

## 第三章 软件需求分析□

虽然在可行性研究阶段已经粗略了解了用户的需求，甚至还提出了一些可行的方案，但是可行性研究的基本目的是用较小的成本在较短的时间内确定是否存在可行的解法。因此许多细节被忽略了。然而在最终的系统中却不能遗漏任何一个微小的细节，所以可行性研究并不能代替需求分析，它实际上并没有准确地回答“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。软件需求分析是一个不断进行揭示和判断的过程。

### 3.1 需求分析的任务

#### 3.1.1 确定系统的综合要求

##### 1 功能需要

划分出系统必须完成的所有功能

##### 2 性能需要

系统必须满足的定时约束或容量约束  
速度（系统的响应时间）

信息速率

主存容量

磁盘容量

安全性

#### 3.1.2 分析系统的数据要求□

任何一个软件系统其本质上都是一个信息处理系统，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概貌，同时也对软件设计有着深远的影响。分析系统的数据要求，这是软件需求分析的一个重要任务。分析系统的数据要求通常采用建立数据模型的方法

系统的数据来源和去处一般含如下几个方面：

- (1) 从系统以外来，再到系统以外去；□
- (2) 从系统以外来，再到系统内部去；□
- (3) 从系统内部来，再到系统内部去；□
- (4) 从系统内部来，再到系统外部去。

#### 3.1.3□导出系统的逻辑模型□

用数据流图、实体 -- 关系图、状态转换图、数据字典、

主要的处理算法描述逻辑模型。

#### 3.1.4 □修正系统开发计划□

准确地估计系统的成本及进度，修正以前我们所制定的开发计划。□

#### 3.2 与用户沟通获取需求的方法

##### 3.2.1 访谈

情景分析技术就是分析对用户将来使用目标解决问题的方法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

##### 3.2.3 简易的应用规格说明技术

##### 3.2.4 快速建立软件原型

#### 3.3 分析建模与规格说明□

##### 3.3.1 分析建模

1 建模：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。模型由一组图形符号和组织这些符号的规则组成

##### 2 模型与工具

数据模型—实体—关系图

功能模型—数据流图

行为模型—状态转换图□

##### 3.3.2 软件需求规格说明

用自然语言完整、准确、具体描述系统的数据需求、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、将来可能提出来的需求

#### 3.4 实体—关系图□

##### 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。复合信息是指具有一系列不同性质或属性的事物，仅有单个值的事物不是数据对象。

数据对象可以是外部实体、事物、行为、事件、角色、单位、地点、结构

数据对象彼此间是有关联的，它只封装了数据，没有对数据的操作

### 3.4.2 属性

定义了数据对象的性质，属性用标识符表示

### 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。联系分为 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。

(3) 多对多联系 (M:N)

例如，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。

### 3.4.4 实体 - 关系图的符号

实体 - 关系图简称 ER 图

### 3.5 数据规范化

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，通常需把数据结构规范化。

通常用“范式”定义消除数据冗余的程度。第一范式数据冗余程度最大，第五范式数据冗余程度最小。但是，范式的级别越高，存储同样数据就需要分解成更多张表，因此，‘存储自身’的过程也就越复杂。

第二，随着范式级别的提高，数据的存储结构与基于问题域的结构间的匹配程度也随之下降，因此，在需求变化时数据的稳定性较差。

第三，范式级别提高则需要访问的表增多，因此性能（速度）将下降。从实用角度看来，在大多数场合选用第三范式都比较恰当。

下面给出第一、第二和第三范式的定义：

(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：符合第二范式的条件，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局性的进一步描述（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将做哪些动作。因此，状态图满足了行为建模的机制。

#### 3.1 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有改变动作。

状态图既可以表示系统循环运行过程，也可以表示单程生命期。

#### 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从一个状态转换的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转换状态的控制信息。

#### 3.6.3 符号

#### 3.6.4 例子

### 3.7 其他图形工具□

#### 3.7.1 层次方框图□

层次方框图是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。树形结构的顶层是一个单独的矩形框，它代表完整的数据结构。下面各层的矩形框代表这个数据的子集，最低层的各个框代表组成这个数据的实际数据元素（不可再分割）。描绘一家计算机公司全部产品的数据结构图如下。 谨

#### 3.7.2 Warnier 图□

Warnier 图是由法国计算机科学家 J.D.Warnier提出的表示信息层次结构的另外一种图形工具。□

在 Warnier 图中花括号用来区分数据结构的层次，在一个花括号内的所有名字都属于同一类信息；符号表示在其上、下方的名字中的一个名字；名字右边圆括号中的符号表示这个名字在信息类中重复出现的次数。□

### 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，可以方便地表示输入数据、数据处理和输出数据三者之间的关系。□

## 3.8 验证软件需求□

### 3.8.1 验证软件需求的途径与方法

一致性：在所有需求中，任何一条需求不能和其他需求互相矛盾。形式化描述□

完整性：软件规格说明书必须包括用户需求的每一个功能或性能。原型

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。仿真和模拟

有效性：软件需求确实能解决用户所面对的问题。原型□

### 3.8.3 用于需求分析的软件工具□

PSL / PSA 系统用描述符从系统信息流、系统结构、数据结构、数据导出、系统规模、系统动态、系统性质和项目管理等八个方面描述信息系统。

一旦用 PSL 对系统做了完整描述，就可以调用 PSA 产生一组分析报告，其中包括所有修改规格说明数据库的记录，用各种形式描述数据库信息的参照报告（包括图形形式的描述），关于项目管理信息的总结报告，以及评价数据库持性的分析报告。

借助 PSL / PSA 系统可以边对目标系统进行自顶向下的逐层分解，边将需求分析过程中遇到的数据流、文件、处理等对象用 PSL 描述出来并输入到 PSL / PSA 系统中。PSL 将对输入信息作一致性和完整性检查，并且保存这些描述信息。

PSL / PSA 系统的主要优点是它改进了文档质量，能保证文档具有完整性、一致性和无二义性，从而可以减少管理和维护的费用。

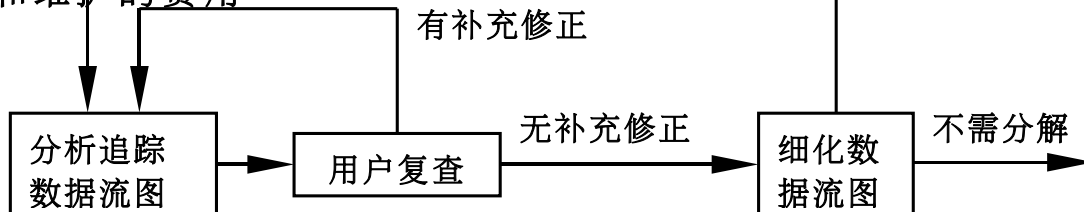


图 3.1 面向数据流自顶向下求精过程

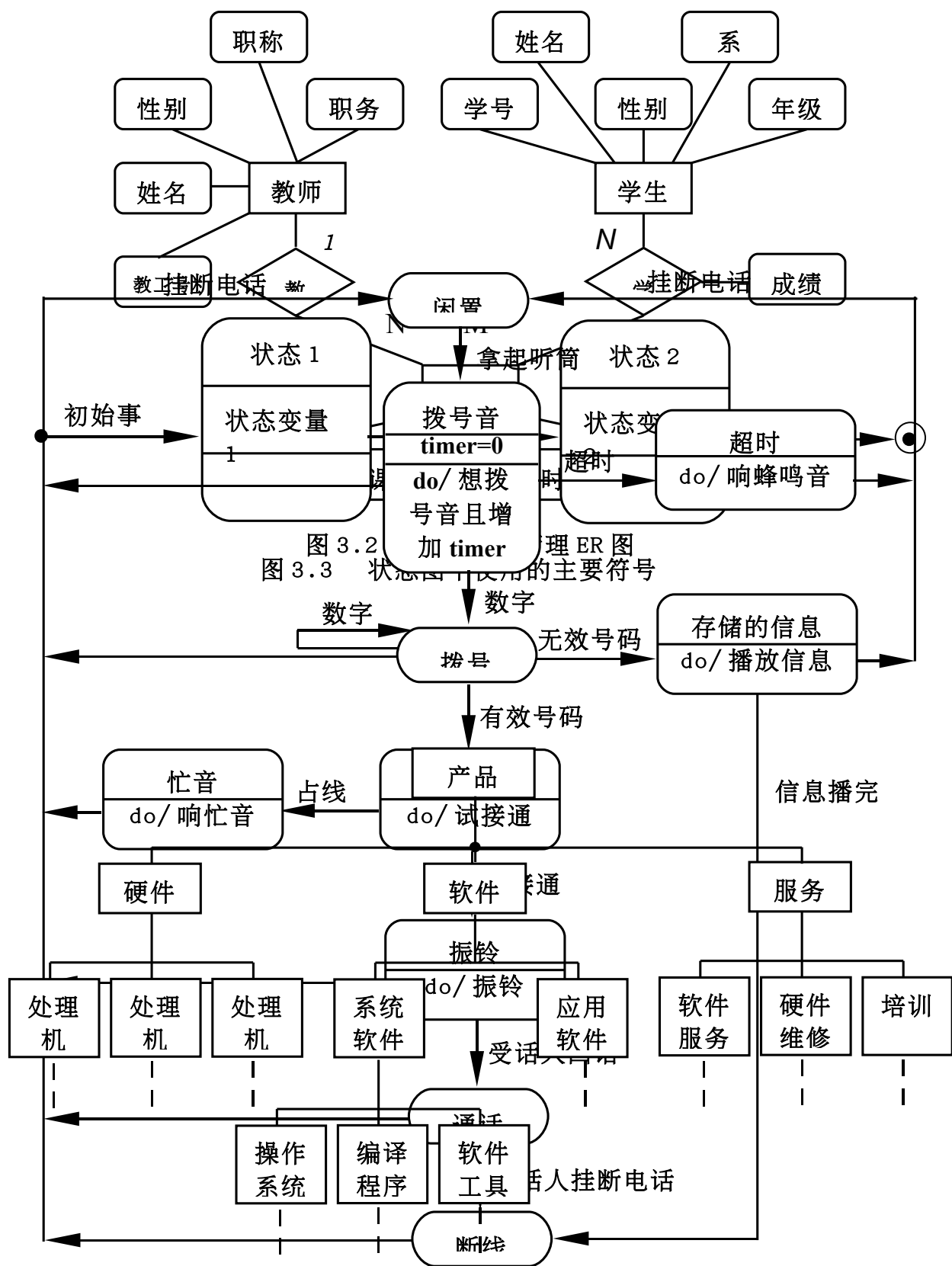


图 3.54 电话系统图的状态图例子

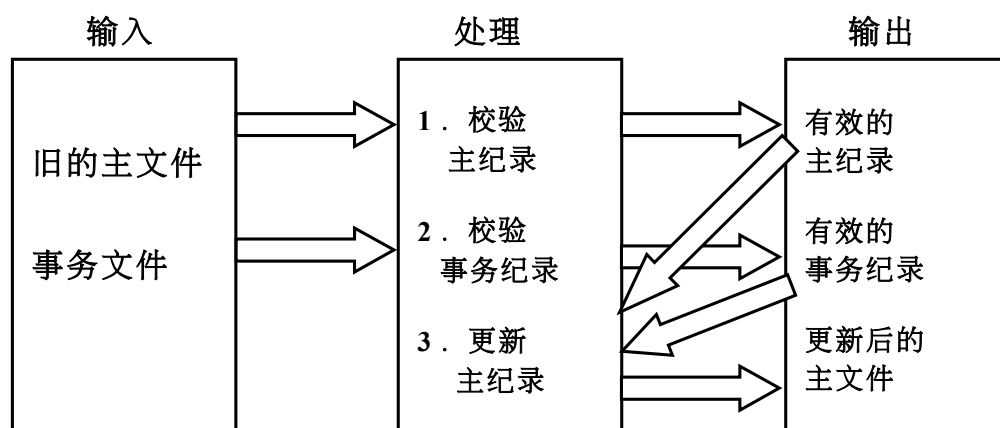
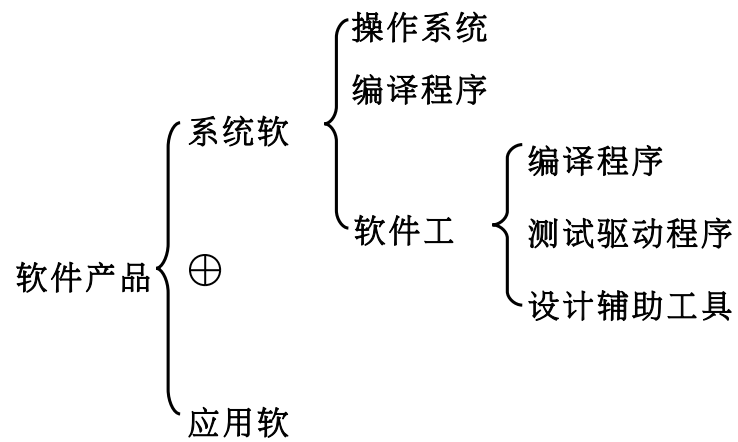


图 3.7 IPO 图的一个例子

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