

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 验证软件需求
- 3.6 小结



需求分析的意义



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，不论我们把设计和编码做得如何出色，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，它的基本任务不是确定系统怎样完成它的工作，**而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，由系统分析员写出**软件需求规格说明书**，以书面形式准确地描述软件需求。即：

—— **准确地回答“系统必须做什么？”**。

分析员和用户的责任:

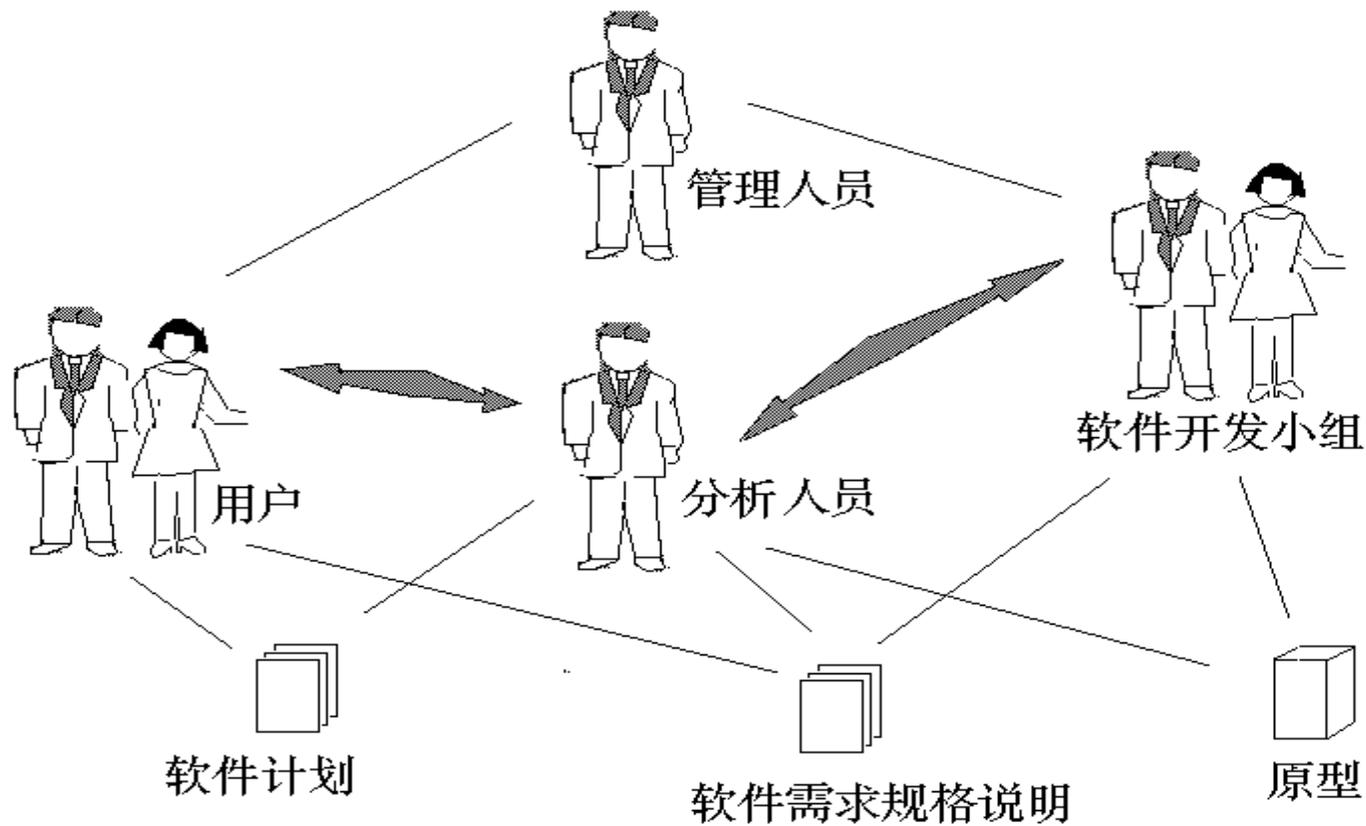
- **分析员**: 又称**系统分析员**、**系统工程师**等。负责沟通用户和开发人员的认识和见解,起着桥梁的作用。
- **分析员应当具备的素质**:
 - ★能够熟练地掌握计算机硬件、软件的专业知识,具有一定的系统开发经验。
 - ★善于进行抽象的思维和创造性的思维。
 - ★善于从相互冲突或混淆的原始资料中抽出恰当的条目来。
 - ★善于调查研究,能够很快学习用户的专业知识,理解用户的环境条件。能够倾听他人的意见,注意发挥其他人员的作用。
 - ★具有良好的书面和口头交流能力。



用户：在软件需求分析中负有重要的责任。

用户对软件的意见和要求是需求分析十分宝贵的原始资料。

- 只有用户才真正知道自己需要什么，但是他们并不知道怎样用软件实现自己的需求，用户必须把他们对软件的需求尽量准确、具体地描述出来；
- 分析员知道怎样用软件实现人们的需求，但是在需求分析开始时他们对用户的需求并不十分清楚，必须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通信技术，集中精力过细地工作，而且必须**严格审查验证需求分析的结果**。



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。



3.1 需求分析的具体任务

1 确定对系统的综合要求

——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、约束等要求。

2 分析系统的数据要求

——建立数据结构模型

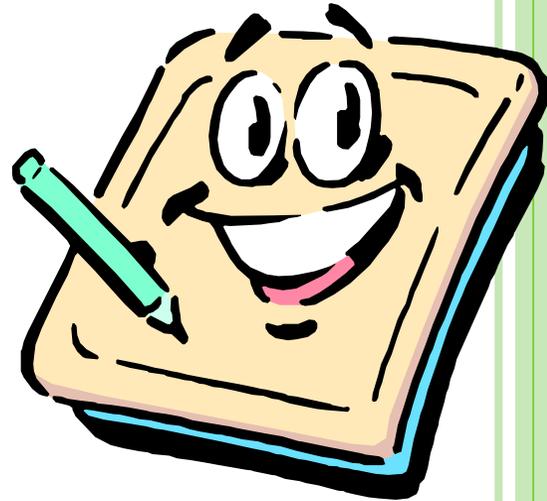
3 导出系统的逻辑模型

4 修正系统开发计划



3.2 获取需求的方法

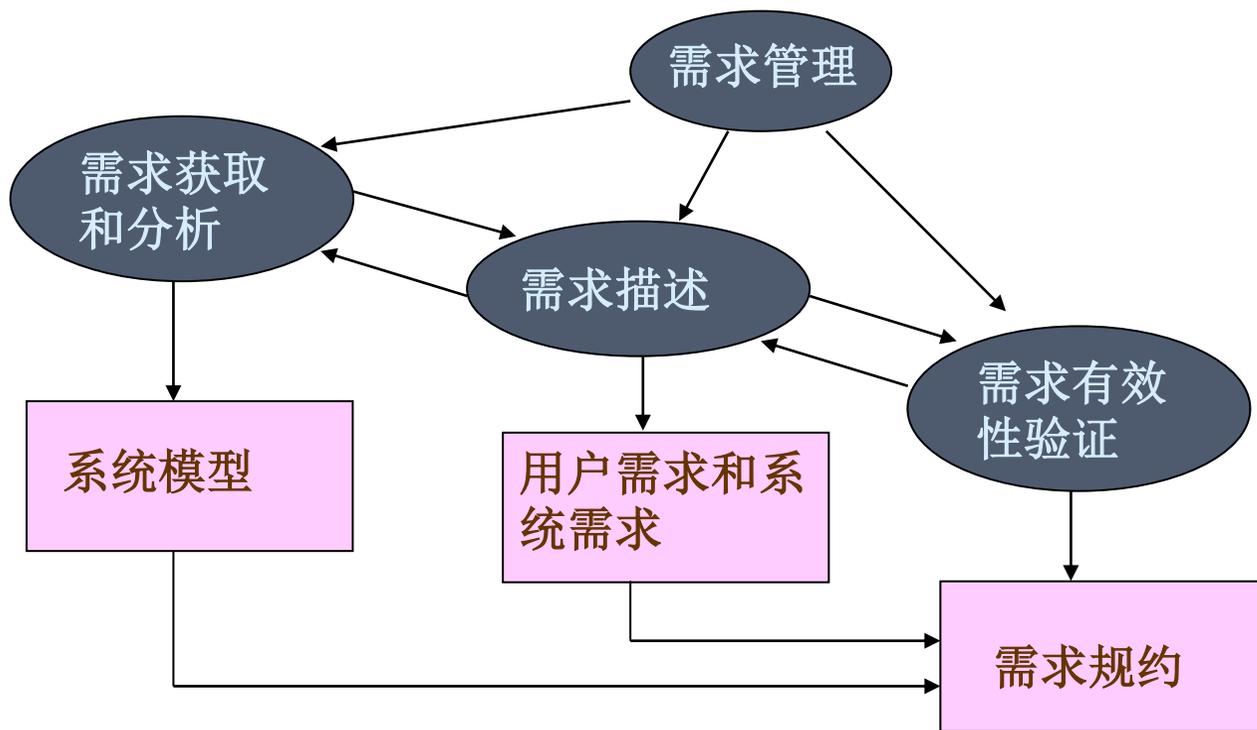
- 3.2.1 访谈
- 3.2.2 面向数据流自顶向下求精
- 3.2.3 简易的应用规格说明技术
- 3.2.4 快速建立软件原型



软件需求获取过程

- 需求分析是一个包括创建和维持系统需求文档所必需的一切活动的过程。它包含了如下活动：

需求获取和分析、需求描述和文档编写、需求有效性验证、需求管理（管理需求工程的变更）。



软件需求过程

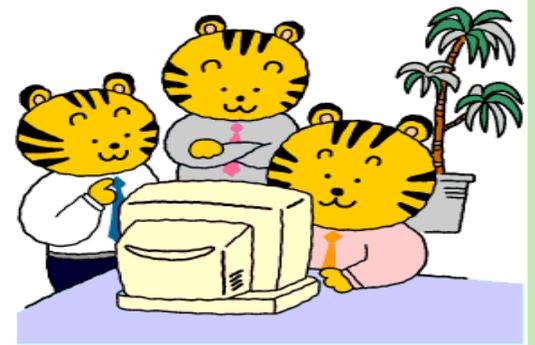


- **需求获取**是开发人员与客户或用户一起对应用领域进行调查研究，收集系统需求的过程。
- **需求分析**是将获取到的需求准确的理解、求精，并将其转化为完整的需求定义（包括建模），进而生成需求规约的过程。
- 需求获取和分析有一定的难度，因为：
客户-项目相关人员通常并不真正知道希望计算机做什么，让他们清晰的表达出需要系统做什么是一件困难的事，他们或许提出不切实际的要求。



3.2.1 需求获取技术-访谈

- 建立由客户（用户）、系统分析员、领域专家参加的**联合小组**。
- 需求获取的方法：个别访谈、召集会议、文档研究、问卷调查、观察用户工作流程、建立原型。
- 获取需求的表达方式：
 - (1) **需求列表**
——需求与系统的特殊视角或环境的关系
 - (2) **业务流程图**（状态/活动图）
 - (3) **数据流图**
 - (4) **实体-联系图**



3.2.2 简易的应用规格说明技术

——面向团队的需求收集法

提倡**用户与开发者**密切合作，共同**标识问题**，提出**解决方案要素**，商讨不同方案并指定基本需求。

- >> 进行初步的访谈
- >> 开发者和用户双方组织的代表出席会议
- >> 每个小组为每张列表中的项目制定小型规格说明
- >> 根据会议成果起草完整的软件需求规格说明书

优点：开发者与用户不分彼此，齐心协力，密切配合，即时讨论并求精，按步骤导出规格说明书。

3.3 分析建模与规格说明

3.3.1 分析建模

(1) 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，由一组**图形符号**和组织这些符号的**规则**组成。

(2) 建模方法

在过去的数年中，人们提出了许多种分析建模的方法，其中两种在分析建模领域占有主导地位：

第一种是结构化分析 (Structured Analysis, SA)，70年代末由DeMarco等人提出，这是传统的建模方法。众多科学家对其进行了扩充，因此它是发展了超过30年的一个混合物。

第二种 面相对象的分析 (Object Oriented, OO)，60年代末在smiula-67首次引入，80年代中期受到关注，90年代成为主流。

软件需求说明书的编写提示（GB856T—88）

1 引言

1.1 编写目的

1.2 背景

1.3 定义

1.4 参考资料

2 任务概述

2.1 目标

2.2 用户的特点

2.3 假定和约束

软件需求说明书的编写提示（GB856T—88）

3 需求规定

- 3.1 对功能的规定
- 3.2 对性能的规定
 - 3.2.1 精度
 - 3.2.2 时间特性要求
 - 3.2.3 灵活性
- 3.3 输入输出要求
- 3.4 数据管理能力要求
- 3.5 故障处理要求
- 3.6 其他专门要求

4 运行环境规定

- 4.1 设备
- 4.2 支持软件
- 4.3 接口
- 4.4 控制

3.4 实体-联系图(ER)

Entity Relationship Diagram



- **ER图** ---- 是用来建立数据模型的工具。
- **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，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 (**实体**)、数据对象的**属性**及数据对象彼此间相互连接的**关系**。

实体-联系图的符号

- 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。

举例:

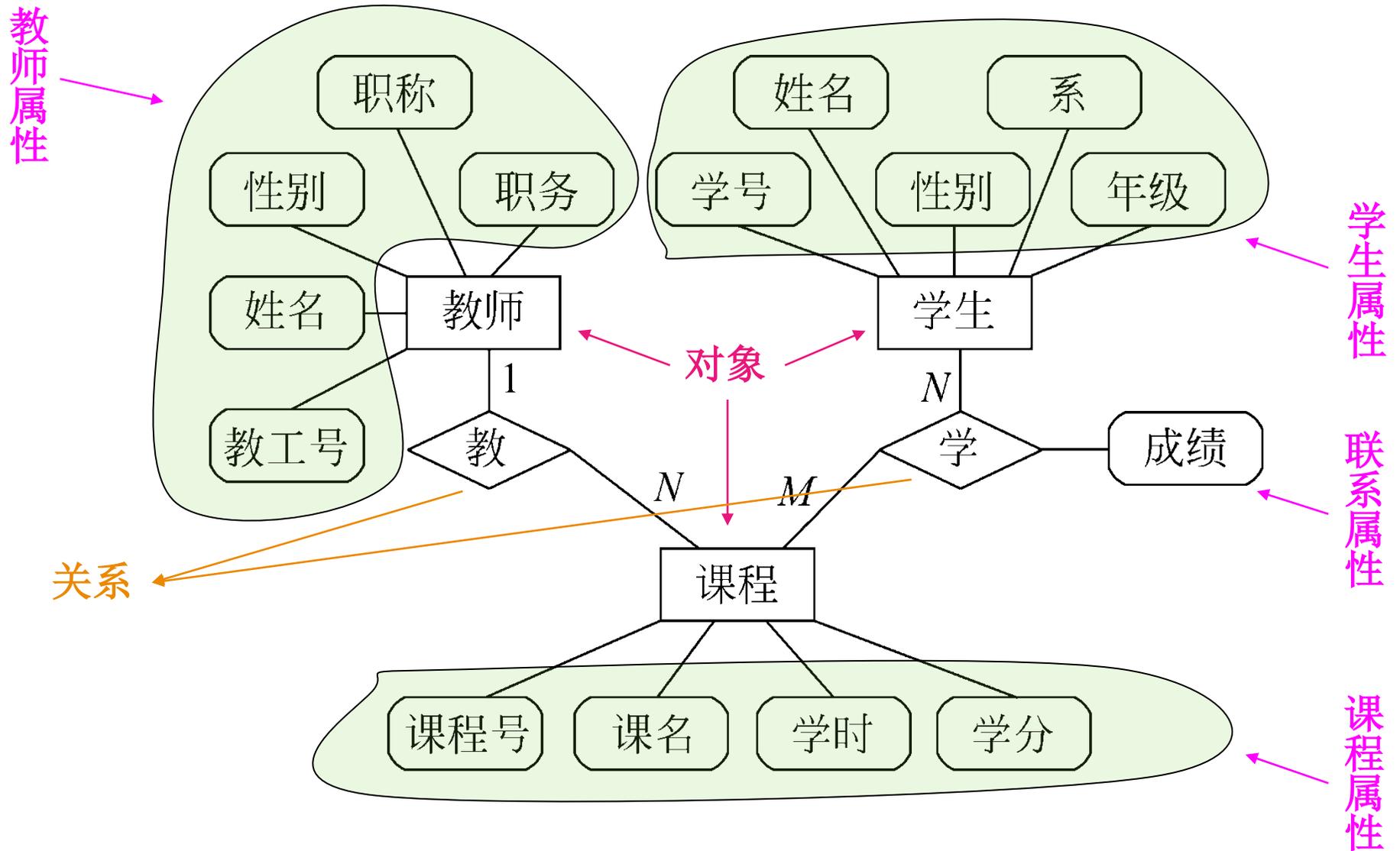
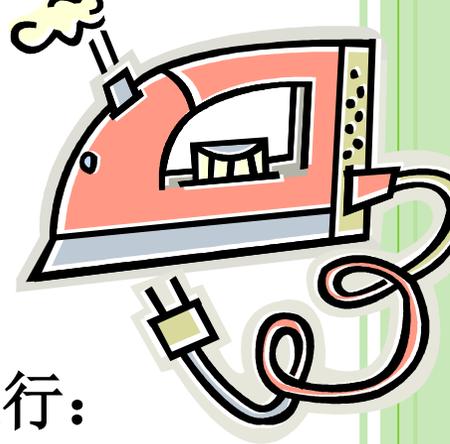


图3.2 某校教学管理ER图

3.5 验证软件需求



3.5.1 验证软件需求的正确性，一般应从4个方面进行：

- (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。

3.5.2 验证软件需求的方法

1. 验证需求的一致性

- 自然语言描述（非形式化）的SRS（Software Requirement Specification）只能靠人工审查，无法保证效果。
- 形式化描述的SRS，可采用软件工具来验证，有效保证一致性。

2. 验证需求的现实性

- 分析员根据经验判断现有软硬件技术实现目标系统的可能性，效果难以保障。
- 采用计算机仿真模拟技术辅助分析，提高效果。

3. 验证需求的完整性和有效性

- 用户积极配合验证，但有许多不确定因素。
- 使用原型系统，让用户试用后判断，提高准确性。

3.5.3 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) 必须采用形式化的语法(或表)描述SRS，才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。

3.5.3 用于需求分析的软件工具

- 作为需求工程方法学的一部分，在1977年美国密歇根大学设计完成了RSL(Requirement Statement Language) (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。

3.5.3 用于需求分析的软件工具

- 1977年美国密歇根大学开发了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需求陈述语言。
- 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。
- 用PSL描述的系统属性放在一个数据库中。一旦建立起数据库之后即可增加信息、删除信息或修改信息,并且保持信息的一致性。
- PSA对数据库进行处理以产生各种报告,测试不一致性或遗漏,并且生成文档资料。

3.5.3 用于需求分析的软件工具

PSL/PSA(问题陈述语言/问题陈述分析程序)系统的功能主要有下述4种：

- (1) 描述任何应用领域的信息系统；
- (2) 创建一个数据库保存对该信息系统的描述符；
- (3) 对描述符施加增加、删除和更改等操作；
- (4) 产生格式化的文档和关于规格说明书的各种分析报告。

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8个方面描述信息系统。

3.5.3 用于需求分析的软件工具

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，以及评价数据库特性的分析报告。



借助**PSL/PSA系统**可以一边对目标系统进行**自顶向下的逐层分解**，一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。PSA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。数据存放在数据库中，**便于增加、删除和更改**，这也是它的一个优点。

规范化的描述



规范化的文档

3.6 小结



- >>需求分析的任务: what functions + other requirements
- >>获取需求的方法: visit, prototyping, refinement
- >>分析建模与规格说明: 数据、功能和行为模型
- >>实体-联系图 & 数据规范化
- >>验证软件需求: 一致性、完整性、现实性和有效性