的进展，经过提问和解答的反复循环，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。





## 3.2 与用户沟通获取需求的方法

### 3.2.3 简易的应用规格说明技术

简易的应用规格说明技术是为了解决使用传统的访谈或面向数据流自顶向下求精方法定义需求时，用户处于被动地位而且往往有意无意地与开发者区分“彼此”。由于不能像同一个团队的人那样齐心协力地识别和精化需求，这两种方法的效果有时并不理想的问题，提出的。



## 3.2 与用户沟通获取需求的方法

简易的应用规格说明技术分析需求的典型过程如下

- 1 • 进行初步的访谈
- 2 • 开发者和用户分别写出“产品需求”。
- 3 • 开会讨论，各自展示需求列表
- 4 • 得出了意见一致，为需求列表制定小型规格说明
- 5 • 根据会议结果，起草完整的软件需求规格说明



## 3.2 与用户沟通获取需求的方法

### 3.2.4 快速建立软件原型

为了快速地构建和修改原型，通常使用下述3种方法和工具。

- 第四代技术
- 可重用的软件构件
- 形式化规格说明和原型环境



## 3.2 与用户沟通获取需求的方法

快速原型就是快速建立起来的旨在演示目标系统主要功能的可运行的程序，快速原型应该具备的特性：

快速原型应该具备的第一个特性是“快速”。

快速原型应该具备的第二个特性是“容易修改”。



# 主要内容

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



## 3.3 分析建模与规格说明

### 3.3.1 分析建模

模型，就是为了理解事物而对事物作出的一种抽象，是对事物的一种无歧义的书面描述。

为了开发复杂的系统，应从不同角度（模型）抽象出目标系统的特性（数据模型、功能模型、行为模型）



## 3.3 分析建模与规格说明

- 实体联系图，描绘数据对象及数据对象之间的关系，是用于建立数据模型的图形。
- 数据流图是建立功能模型的基础。
- 状态转换图描绘了系统的各种行为模式和在不同状态间转换的方式。



## 3.3 分析建模与规格说明


### 3.3.2 软件需求规格说明

软件需求规格说明是需求分析阶段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、逆向需求以及将来可能提出的要求。



# 主要内容

- 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



## 3.4 实体联系图

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 实体联系图

### 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。

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 实体联系图

### 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，当人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”)。



## 3.4 实体联系图

### 3.4.3 联系

客观世界中的事物彼此间往往是有联系的。

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。

- 一对一联系 (1 : 1)
- 一对多联系 (1 : N)
- 多对多联系 (M : N)



## 3.4 实体联系图

### (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，图3.2表示学生与课程间的联系(“学”)是多对多的，即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。



## 3.4 实体联系图

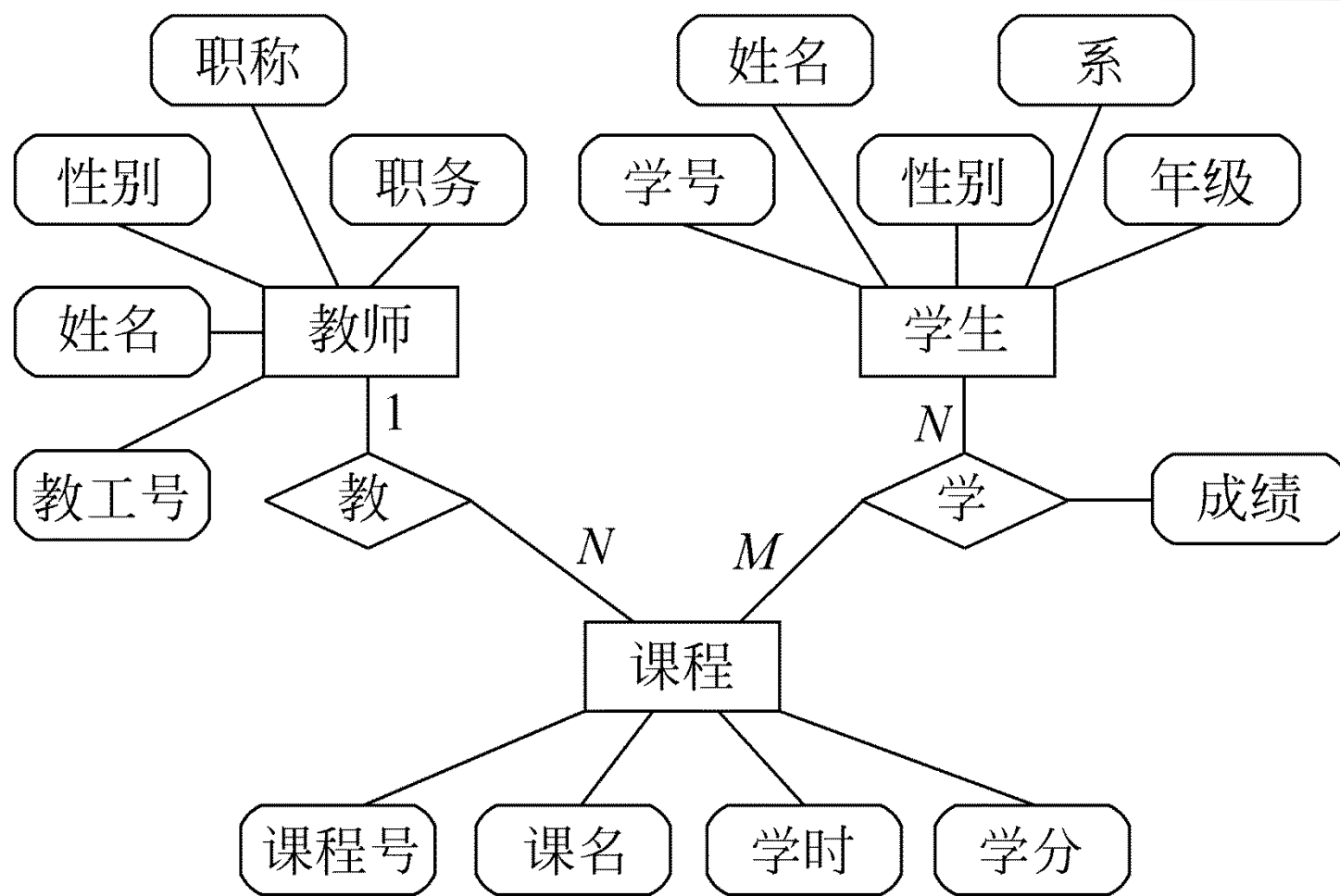


图3.2



## 3.4 实体联系图

### 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图中包含了实体(即数据对象)、关系和属性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。

ER模型可以作为用户与分析员之间有效的交流工具。



# 主要内容

- 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



## 3.5 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

第二  
范式

第三  
范式



# 主要内容

- 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



## 3.6 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。



## 3.6 状态转换图

### 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。

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。



## 3.6 状态转换图

### 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。

事件就是引起系统做动作或(和)转换状态的控制信息。



## 3.6 状态转换图

### 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，1这部分也是可选的。



## 3.6 状态转换图

### 3.6.3 符号

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。



## 3.6 状态转换图

### 3.6.3 符号

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)。



## 3.6 状态转换图

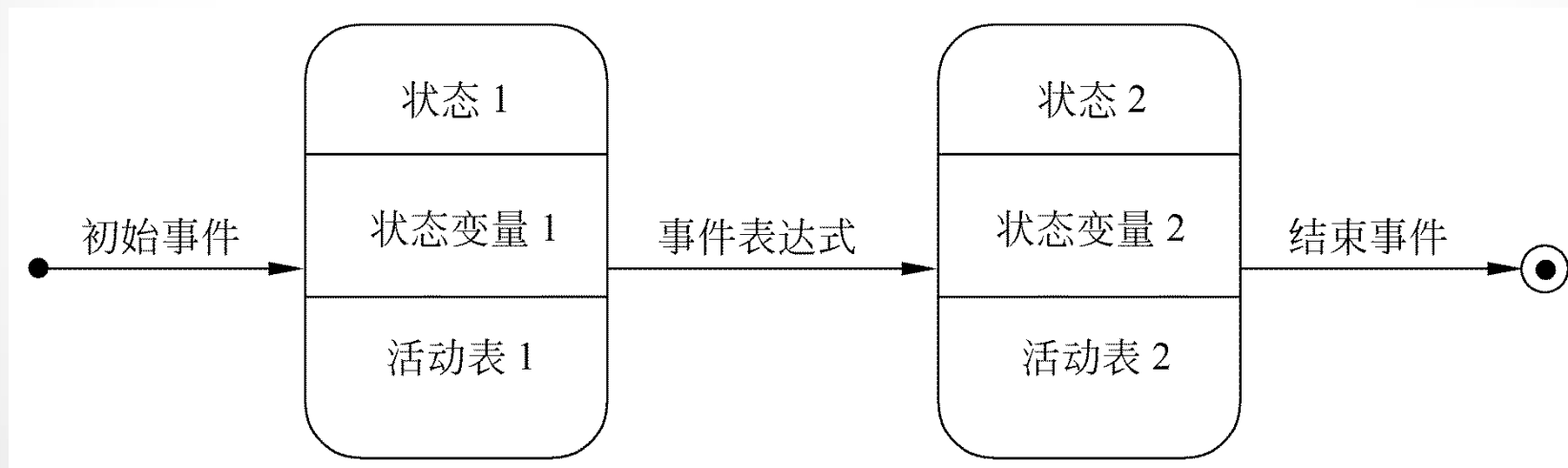
状态图中两个状态之间带箭头的连线称为状态转换，箭头指明了转换方向。状态变迁通常是由事件触发的，在这种情况下应在表示状态转换的箭头线上标出触发转换的事件表达式；如果在箭头线上未标明事件，则表示在源状态的内部活动执行完之后自动触发转换。

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 )。



## 3.6 状态转换图



状态图中使用的主要符号



## 3.6 状态转换图

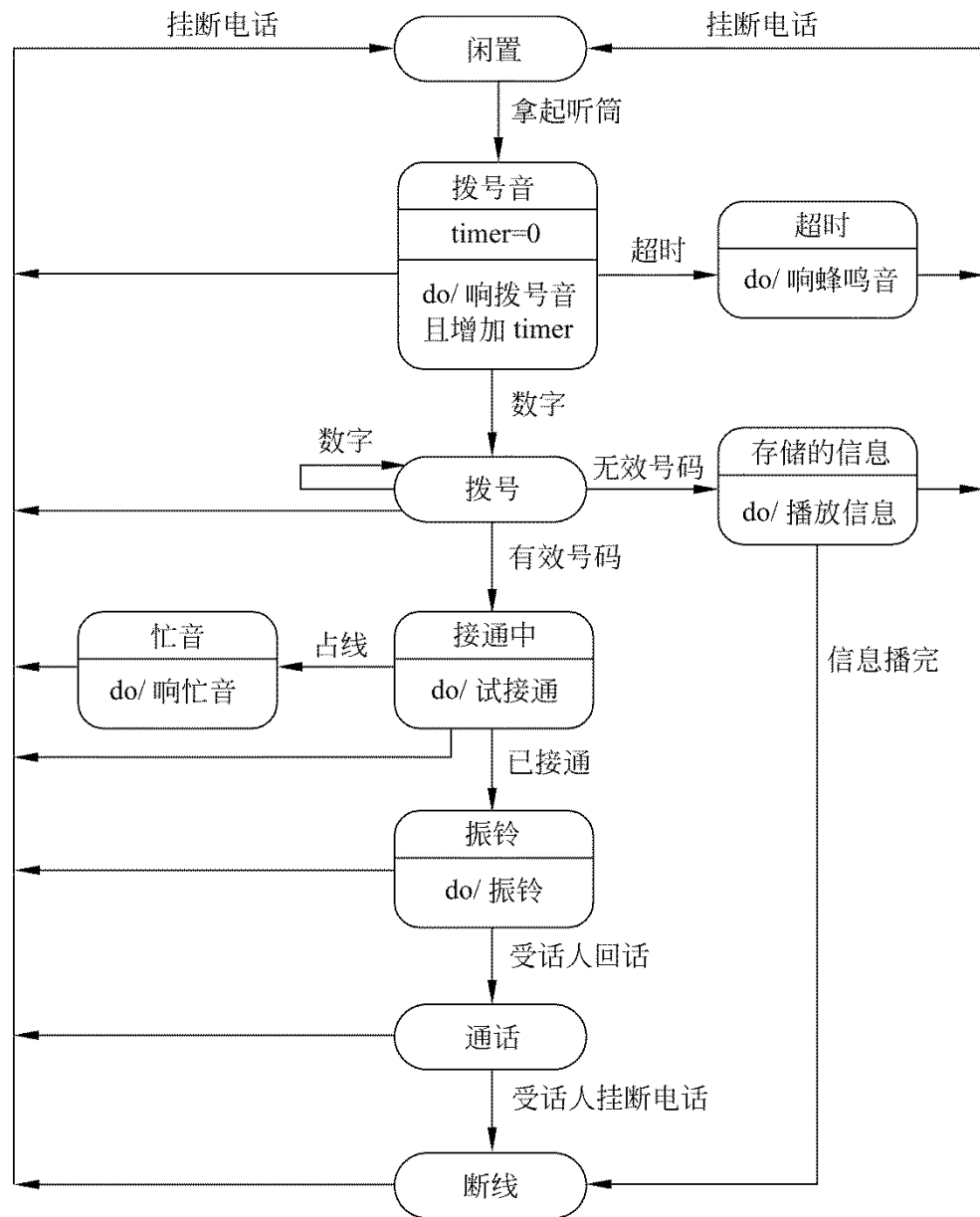
### 3.6.4 例子

为了具体说明怎样用状态图建立系统的行为模型，下面举一个例子

下图是人们非常熟悉的电话系统的状态图。



## 状态图中使用的主要符号





# 主要内容

- 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



## 3.7 其他图形工具

描述复杂的事物时，图形远比文字叙述优越得多，它形象直观容易理解。前面已经介绍了用于建立功能模型的数据流图、用于建立数据模型的实体-联系图和用于建立行为模型的状态图，本节再简要地介绍在需求分析阶段可能用到的另外3种图形工具。



## 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。

例如，描绘一家计算机公司全部产品的数据结构可以用下图3.5层次方框图表示。

这家公司的产品由硬件、软件和服务3类产品组成，软件产品又分为系统软件和应用软件，系统软件又进一步分为操作系统、编译程序和软件工具等。



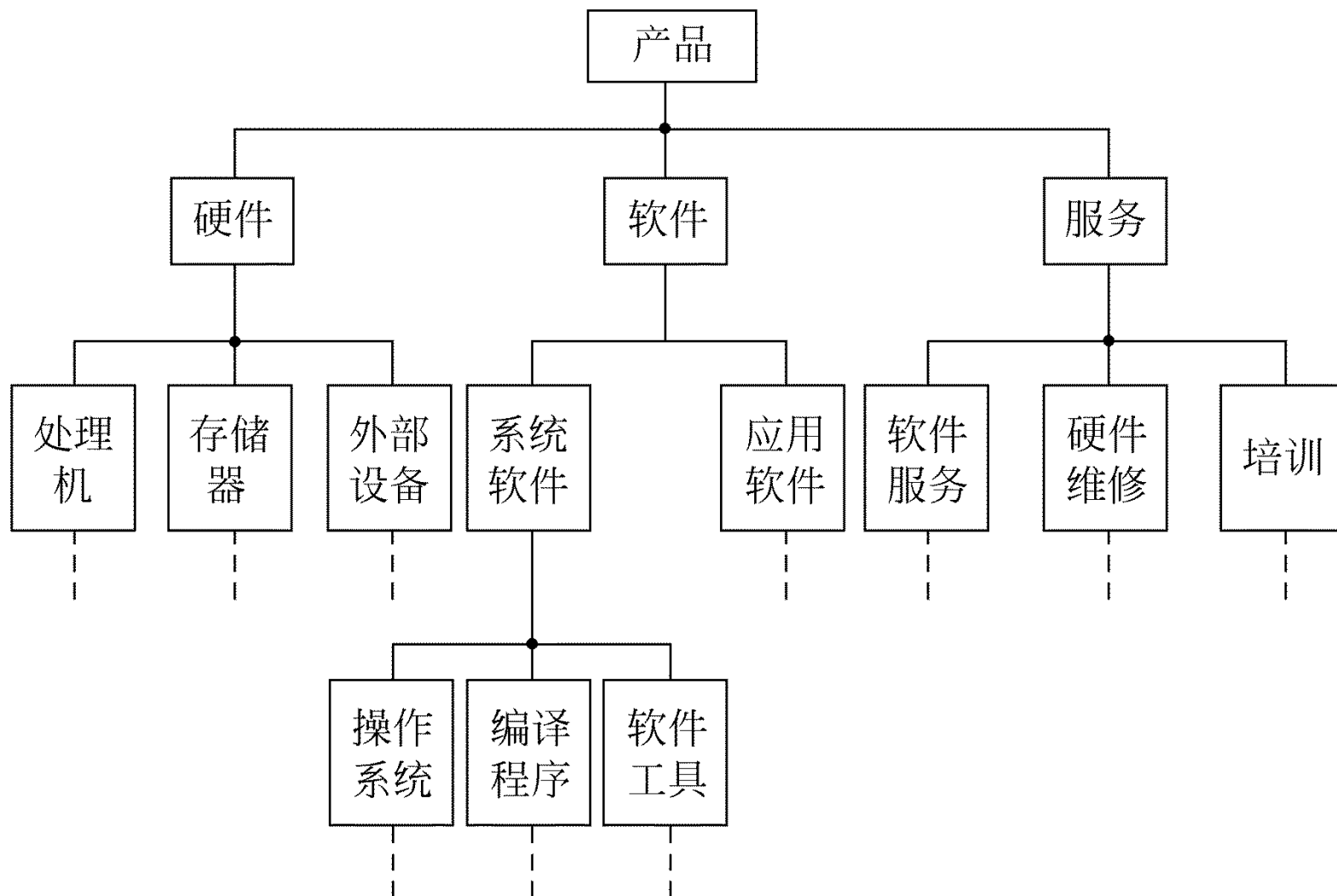


图3.5



## 3.7 其他图形工具

### 3.7.2 Warnier图

和层次方框图类似，Warnier图也用树形结构描绘信息，但是这种图形工具比层次方框图提供了更丰富的描绘手段。

用Warnier图可以表明信息的逻辑组织，也就是说，它可以指出一类信息或一个信息元素是重复出现的，也可以表示特定信息在某一类信息中是有条件地出现的。



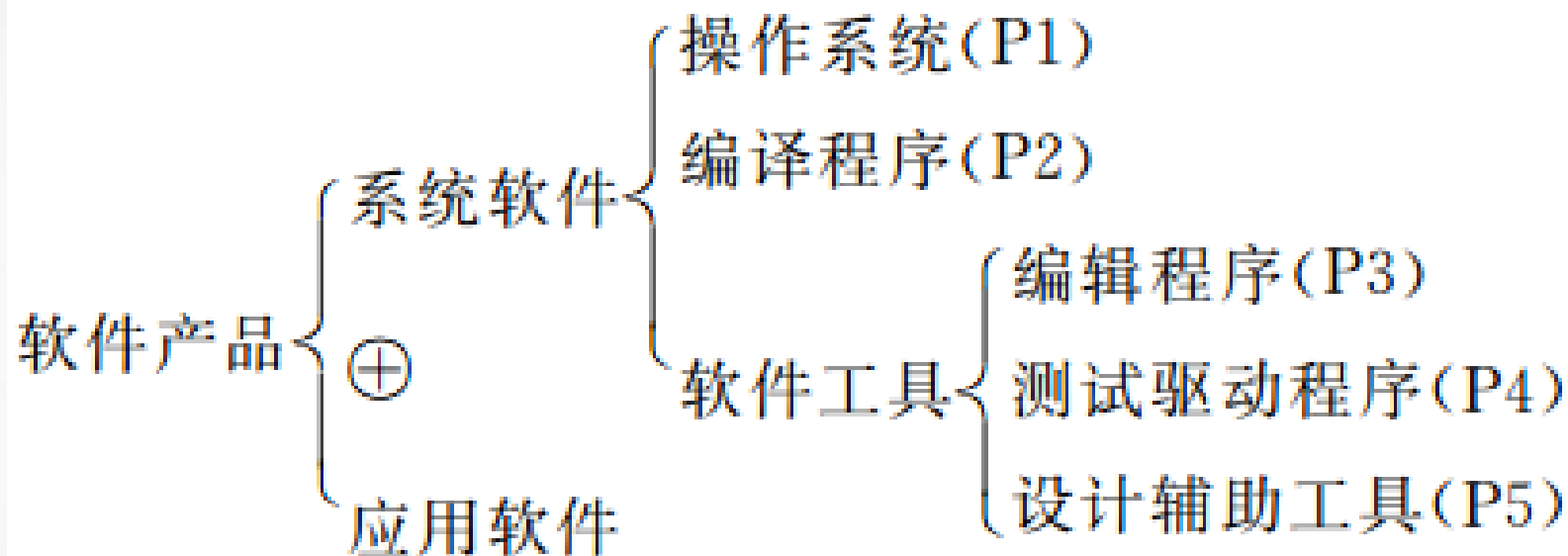


图 3.6 Warnier 图的一个例子

Warnier图描绘一类软件产品的例子



## 3.7 其他图形工具

### 3.7.3 IPO图

IPO图是输入、处理、输出图的简称，它是由美国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。

下图3.7是主文件更新的例子，通过这个例子不难了解IPO图的用法



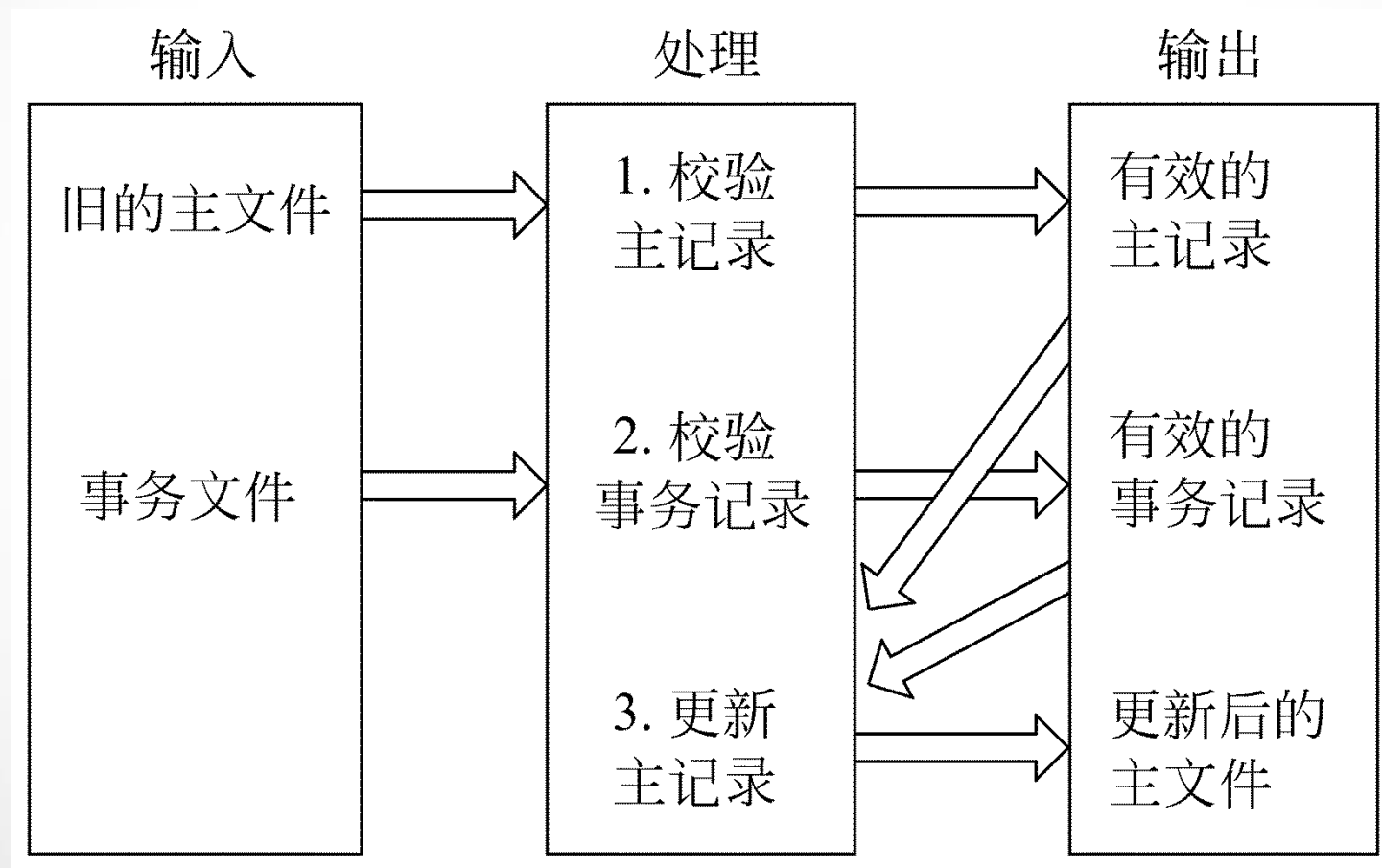


图3.7



## 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：

改进的IPO图



# 主要内容

- 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





## 3.8 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，大量统计数字表明，软件系统中15%的错误起源于错误的需求。

为了提高软件质量，确保软件开发成功，降低软件开发成本，一旦对目标系统提出一组要求之后，必须严格验证这些需求的正确性。一般说来，应该从下述4个方面进行验证。



## 3.8 验证软件需求

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。

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。

现实性指定的需求应该是用现有的硬件技术和软件技术基本上可以实现的。

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 验证软件需求

### 3.8.2 验证软件需求的方法

上一小节已经指出，至少必须从一致性、完整性、现实性和有效性这4个不同角度验证软件需求的正确性。那么，怎样验证软件需求的正确性呢？验证的角度不同，验证的方法也不同。

验证需求的一致性

验证需求的现实性

验证需求的完整性和有效性



## 3.8 验证软件需求

### 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，特别在目标系统规模庞大、规格说明书篇幅很长的时候，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。



## 3.8 验证软件需求

### 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。



## 3.8 验证软件需求

### 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。然而许多用户并不能清楚地认识到他们的需要(特别在要开发的系统是全新的，以前没有使用类似系统的经验时，情况更是如此)，不能有效地比较陈述需求的语句和实际需要的功能。只有当他们有某种工作着的软件系统可以实际使用和评价时，才能完整确切地提出他们的需要。



## 3.8 验证软件需求

### 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。



## 3.8 验证软件需求

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。

使用这个软件工具能够导出详细的文档。

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。

使用这个软件工具之后, 应该能够改进通信状况。



## 3.8 验证软件需求

PSL/PSA系统是CADSAT（计算机辅助设计和规格说明分析工具）的一部分。其中PSL是用来描述系统的形式语言，PSA是处理PSL描述的分析程序。



## 3.8 验证软件需求

PSL/PSA系统的功能主要有下述4种。

描述任何应用领域的信息系统。

创建一个数据库保存对该信息系统的描述符。

对描述符施加增加、删除和更改等操作。

产生格式化的文档和关于规格说明书的各种分析报告。



# 本章小结

1. 本章讲解了需求分析的重要性，以及如何与用户交谈获取需求
2. 讲述建立模型的方法，在需求分析阶段建立数据模型、功能模型和行为模型。
3. 讲述如何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。
4. 详细阐述状态转换图
5. 最后讲解如何验证软件需求。



本章结束