

## 第2章 软件需求工程

软件需求（Software Requirements）作为软件生存周期的第一个阶段，直接影响到整个软件生存周期，其重要性越来越突出，到 20 世纪 80 年代中期，逐步形成了软件工程的子领域——需求工程。进入 90 年代后，需求工程成为软件界研究的重点之一。从 1993 年起，每两年举办一次需求工程国际研讨会（ISRE），1994 年起，每两年举办一次需求工程国际会议（ICRE）。一些关于需求工程的工作小组相继成立，使需求工程的研究得到了迅速进展。

### 2.1 软件需求的基本概念

软件需求无疑是当前软件工程中的关键问题，没有需求就没有软件。美国于 1995 年开始对全国范围内的 8000 个软件项目进行跟踪，调查结果表明，有 1/3 的项目没能完成，而在完成的项目中，又有 1/2 的项目没有成功实施。分析失败的原因后发现，与需求过程相关的原因占了 45%，而其中缺乏最终用户的参与以及不完整的需求又是两大首要原因，各占 13% 和 12%。该例足以说明需求阶段是软件生存周期中重要的第一步，也是决定性的一步。

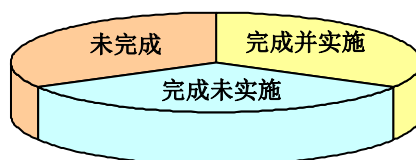


图 2.1 软件项目跟踪

#### 2.1.1 软件需求的任务

在需求分析阶段，要对经过可行性分析所确定的系统目标和功能作进一步的详细论述，以确定系统是“做什么”的。

在传统软件工程生命周期中，涉及软件需求的阶段称作需求分析。需求分析的主要功能任务是：定义软件的范围及必须满足的约束，确定软件的功能和性能及与其他系统成分的接口，建立数据模型、功能模型和行为模型，最终提供需求规格说明，并用于作为评估软件质量的依据。

显然，随着软件需求的不断发展而提出的软件需求工程，其内容更加广泛。需求工程是一个包括创建和维护系统需求文档所必须的一切活动，是对系统应该提供的服务和所受到的约束进行理解、分析、检验和建立文档的过程。

软件需求的基本任务有以下 4 方面：

##### (1) 确定系统的综合要求

- ① 确定系统功能要求。这是最主要的需求，确定系统必须完成的所有功能。
- ② 确定系统性能要求。系统性能根据系统应用领域的具体需求确定，如可靠性、联机系统的响应时间、存储容量、安全性能等。
- ③ 确定系统运行要求。主要是对系统运行时对环境的要求，如系统软件、数据库管理系统、外存和数据通信接口等。
- ④ 将来可能提出的要求。对将来可能提出的扩充及修改预先做准备。

##### (2) 分析系统的数据要求

软件系统本质上是信息处理系统，因此，必须考虑以下要求。

- ① 数据要求。需要哪些数据、数据间联系、数据性质、结构等。

② 数据处理要求。包括处理的类型、处理的逻辑功能等。

### (3) 导出系统的逻辑模型

系统的逻辑模型与开发方法有关。例如，采用结构化分析法（Structured Analysis, SA），可用 DFD 图来描述；若采用面向对象的分析方法（Object-Oriented Analysis, OOA），用例模型（Use case model）则是首选的描述工具。

### (4) 修正系统的开发计划

通过需求对系统的成本及进度有了更精确的估算，可进一步修改开发计划，最大限度地减小软件开发的开发风险。

必须注意的是，对于大、中型的软件系统，很难直接对它进行分析设计，人们经常借助模型来分析设计系统。模型是现实世界中的某些事物的一种抽象表示，抽象的含义是抽取事物的本质特性，忽略事物的其他次要因素。因此，模型既反映事物的原型，又不等于该原型。模型是理解、分析、开发或改造事物原型的一种常用手段。例如，建造大楼前常先做大楼的模型，以便在大楼动工前就能使人们对未来的大楼有一个十分清晰的感性认识，显然，大楼模型还可以用来改进大楼的设计方案。

## 2.1.2 功能需求与非功能需求

软件系统的需求分为用户需求和系统需求两类（如图 2.2 所示），用户需求（User Requirement）包括由组织机构或客户对系统、产品高层次的业务需求（business requirement），或是从用户的角度由用户、管理员等用自然语言描述了用户使用软件产品必须要完成的任务，即是针对应用领域提出的软件所需要解决的问题。用户需求通常只描述了系统的外部行为，而不涉及系统内部的特性。由于自然语言易产生二义性，因此用户所提出的需求往往是较模糊的。

系统需求（system requirement）则较详细地给出系统将要提供的，以及系统所受到的约束。对系统的需求文档的描述应该是精确的。系统需求分为功能需求、非功能需求和领域需求：

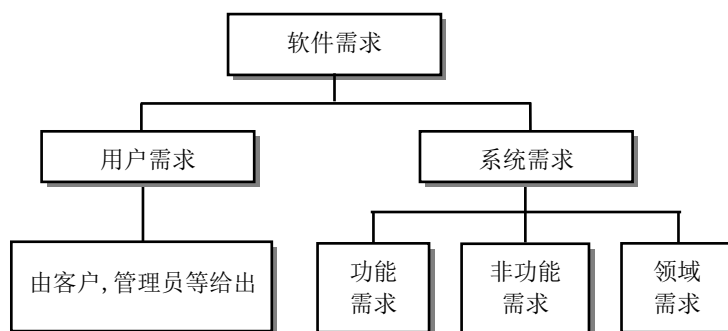


图 2.2 软件系统需求

### 1. 功能需求

**功能需求**是对系统应该提供的服务、功能以及系统在特定条件下的行为的描述。它与软件系统的类型、使用系统的用户等相关,有时需要详细描述系统的功能、输入/输出、异常等,有时还需要申明系统不应该做什么。

例：有一个图书管理系统，该系统除了具备一般的图书管理功能外，还通过网络，为读者提供从其他图书馆借阅图书和文献资料的服务。因此系统应该具备以下功能：

#### (1) 图书管理的基本业务功能

- 读者借、还书籍的登记管理功能，并随时根据读者借、还书籍的情况更新数据库系统，以及书籍的编目、入库、更新等操作。如果书籍已经借出，还可以进行预留操作。
- 与其他图书馆联网，实现对其他图书馆的部分书籍、文献资料的借阅功能。

## (2)基本数据维护及数据库管理功能

对所有图书信息及读者信息进行统一管理维护的功能协调整个图书馆的运作。提供使用者录入、修改并进行维护基本数据的接口。基本数据包括读者的信息、图书资料的相关信息，提供对这些信息进行修改、增加、删除，更新等操作。

## (3)信息查询功能

提供对各类信息的查询功能，如对本图书馆的用户借书信息，还书的信息，书籍源信息，预留信息等进行查询，对其他图书馆的书籍、资料源信息的查询功能。

## 2. 非功能需求

非功能需求常常指不与系统功能直接相关的一类需求，主要反映用户提出的对系统的约束，与系统的总体特性有关，如可靠性、反应时间、存储空间等。由于系统的非功能需求反映了系统的总体特性，因此常常比系统的功能需求更加关键。例如一个载人宇宙飞船的可靠性不符合要求，根本不可能发射。

非功能需求还与系统开发过程有关。图 2.3 描述了非功能需求的分类。

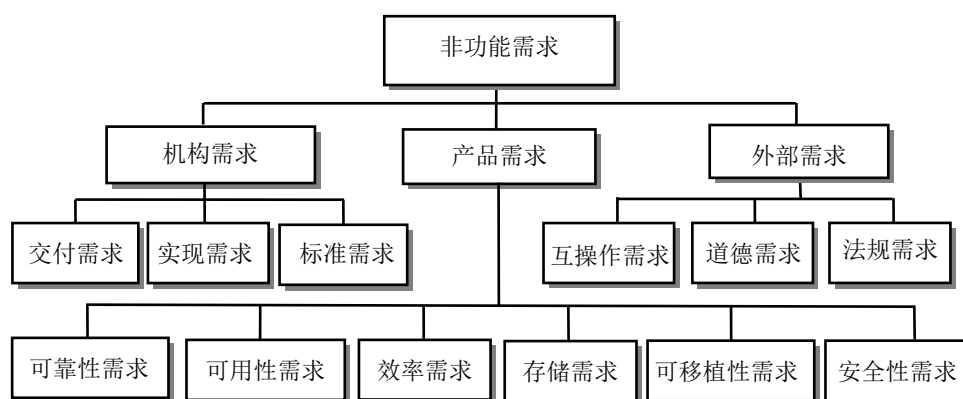


图 2.3 非功能需求类型

机构需求是由用户或开发者所在的机构对软件开发过程提出的一些规范，例如交付、实现、标准等方面的需求。

产品需求主要反映了对系统性能的需求；可用性、可靠性、可移植性、效率和存储等方面的需求，直接影响到软件系统的质量；安全性需求则将关系到系统是否可用的问题。

外部需求的范围较广，包括所有系统的外部因素及开发、运行过程。互操作需求是指该软件系统如何与其他系统实现互操作；法规需求和道德需求确保系统在法律允许的范围内工作和保证系统能够被用户和社会公众所接受。

还要注意，一般对非功能需求进行量化是比较困难的，因此，对非功能需求的描述往往是模糊的，对其进行验证也是比较困难的。

由于非功能需求和功能需求之间存在着相互作用的关系，在进行需求描述时，有时很难将它们区分开，这将影响到对功能和非功能指标的分析。但如果能够将两者区分开来，却又很难看出他们之间的相互关系，在具体进行软件需求分析时要根据所开发的软件系统的类型和具体情况确定。

例如：对上述“图书管理系统”可提出以下非功能需求：

① 系统安全性需求：为保证系统安全性，对本图书馆的各项功能进行分级、分权限操作，对各类用户进行确认。对其它图书馆借阅图书和文献资料的服务控制访问范围：如限 IP、限用户等。

② 对系统可用性的需求：为了方便使用者，要求对所有交互操作提供在线帮助功能。

③ 对系统查询速度的需求：要求系统在15S之内响应查询服务请求。

④ 对系统可靠性的需求：要求系统平均失败发生率小于1%。

### 3 领域需求

领域需求是由软件系统的应用领域所决定的特有的功能需求,或者是对已有功能的约束,如果这些需求不满足,会影响系统的正常运行。

例如:对“图书管理系统”,提出一些与图书管理的业务相关的需求:

(1) 图书编目要求遵循我国图书管理的规定,按照《中国图书馆分类法》进行,执行对图书的分类管理的标准。

(2) 版权法对图书馆文献资料的保护的需,对一类文献资料有限制的使用和服务。对某些文献资料只能在图书馆规定的阅览室阅读,并限制复制和打印。

## 2.2 需求工程过程

需求工程是系统工程和软件工程的一个交叉分支,涉及到软件系统的目标、软件系统提供的服务、软件系统的约束和软件系统运行的环境,如图 2.4 所示,它还涉及这些因素和系统的精确规格说明以及系统进化之间的关系。图 2.4 描述了需求工程所涉及的各种因素。

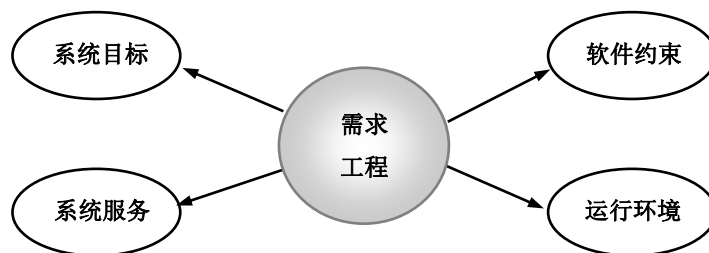


图 2.4 需求工程涉及因素

### 2.2.1 需求工程的基本活动

需求工程是一个不断反复进行需求定义、记录和演进的过程,需求工程过程包括如下主要活动:

(1) **获取需求。**深入实际,在充分理解用户需求的基础上,积极与用户交流,捕捉、分析和修订用户对目标系统的需求,并提炼出符合解决领域问题的用户需求。

(2) **需求分析与建模。**对已获取的需求进行分析和提炼,进行抽象描述,建立目标系统的概念模型,进一步对模型(原型)进行分析。

(3) **需求规格说明。**对需求模型进行精确的、形式化的描述,为计算机系统的实现提供基础。

(4) **确认需求。**以需求规格说明为基础输入,通过符号执行、模拟或快速原型等方法,分析和验证需求规格说明的正确性和可行性,确保需求说明准确、完整地表达系统的主要特性。

(5) **需求管理。**在整个需求工程过程中,贯穿了需求管理活动。需求管理主要包括跟踪和管理需求变化,支持系统的需求演进。由于客户的需要总是不断(连续)增长的,进化需求是十分必要的。

需求工程活动主要围绕着需求的获取和分析、文档的编写和需求验证进行。实际上,在软件开发过程中,所有的需求都有可能变化,为了有效地控制和适应需求的变化,需求管理,主要是对需求变更的管理就成为需求工程过程的另外一类重要活动。

图2.5描述了需求工程过程的主要活动及其它它们之间的关系。

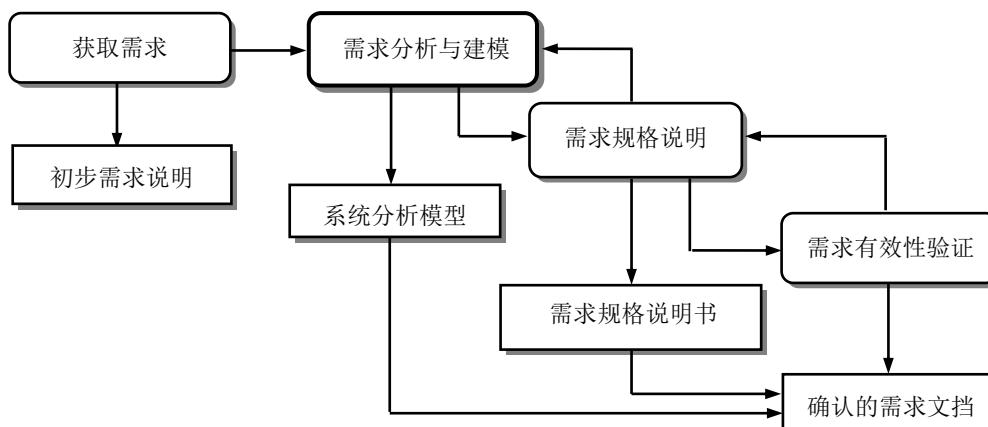


图 2.5 需求工程过程

## 2.2.2 需求的获取和分析

需求的获取和分析是需求工程的关键和核心步骤，直接影响到后期的开发工作和系统的成败。然而，需求的获取和分析却非常困难。

为了取得该项活动的成功，需要所有项目相关人员的共同努力，充分交流和协作，需要用户与软件开发人员一起对应用领域进行调查研究，明确软件系统应该提供的服务及应该具备的性能，确定系统可能的限制和约束。

### (1) 需求获取

在深入实际调查研究，充分理解用户需求的基础上，获取系统需求。获取过程为：

① 了解领域知识，工程技术人员需要依靠领域专家，学习和理解相关的专业知识，才能正确抽取用户需求。

② 需求收集，与项目相关人员进行沟通，在进一步了解专业领域的基础上，发现系统需求的过程。

### (2) 需求分析

需求分析的过程是对收集到的需求进行提炼、分析和审查的过程，最终确定需求，并确保所有项目相关人员对需求取得一致性认识。分析阶段的主要工作包括：

① 确定系统范围。确定系统与其他外部实体或其他系统的边界和接口。

② 分类排序。对所收集的需求进行重新组织整理、分类和筛选，并对每类需求进行排序，确定哪些是最重要的需求。

③ 建立需求分析模型。这是分析阶段的核心工作，需求分析模型是对需求的主要描述手段，是根据不同的分析方法建立的各种视图，例如数据流图（DFD）、实体关系图（E-R）、用例图（Use Case）、类图、状态图、各种交互图等。还可建立辅助的说明，如数据词典。

④ 建立需求规格说明。是将需求的结果按照规定的格式用文档形式描述出来。需求规格说明在整个开发过程中具有很重要的作用，是用户和开发人员之间进行交流和理解系统的手段。用户通过需求规格说明检查是否符合和满足所提出的全部需求。开发者则通过需求规格文档，了解和理解所开发系统的内容，并以此作为软件设计和软件测试的依据。项目管理人员以它为依据，规划软件开发过程、计划，估算软件成本和控制需求的变更过程。

## 2.2.3 需求的有效性验证

### (1) 需求验证的重要性

需求分析是软件开发的第一阶段，直接影响后面各阶段的开发。需求的有效性验证是保证需求说明准确、完整地表达系统需求的重要步骤。需求验证是通过评审方式，检查需求文档中存在的错误和缺陷，验证所确定的需求是否与用户的所有需求相符合，是否具有二义性，以及这些需求能否实现，以便及早发现问题，确保对需求变更的有效控制，确保后期开发工作能够顺利进行。

### (2) 需求验证的内容

需求验证主要围绕需求规格说明中有关软件的质量特性展开，其内容如下。

① **正确性**。正确性是指需求规格说明所确定的系统功能、行为、性能等与用户所提出的需求完全一致，代表了用户的真正需求，并且描述不存在二义性。

② **完整性**。完整性指需求规格说明包含了软件系统应该完成的全部任务和约束，没有遗漏的功能或业务过程及非功能的需求，规格说明包含了功能之间的接口和消息传递。

③ **一致性**。在需求规格说明中，需求不应该冲突，对需求的描述应该一致；包括对术语的使用、功能和性能的描述和时序不能前后矛盾。

④ **可修改性**。由于需求是变动的，需求规格说明的可修改性，就保证了系统对需求变动的适应能力。可修改性包括规格说明的格式、组织方式、内容等都易于修改。

⑤ **可验证性**。可验证性是指需求规格说明所描述的需求应该总是可以检验的，也就是说，总是可以设计出一组有效的检查方法来验证所描述的需求。

必须说明，需求的有效性验证并不是一件容易的事，要验证一组需求是否符合用户的需要是很困难的。而且，需求的有效性验证也不能发现所有的需求问题。

需求验证的方法可以是非正式的，也可以是正式的。非正式的验证，是开发者与用户充分讨论交流的过程，开发者反复向用户进行需求的说明解释，以取得一致认识。在此基础上，进行正式验证，通常以评审会的形式进行，对上述需求验证的内容逐条进行检查，找出问题，协商解决。

## 2.2.4 需求管理

需求管理贯穿软件需求工程全过程，而且与需求工程的其他过程密切相关，因此有效实施需求管理是项目成功的基础。需求工程过程存在两大难题，一是需求的确定是困难的，二是需求是不断变动的。尤其是对于一些大型软件系统，开发周期长，系统规模大，复杂性高。例如 Windows 2000 仅核心开发人员就多达 2500 多人，如此大规模的软件，在其开发过程中，需求是不可能不变的。

因此，需求管理的主要任务是：

### 1. 需求变更的控制

需求的变更又称为进化需求，需求的变更是造成软件开发困难的根本原因之一。进化需求的必要性是明显的，因为客户的需要总是不断（连续）增长的，但是一般的软件开发又总是落后于客户需求的增长，如何管理需求的变更就成为软件进化的首要问题。

需求管理的主要任务是对系统需求变更进行跟踪和控制。对传统的变化管理过程来说，其基本内容包括软件配置、软件基线和变更审查等。

当前推出了一些新的管理方法，如软件家族法，即软件产品线方法。该方法是源于工业界产品线的概念，关注一个软件企业如何组织一组可管理的、具有共性特征的相似软件产品的生产，并应用软件复用的相关原理与技术，将软件的开发分为核心资产的开发，基于核心资产的集成。此外，多视点方法也是管理需求变化的一种新方法，它可以用于管理不一致性并进行关于变化的推理。是从多个视点出发在软件工具的协助下对需求描述，进行自动需求建模，从而提高需求模型的完整性。

图2.6描述了需求变更管理的主要过程：

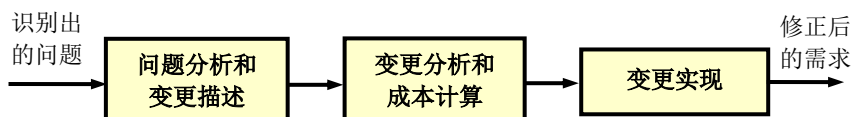


图 2.6 需求变更管理过程

必须对识别出的需求问题或提出的需求变更进行分析，以检查它的有效性和可行性。

对需求变更可能产生的影响进行评估，计算需求变更成本，包括需求文档变更的成本，设计和实现的成本也应重新估算，在此基础上做出变更的决策和规划。

## 2.版本控制

变更的实现，主要涉及对需求文档的修改，系统设计及实现方案的修改。为了尽量减少修改量，在建立初始需求文档时，就要考虑到其易修改性、易扩充性。

版本控制，要确定文档版本，记录所有文档版本变化的相关信息，包括变更的内容、变更的日期、变更的原因、处理变更的情况、变更人等。

一般需求管理过程需要CASE工具支持。

## 2.3 软件需求获取技术

软件开发工作只有基于用户需求，立足于可行的技术才有可能成功。需求的获取过程，是用户和开发者密切交流的结果。

### 2.3.1 需求获取的困难

软件需求是软件工程中最复杂的过程之一，需求的获取是十分困难的，其主要原因是：

- (1) 应用领域的广泛性，它的实施无疑与各个应用行业的特征密切相关。
  - (2) 非功能性需求建模技术的缺乏及其与功能性需求有着错综复杂的联系，大大增加了需求工程的复杂性。
  - (3) 沟通上的困难。由于系统分析员、需求分析员等各方面人员有不同的着眼点和不同的知识背景，给需求工程的实施增加了人为的难度。
  - (4) 需求的变化是造成软件需求复杂性的一个重要原因，尤其对于大型系统，需求的变化总是存在的，如果不能有效地控制和适应需求的变化，将会影响到软件项目的成败。
- 因此必须采用相应技术和方法，才能够正确地获取需求。

### 2.3.2 需求获取方法

需求获取的方法很多，下面介绍几种常用的获取技术。

#### 1. 面谈法

这是一种重要而直接简单的发现和获取需求的方法，而且可以被随时使用。面谈的对象主要有用户和领域专家：与用户面谈主要了解和提取需求，这需要反复进行；与领域专家面谈，是一个对领域知识的学习和转换过程，开发人员如果没有足够的领域知识，是不可能进行成功开发的。使用该方法时应注意以下问题和面谈技巧，才能取得好的效果。

- (1) 面谈前的准备要充分，事先拟订谈话提纲，列出面谈的问题，可包括以下内容：



① 用户或客户的情况表；② 当前存在和需要解决的问题；③ 了解主要用户群体的环境，包括教育背景、计算机应用和使用的水平；用户多少；④ 用户对可靠性、性能有何需求；⑤ 对安全性有无特殊的要求；⑥ 对服务和支持有何要求；

**(2) 面谈后注意认真分析总结**，在对面谈内容进行整理的基础上，提出初步需求并请用户评估。

**(3) 注意掌握面谈的人际交流技能**，这是面谈能否成功的重要因素，在交谈过程中注意耐心认真倾听面谈对象叙述，又要控制面谈过程。

## 2. 问卷法调查法

问卷法是指开发方就用户需求中的一些个性化的、需要进一步明确的需求（或问题），通过采用向用户发问卷调查表的方式，达到彻底弄清项目需求的一种需求获取方法。是从多个用户中收集需求信息的有效方式，是对面谈法的补充。这种方法适合于开发方和用户方都清楚项目需求的情况。

一般问卷设计应该采用以下形式：

- ① 多项选择问题，用户必须从提供的多个答案中选择一个或者多个答案。
- ② 评分问题，可以提供分段的评分标准；如很好、好、一般、差等。
- ③ 排序问题，对回答的问题给出排列的序号或以百分比形式排序。

这种方法比较简单、侧重点明确，因此能大大缩短需求获取的时间、减少需求获取的成本、提高工作效率。

## 3. 会议讨论法

所谓会议讨论法，是指开发方和用户方召开若干次需求讨论会议，达到彻底弄清项目需求的一种需求获取方法。例如 Windows 2000 在开发过程中，项目关键人员每周七天都要开会研究，有时一天多达三次会议。

这种方法适合于开发方不清楚项目需求，如开发方是刚开始做这种业务类型的工程项目，但用户方清楚项目需求的情况，并能准确地表达出他们的需求，而开发方有专业的软件开发经验，对用户提供的需求一般都能准确地描述和把握。

这种方法的一般操作步骤是：

- ① 开发方根据双方制定的《需求调研计划》召开相关需求主题沟通会；
- ② 会后开发方整理出《需求调研记录》提交给用户方确认；
- ③ 如果此主题还有未明确的问题则再次沟通，否则开始下一主题；
- ④ 所有需求都沟通清楚后，开发方根据历次《需求调研记录》整理出《用户需求说明书》，提交给用户方确认签字。

由于开发方不清楚项目需求，因此需要花较多的时间和精力进行需求调研和需求整理工作。

## 4. 原型法

对于某些试验性、探索性的项目，更是难于得到一个准确、无二义性的需求。而原型化方法（Prototyping Method）是获取这一类项目需求的有效方法。

原型化方法强调的是软件开发人员与用户的不断交互，使用户及早获得直观的学习系统的机会，通过原型的不断循环、迭代、演进，不断适应用户任务改变的需求，在原型的不断演进中获取准确的用户需求。将维护和修改阶段的工作尽早进行，使用户验收提前，从而使软件产品更加实用。

原型法的主要优点在于它是一种支持用户的方法，使得用户在系统生存周期的设计阶段起到积极的作用；它能减少系统开发的风险，特别是在大型项目的开发中，由于对项目需求的分析难以一次完成，应用原型法效果更为明显。

原型法的概念既适用于系统的重新开发，也适用于对系统的修改；快速原型法要取得成功，要求有像第四代语言（4GL）这样的良好开发环境/工具的支持。原型法可以与传统的生命周期方法相结合使用，这样会扩大用户参与需求分析、初步设计及详细设计等阶段的活动，加深对系统的理解。

## 5. 面向用例的方法



随着面向对象技术的发展，使用“用例”来表达需求已逐步成为主流，分析建立“用例”的过程，也就是提取需求的过程。

用例是对一组动作序列（其中包括它的变体）的描述，系统执行该动作序列来为参与者产生一个可观察的结果值，这个动作序列就是业务工作流程 Z。

用例是对用户目标或用户需要执行的业务工作的一般性描述；是一组相关的使用场景。描述了系统与外部角色之间的交互。使用场景（usage scenario）则是某个用例的一条特定路径，是用例的特定的实例。通过用例描述，能将业务的交互过程用类似于流程的方式文档化。阅读用例能了解交互流程。

用例特别适宜描述用户的功能性需求，它描述的是一个系统做什么（What），而不是说明怎么做（How），用例不关心系统设计。

用例特别适用于增量开发，一方面通过优先级指导增量开发，另一方面用例开发的本身也是强调采用迭代的、宽度优先的方法进行开发，即先辨认出尽可能多的用例（宽度），再细化用例中的描述，再回过头来看还有哪些用例（下一次迭代）。

## 2.4 需求分析与建模

需求工程的目的是通过与用户广泛地交流确定应用系统的目标。需求活动以“工程化”的方法被提出、分析和组织，它鼓励用户以一种积极的方式参与需求分析活动，并在整个软件生存周期强调用户参与和领域专家的指导作用，促使目标系统更好地满足用户需求。

### 2.4.1 需求分析方法

需求分析是需求工程的关键步骤。常见的、有效的需求分析方法和描述方式都要考虑便于用户的参与和理解。常用的分析方法有：

① **功能分析方法**。将系统看作若干功能模块的集合，每个功能又可以分解为若干子功能，子功能还可继续分解。分解的结果已经是系统的雏形。

② **结构化分析方法**。它是一种以数据、数据的封闭性为基础，从问题空间到某种表示的映射方法，用数据流图（Data Flow Diagram, DFD）表示。

③ **信息建模法**。它是从数据的角度来建立现实世界的模型。大型信息系统通常十分复杂，很难直接对它进行分析设计，人们经常借助模型来设计分析系统。

模型是开发过程中的一个不可缺少的工具。信息系统包括数据处理、事务管理和决策支持。实质上，信息系统可以看成是由一系列有序模型构成的，这些有序模型通常为功能模型、信息模型、数据模型、控制模型和决策模型，所谓有序是指这些模型是分别在系统的不同开发阶段、不同开发层次上建立的。建立信息系统常用的基本工具是 E-R 图。

④ **面向对象的分析方法(Object-Oriented Analysis, OOA)**。其关键是识别问题域内的对象，分析它们之间的关系，并建立起三类模型：对象模型、动态模型和功能模型。

下面以结构化分析方法和面向对象的分析方法为例，介绍需求分析的方法。

### 2.4.2 结构化分析（SA）方法

结构化开发方法（Structured Developing Method）是现有的软件开发方法中最成熟，应用最广泛的方法，其主要特点是快速、自然和方便。结构化开发方法由结构化分析方法（SA 法）、结构化设计方法（SD 法）及结构化程序设计方法（SP 法）构成的。

#### 1. SA 法概述

结构化分析（Structured Analysis, SA）方法是面向数据流的需求分析方法，是 20 世纪 70 年代末由 Yourdon, Constantine 及 DeMarco 等人提出和发展，并得到广泛的应用。它适合于分析大型的数据处理系统，特别是企事业单位的信息管理系统。

SA 法也是一种建模的活动，主要是根据软件内部的数据传递、变换关系，自顶向下逐层分解，描绘出满足功能要求的软件模型。

结构化分析方法的基本思想是“分解”和“抽象”。

### (1) 分解

是指对于一个复杂的系统，为了将复杂性降低到可以掌握的程度，可以把大问题分解成若干小问题，然后分而治之。

图 2.7 是自顶向下逐层分解的示意图。顶层抽象地描述了整个系统，说明系统的边界,即系统的输入和输出数据流。底层具体地描述了系统的每一个细节，由一些不能再分解的基本加工组成，而中间层是从抽象到具体的逐层过渡。

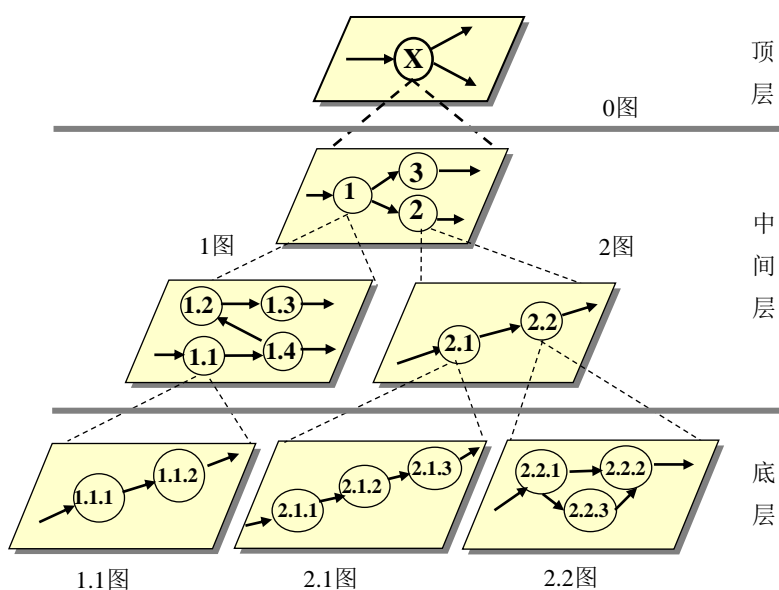


图 2.7 分层 DFD 图

### (2) 抽象

分解可以分层进行，即先考虑问题最本质的属性，暂把细节略去，以后再逐渐添加细节，直至涉及到最详细的内容。这种用最本质的属性表示一个系统的方法就是抽象。

## 2. SA 法的步骤

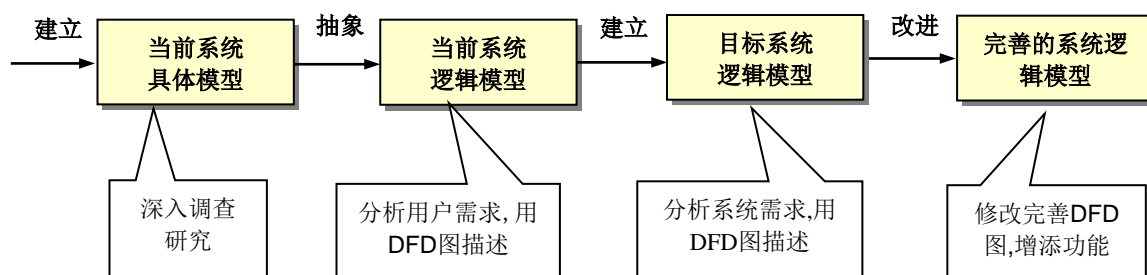


图 2.8 SA 法的步骤

如图 2.8 所示，结构化分析方法，分为以下 4 个主要步骤。

(1) 建立当前系统的“具体模型”。系统的“具体模型”就是现实环境的忠实写照，即将当前系统用 DFD 图描述出来。这样的表达与当前系统完全对应，因此用户容易理解。

(2) 抽象出当前系统的逻辑模型。分析系统的“具体模型”，抽象出其本质的因素，排除次要因素，获得用 DFD 图描述的当前系统的“逻辑模型”。

(3) 建立目标系统的逻辑模型。分析目标系统与当前系统逻辑上的差异，从而进一步明确目标系统是“做什么”的，建立目标系统的逻辑模型，即修改、完善后的 DFD 图。

(4) 为了对目标系统作完整的描述，还需要考虑人机界面和其它一些问题。

### 3. SA 法建模

SA 法建立的分析模型以分层的数据流图为主，再辅以数据词典和加工的逻辑说明，对构成数据流图的主要成分作进一步描述。

#### (1) 数据流图

数据流图 (Data Flow Diagram, DFD) 是描述系统中数据流程的图形工具，它标识了一个系统的逻辑输入和逻辑输出，以及把逻辑输入转换逻辑输出所需的加工处理。

在图 2.9 中给出了构成数据流图的 4 种基本图形符号：



图 2.9 DFD 图的基本符号

箭头表示数据流，圆或椭圆表示加工，双杠或者单杠表示数据存储，矩形框表示数据的源点或终点，即外部实体。

① **数据流**。是数据在系统内传播的路径，由一组成固定的数据项组成。除了数据存储（文件）的输入、输出数据流可以不命名外，其余数据流都应该用名词或名词短语命名。数据流可以从加工流向加工，也可以从加工流向文件或从文件流向加工，还可以从源点流向加工或从加工流向终点。

② **加工**。也称为数据处理，它对数据流进行某些操作或变换。每个加工也要有名字，通常是动词短语，简明地描述完成什么加工。在分层的数据流图中，加工还应有编号。

③ **数据存储**。指暂时保存的数据，它可以是数据库文件或任何形式的数据组织。流向数据存储的数据流可理解为写入文件，或查询文件，从数据存储流出的数据可理解为从文件读数据或得到查询结果。

④ **数据源点和终点**。是软件系统外部环境中的实体（包括人员、组织或其他软件系统），统称为外部实体。一般只出现在数据流图的顶层图中。

【例 2-1】现有一图书预定系统，接收由顾客发来的订单，并对订单进行验证，验证过程是根据图书目录检查订单的正确性，同时根据顾客档案确定是新顾客还是老顾客，是否有信誉。经过验证的正确订单，暂存在待处理的订单文件中。对订单进行成批处理，根据出版社档案，将订单按照出版社进行分类汇总，并保存订单存根，然后将汇总订单发往各出版社。建立图书预定系统顶层的 DFD 图。画图步骤是：

① 首先确定外部实体（顾客、出版社）及输入、输出数据流（订单、出版社订单）。

② 再分解顶层的加工（验证订单、汇总订单）。

③ 确定所使用的文件（图书目录文件、顾客档案等 5 个文件）。

④ 用数据流将各部分连接起来，形成数据封闭。

图 2.10 描述了图书预定系统顶层的 DFD 图。

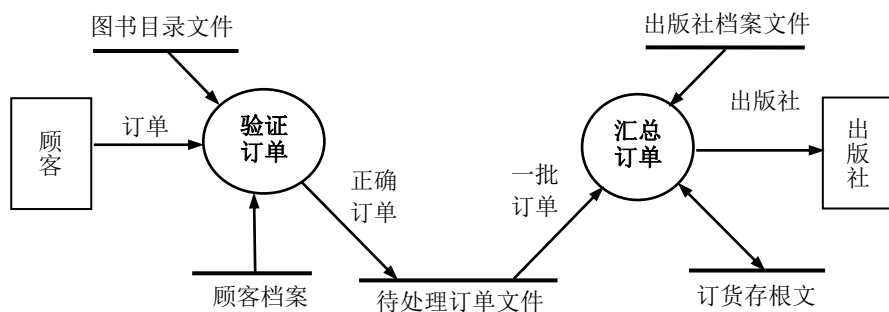


图 2.10 图书预定系统的 DFD 图

特别要注意的是：数据流图不是传统的流程图或框图，数据流也不是控制流。数据流图是从数据的角度来描述一个系统，而框图则是从对数据进行加工的工作人员的角度来描述系统。数据流图中的箭头是数据流，而框图中的箭头则是控制流，控制流表达的是程序执行的次序。

## (2) 建立分层的 DFD 图

当系统规模较大时，仅用一个 DFD 图难以描述，会使得系统变得复杂，且难以理解。为了降低系统的复杂性，采取“至顶而下，逐层分解”的技术，建立分层的 DFD 图。

分层数据流图是一种比较严格又易于理解的描述方式，它的顶层描绘了系统的总貌，底层画出了系统所有的细部，而中间层则给出了从抽象到具体的逐步过渡。

建立分层 DFD 图的一般原则是：“先全局后局部，先整体后细节，先抽象后具体”。同时还应遵循以下具体原则：

### ① 数据守恒与数据封闭原则

所谓数据守恒是指加工的输入输出数据流是否匹配，即每一个加工既有输入数据流又有输出数据流。或者说一个加工至少有一个输入数据流，一个输出数据流。

### ② 加工分解的原则

自然性：概念上合理、清晰；

均匀性：理想的分解是将一个问题分解成大小均匀的几个部分；

分解度：一般每一个加工每次分解最多不要超过 7 个子加工,否则影响易读性和易理解性。应分解到基本加工为止。基本加工是具有完整的功能而又不能再分解的加工。

### ③ 子图与父图“平衡”的原则

父图中某个加工的输入输出数据流应该与相应的子图的输入输出相对应，分层数据流图的特点称为子图与父图“平衡”。

### ④ 合理使用文件

当文件作为某些加工之间的交界面时，文件必须画出来，一旦文件作为数据流图中的一个独立成份画出来了，那么它同其它成份之间的联系也应同时表达出来。

## (3) 分层 DFD 图的改进

理解一个问题总要经过从不正确到正确，从不确切到确切的过程，需求分析的过程总是要不断反复的，一次就成功的可能性是很小的，对复杂的系统尤其如此，因此，系统分析员应随时准备对数据流图进行修改和完善，与用户取得共识，获得无二义性的需求，才能获得更加正确清晰的需求说明，使得设计、编程等阶段能够顺利进行，这样做是必须和值得的。

对 DFD 图的改进的原则与建立分层 DFD 图的基本原则是一致的，可从以下方面考虑对 DFD 图的改进及检查数据流的正确性：

### ① 数据守恒；

- ② 子图、父图的平衡；
- ③ 文件使用是否合理。特别注意输入/出文件的数据流。

改进 DFD 图的易理解性，可以考虑以下方面：

- ① 简化加工之间的联系（加工间的数据流越少，独立性越强，易理解性越好）；
- ② 改进分解的均匀性；
- ③ 适当命名（各成分名称无二义性，准确、具体）。

#### (4) 数据词典

分层数据流图只是表达了系统的“分解”，为了完整地描述这个系统，还需借助“数据词典 (data dictionary)”和“小说明”对图中的每个数据和加工给出解释。

对数据流图中包含的所有元素的定义的集合就构成了数据词典。它有四类条目：数据流、数据项、文件及基本加工。在定义数据流或文件时，使用表 2-1 给出的符号。将这些条目按照一定的规则组织起来，构成数据词典。

表 2-1 在数据词典的定义中出现的符号

符 号	含 义	例及说明
=	被定义为	
+	与	$X=a+b$ ，表示 X 由 a 和 b 组成
[... ...]	或	$X=[a b]$ ，表示 X 由 a 或 b 组成
{...}	重复	$X=\{a\}$ ，表示 X 由 0 个或多个 a 组成
$m\{\dots\}n$ 或 $\{\dots\}_m^n$	重复	$X=2\{a\}6$ ，或 $x=\{a\}_2^6$ ，表示重复 2—6 次 a
(...)	可选	$X=(a)$ ，表示 a 可在 X 中出现，也可不出现
“...”	基本数据元素	$X="a"$ ，表示 X 是取值为字符 a 的数据元素
..	连接符	$X=1..8$ ，表示 X 可取 1 到 8 中的任意一个值

##### ① 数据流条目

给出了 DFD 中数据流的定义，通常对数据流的简单描述为列出该数据流的各组成数据项。

例：数据流“乘客名单”由若干“乘客姓名”、“单位名”和“等级”组成，则词典中的“乘客名单”条目是：

乘客名单={乘客姓名+单位名+等级}

又如：报名单=姓名+单位名+年龄+性别+课程名

也可以对数据流进行较详细的描述，如下例：

例如：某查询系统中，有个名为“查询”的数据流，目前“查询”有三种类型，即“顾客状况查询”、“存货查询”和“发票存根查询”，预计至 1990 年底还将增加 3 至 4 种其他类型的查询。系统每天约需处理 2 000 次查询，每天上午 9：00—10：00 是查询的高峰，此时约有 1 000 次查询。上述信息都是“用户要求”的一部分，在分析阶段应该认真收集，并记录在词典的有关条目中，所以“查询”条目描述如下。

数据流名：查询

简 述：系统处理的一个命令

别 名：无

组 成：[顾客状况查询|存货查询|发票存根查询]

数据量：2000 次/天

峰 值：每天上午 9：00—10：00 有 1000 次

注 释：至 1990 年底还将增加 3 至 4 种查询

## ② 文件条目

给出某个文件的定义，文件的定义通常是列出文件记录的组成数据流，还可指出文件的组织方式。

例如：某销售系统的订单文件：

订单文件=订单编号+顾客名称+产品名称+订货数量+交货日期

## ③ 数据项条目

给出某个数据单项的定义，通常是该数据项的值类型、允许值等。

例如：帐号= 00000 ~ 99999 ； 存款期=[ 1 | 3 | 5 ](单位:年)

## ④ 加工条目

加工条目就是“加工小说明”。由于“加工”是 DFD 图的重要组成部分，一般应单独进行说明。

因此，数据词典是对数据流图所包含的各种元素定义的集合。它对数据流、数据项、文件及基本加工的 4 类条目进行了描述，是对 DFD 图的补充。

## (5) 加工逻辑说明

加工逻辑说明又称为“加工小说明”，对数据流图中每一个不能再分解的基本加工都必须有一个“加工小说明”给出这个加工的精确描述。小说明中应精确地描述加工的激发条件、加工逻辑、优先级、执行频率和出错处理等。加工逻辑是其中最基本的部分，是指用户对这个加工的逻辑要求。

对基本加工说明有三种描述方式：结构化语言，判定表和判定树。在使用时可以根据具体情况，选择其中一种方式对加工进行描述。

### ① 结构化语言

结构化语言是介于自然语言（英语或汉语）和形式语言之间的一种半形式语言，它是自然语言的一个受限制的子集。自然语言容易理解，但容易产生二义性；而形式化语言精确、无二义性，却难理解，不易掌握。结构化语言则综合二者的优点，在自然语言的基础上加上一些约束，一般分为两层结构：外层语法较具体，为控制结构（顺序、选择、循环）；内层较灵活，表达“做什么”。

例如：外层可为顺序结构、选择结构（如 IF-THEN-ELSE，CASE-OF-ENDCASE）、循环结构（如 WHILE-DO; REPEAT-UNTIL）

结构化语言特点：简单，易学，少二义性，但不好处理组合条件。

### ② 判定表

判定表是一种二维的表格，常用于较复杂的组合条件。通常由四部分组成，如表 2-2 所示。对用结构化语言不易处理的较复杂的组合条件问题，可使用判定表。

表 2-2 判定表

条件框	条件条目
操作框	操作条目

其中：条件框表示条件定义；操作框表示操作的定义；条件条目表示各条件的取值及组合；而操作条目表示在各条件取值组合下所执行的操作。表 2-3 是商店每天的营业额所收税率的描述。

表 2-3 税率

营业额X （¥）	$0 \leq X < 5000$	$5000 \leq X < 10000$	$X \geq 10000$
税 率	5%	8%	10%

判定表的特点：可处理较复杂的组合条件，但不易理解，不易输入计算机。

【例 2-2】有一个图书销售系统，其中一个加工为“优惠处理”，条件是：顾客的营业额大于等于 1000 元，同时必须信誉好，或者虽然信誉不好，但是 20 年以上的老主顾。

分析：共有 3 个判定条件，有 8 种可能的组合情况，用 Y 表示满足条件，N 表示不满足条件，X 表示选中判定的结论。其判定表如图 2.11 所示。对图 2.11 进行化简，例如情况 1、2 可以合并，即只要是营业额大于等于 1000 元，同时信誉好，无论是否大于 20 年的老主顾，均可以优惠处理。化简后的判定表为图 2.12。

	1	2	3	4	5	6	7	8
≥1000 元	Y	Y	Y	Y	N	N	N	N
信誉好	Y	Y	N	N	Y	Y	N	N
>20 年	Y	N	Y	N	Y	N	Y	N
优惠处理	X	X	X					
正常处理				X	X	X	X	X

Y-满足条件 N-不满足条件 X-选中判定

图 2.11 判定表

	1	2	3	4
≥1000 元	Y	Y	Y	N
信誉好	Y	N	N	—
>20 年	—	Y	N	—
优惠处理	X	X		
正常处理			X	X

图 2.12 化简后的判定表

### ③ 判定树

与判断表相比，判定树的特点是：描述一般组合条件较清晰，但不易输入计算机。

【例 2-3】仍然以例 2-2 的图书销售系统的处理为例，其判定树如图 2.13 所示。

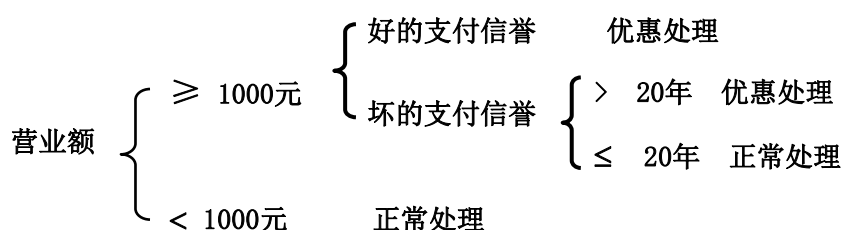


图 2.13 判定树

## 2.4.3 面向对象的分析方法

面向对象的分析方法(Object-Oriented Analysis, OOA)已成为现代软件过程中进行软件分析的主流方法，对解决软件分析中所存在的对问题域的理解，对需求变更的有效管理等问题，都提供了有力的支持。

本节只对面向对象分析方法的重要建模工具——Use Case 模型做简单介绍，并在 2.5.2 节中给出一个应用的实例。

Use Case 的概念是 1992 年由 Jacobson 提出的，并提出可视化的表示方法——Use Case 图，这是他所倡导的面向对象的开发方法(OOSE)的核心。Use Case 的概念一推出就受到了 IT 界的欢迎，被广泛应用到了面向对象的系统分析中。用例驱动的系统分析与设计方法已成为面向对象的系统分析方法的主流，是面向对象方法识别和获取需求的首选工具。

Use Cases 是既能捕获商业处理流程又能捕获系统需求的有效方法，并且它本身比较简单和容易掌握，已经得到越来越广泛的应用。与其它需求捕获技术相比，它成功的原因在于：

① Use Cases 把系统当作一个黑盒，它使得确定系统需求变得更加容易。Use Case 建模是从外部的角度来看系统功能。“外部”即是指“角色”或者“执行者”(Actor)，可以是人也可以是正在设计的



系统与之交互的外部系统。“用例”表示系统的一个子系统，或者表示系统的一个独立的功能。

② Use Case 模型是一种可视化的图形，描述了系统的静态结构。用例图由用例和参与者及用例之间的连接关系构成。如图 2.14 所示，用例的图形为一个椭圆，参与者用一个小人表示，它们之间的关系是由参与者驱动用例，而用例之间的连接关系，常常是<<extend>>或者<<include>>。



图 2.14 用例和角色的表示

总而言之，Use Cases 是非常有效的需求捕获技术，它能为需求提供一个简单方便的软件建模工具。

## 2.5 软件需求案例分析

### 2.5.1 案例 1——医院病房监护系统

【例 2-4】用 SA 法建立“医院病房监护系统”的需求模型，并画出系统的分层 DFD 图。

分析：人们对一个软件系统的交互常常通过对实际的“场景”来进行描述，通过对“场景”的描述，可以容易地了解系统的交互，功能和执行情况。

如图 2.15 所示，各种病症监视器安置在每个病床旁，将病人的组合病症信号（例如由血压、脉搏、体温组成）实时地传送到中央监护系统进行分析处理。在中心值班室里，值班护士使用中央监护系统对病员的情况进行监控，监护系统实时地将病人的病症信号与标准的病症信号进行比较分析，当病症出现异常时，系统会立即自动报警，并实时打印病情报告和更新病历。未报警时，系统将定期自动更新病历，还可根据医生的要求随时打印病人的各类病情报告。

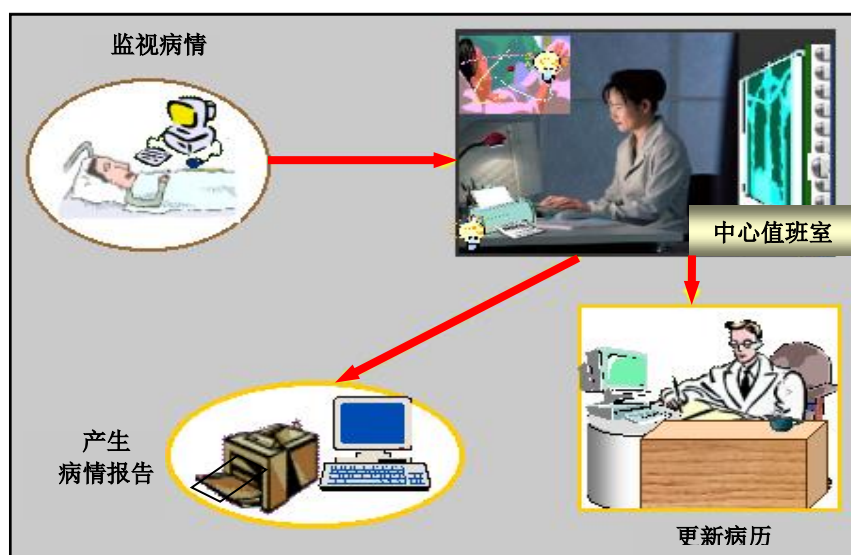


图 2.15 医院病房监护系统的场景

经过初步的需求分析，得到系统的主要功能要求：

- ① 监视病员的病症信号（如：血压、体温、脉搏等）

- ② 定时更新病历
- ③ 病员出现异常情况时报警。
- ④ 打印某一病员的病情报告。

还有一些非功能需求，如：

- ① 监视器与网络的可靠性要求，它涉及病人的生命安全。
- ② 效率需求中对时间、空间的需求，对所采集的大数据量病症信号的存储要求。
- ③ 互操作需求，如要求监视器采样频率可人工调整等。
- ④ 对病人病历的隐私的要求。

由于非功能需求中，主要涉及到硬件，因此这里只考虑功能需求，根据分析得到的系统功能要求，画出医院病房监护系统的分层 DFD 图，首先画顶层的 DFD 图（如图 2.16）。顶层确定了系统的范围，其外部实体为病员和护士。

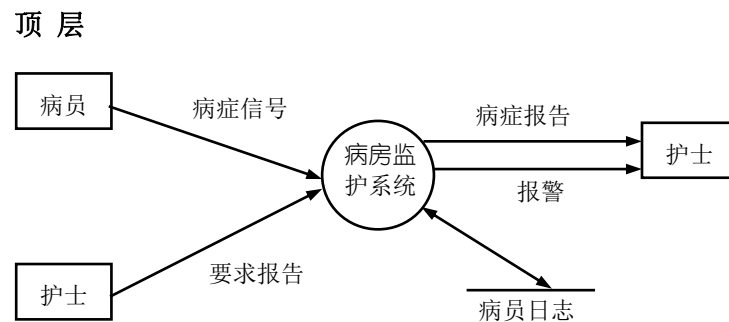


图 2.16 病房监护系统顶层 DFD 图

在顶层 DFD 图的基础上再进行分解，为此，要对系统功能需求做进一步分解：

- (1) 监视病员的病症，即“局部监视”。包含以下功能：

- ① 采集病人的病症信号(血压、体温、脉搏等)。
- ② 将采集到的模拟病症信号转换为数字信号 (A/D 转换)。
- ③ 组合病症信号，以便在网络上传输。

(2) 定时更新病历。病历库管理系统是数字化医院的另一个子系统，必须对病人的病症信号进行处理后，才能写入病历库，称为“更新日志”。包括以下功能：

- ① 将病症信号进行格式化并加入更新日期、时间。
- ② 更新病历库中病人的信息。
- ③ 可人工设定更新病历的时间间隔。

(3) 病情出现异常情况时报警，这是系统的主要功能。由“中央监护”系统完成，首先要开解从“局部监视”传来的组合病症信号，再将开解后的信号与标准病症信号库（专家系统）中的值进行比较，如果超过极限值，立即自动报警。报警过程应完成以下功能：

- ① 根据标准病症信号库中的值，判断是否报警。
- ② 将报警信号转换为各种模拟信号 (D/A 转换)。
- ③ 实时打印病情报告，立即更新病历。

(4) 医生根据病人的病情需要，向值班护士提出生成并打印某一病员的病情报告。可产生如文字、图表、图象等多种类型的病情报告。

通过以上分析，建立第一层的 DFD 图，如图 2.17 所示。

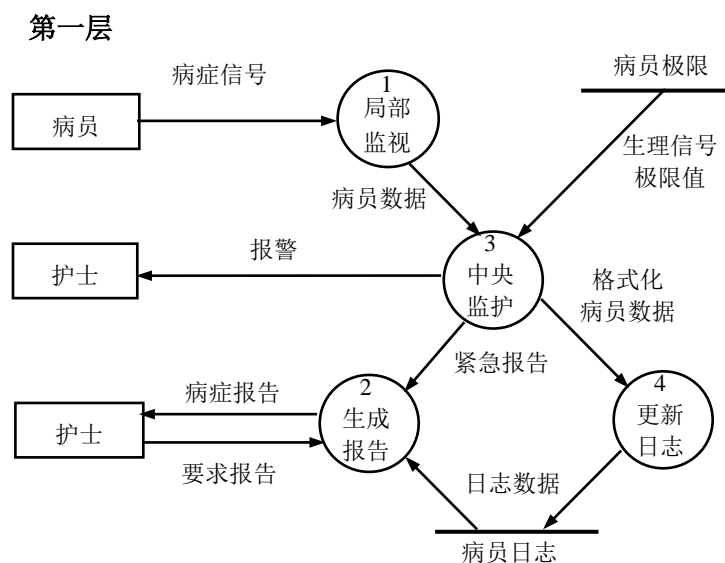


图 2.17 病房监护系统第一层 DFD 图

第一层将系统分解为局部监视、生成报告、中央监视、更新日志 4 个加工。这层的分解是关键，是根据初步的需求分析所得系统主要功能要求来进行分解的。

(1) 加工“局部监视”，是用于监视病员的病症（血压、体温、脉搏等），因此该加工一定有来自病员的输入数据流“病症信号”，输入的病症信号是模拟信号，经过“局部监视”加工后，转换为数字信号，因此该加工应该有数值型的输出数据流“病员数据”。

(2) 加工“中央监视”，显然这是系统中最重要加工。它要接受来自局部监视的病员数据，同时要将病员数据与标准的“病员极限文件”中的“生理信号极限值”进行比较，一旦超过，立即报警。为了定时更新病历，还需要输出“格式化病员数据”。

(3) 定时更新病历的功能就由加工“更新日志”完成，它接收由加工“中央监视”输出的已格式化病员数据，对数据进行整理分类等加工后写入“病员日志文件”（病历），日志数据通常是以压缩的二进制代码存储的。

(4) 为了实现根据医生要求，随机地产生某一病员的病情报告，需要从病员日志文件中提取病员日志数据，由加工“生成报告”进行格式转换，生成并打印输出“病症报告”。

在第一层分解的基础上，应对 4 个加工进行进一步分解，下面以其中最重要的加工“中央监视”为例，进行第二层分解。

如图 2.18 所示，“计算是否超过极限值”显然是中央监视最重要的功能。为了将病员数据与生理信号极限值进行比较，以计算是否超过极限值，首先要将来自加工“局部监视”的病员组合病症信号数据进行“开解信号”。当病员的病症信号产生异常时，要立即报警。由于报警信号有灯光报警、铃声报警及大屏幕上病员的图像显示等多种形式，因此由加工“产生报警信息”来完成。在进行“格式化病员数据”加工时，还要由时钟加入日期和时间。

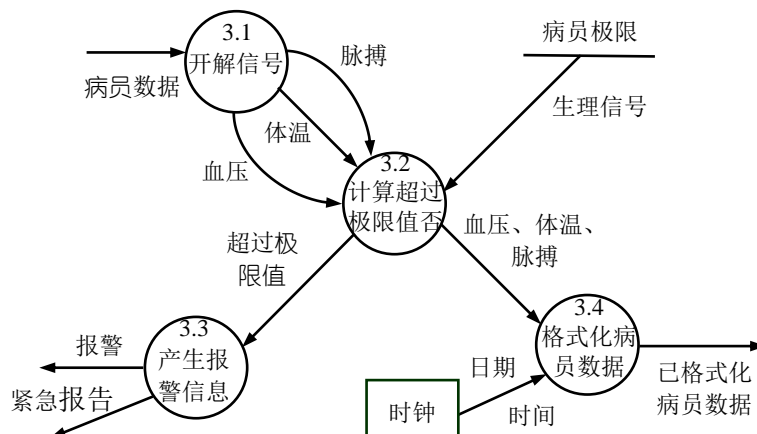


图 2.18 加工“中央监视”分解

### 2.5.2 案例 2——网上竞价拍卖系统

【例 2-5】随着 Internet 技术的发展和互联网的日益普及,互联网用户迅速增加,其中约 1/4 的用户使用 Internet 进行互联网通信或经贸活动。电子商务总额可达到 6 万亿美元。

网上竞价拍卖系统就是一个在互联网上模拟拍卖环境的典型的范例。可实现从展示产品、相互竞价到最后产品成交等一系列功能,用户可以轻松实现在线商品的拍卖和竞标。

采用“基于用例的方法”获取需求,建立系统的 Use Case 模型。

分析:首先确定系统的用户。在本例中,系统是通过网络提供给商品的销售者和购买者一个交易平台,因此所有上网用户都是本系统的用户,也就是执行者。具体又分为商品购买者和商品销售者、系统管理员。考虑到一般用户既可能是商品购买者也可能是商品销售者,所以将用户分为:非会员用户和会员用户。

非会员用户,即未注册的用户,只能在网站上浏览商品,不能参与竞标,也不能提供物品出售。

会员用户,即已注册的用户,可以直接参与拍卖或竞标。

用例表示系统的一个子系统或功能,从外部的执行者(用户)的角度进行分析,该系统应具有以下功能:

- 提供高效的内容丰富的 Web 拍卖商业服务;包括展示产品、相互竞价、产品成交。
- 实现拍卖商品种类的更新和有关商品信息的发布。
- 实现个人物品流通和网上有关拍卖或竞标信息发布、留言。

在以上分析的基础上,进一步确定系统的用例:

- (1) 会员注册。填写用户有关信息,如帐号,用户名,密码,Email 等。
- (2) 会员天地。查看并修改个人信息,交易记录,收邮件,信用评价等。
- (3) 商品分类浏览。选择不同类型的商品进行浏览,还可对商品进行更新、最新商品推荐等。
- (4) 查找商品。按关键字查找、输出打印商品信息。
- (5) 拍卖商品。包括商品上架、商品下架,提供商品信息:商品名称、类别、图片、,起拍价格、新旧程度、使用时间等,及编辑商品等功能。
- (6) 购买商品。即出价参与竞标,拍卖结束时按照竞价规则获得商品。
- (7) 网上支付。通过银行网络系统进行交易,设置多种支付方式。考虑到此功能,应该增加执行者“银行”。
- (8) 会员管理。包括查看会员信息,封锁会员账号,激活会员账号等功能。
- (9) 商品类别管理。添加商品类别,编辑商品类别,删除商品类别。
- (10) 交易管理。主要有查看交易,查看交易报表,关闭交易,退款管理,申诉管理等。

(11) 公告栏管理。添加公告，修改公告，删除公告。  
如图 2.19 所示，根据功能需求，画出网上拍卖系统的用例图。

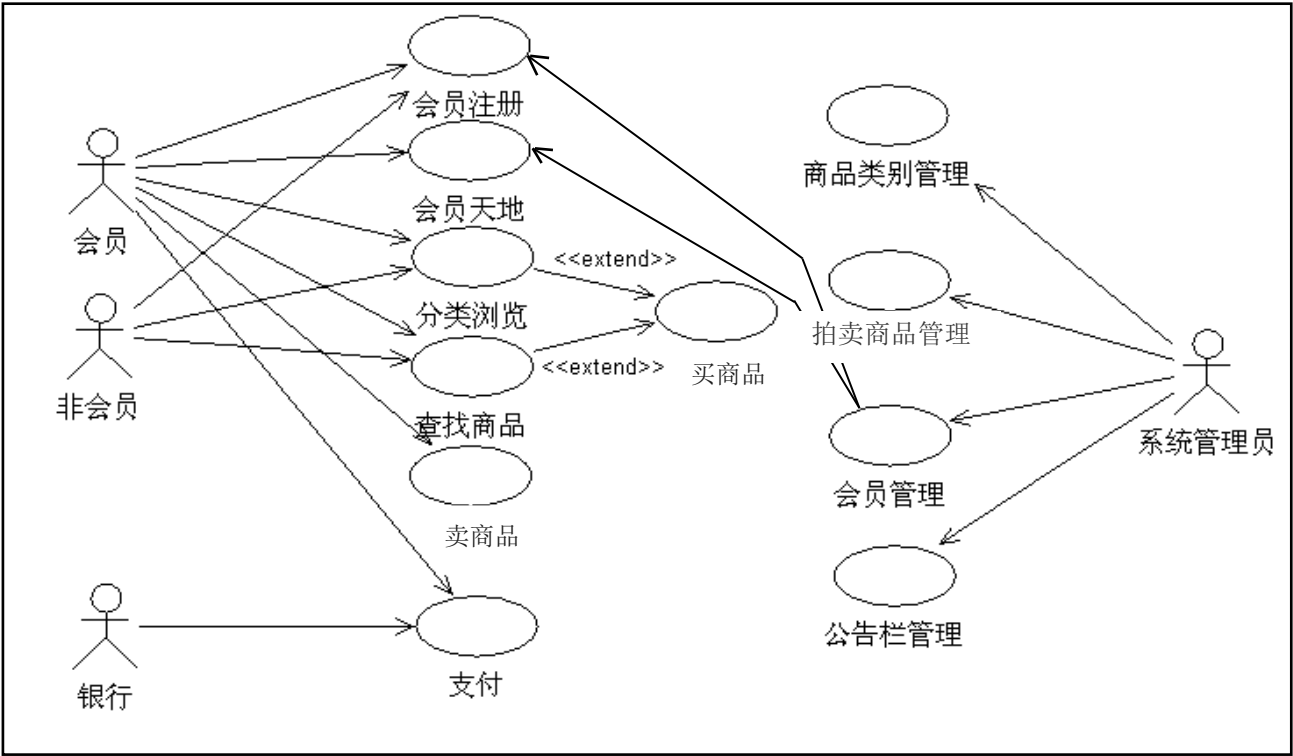


图2.19 网上拍卖系统的用例图

此外，可确定系统的非功能需求如下：

① 时间特性要求

系统采用 JDBC 连接数据库，保证较快的响应时间和更新处理时间，采用 JSP Servlet 技术，以满足用户对数据的转换和传送时间要求。

② 灵活性和精度需求

要求当用户需求如操作方式、运行环境、结果精度、数据结构及其他软件接口等发生变化时，增加新模块时，不会修改原有的模块。

③ 故障处理能力要求

当出现错误时，要求以界面形式向用户说明，并用一览表方式表示各类可能的错误或故障出现时，系统的处理方法和补救措施。

根据上述非功能需求，对初始的用例图（图 2.19）进行修改，针对故障处理能力要求，增加用例“出错处理”。此外，考虑到“分类浏览”和“查找商品”都要从存储商品处获取商品，故增加用例“收藏商品”，包括添加收藏，取消收藏，修改收藏等，图 2.20 为修改后的用例图。

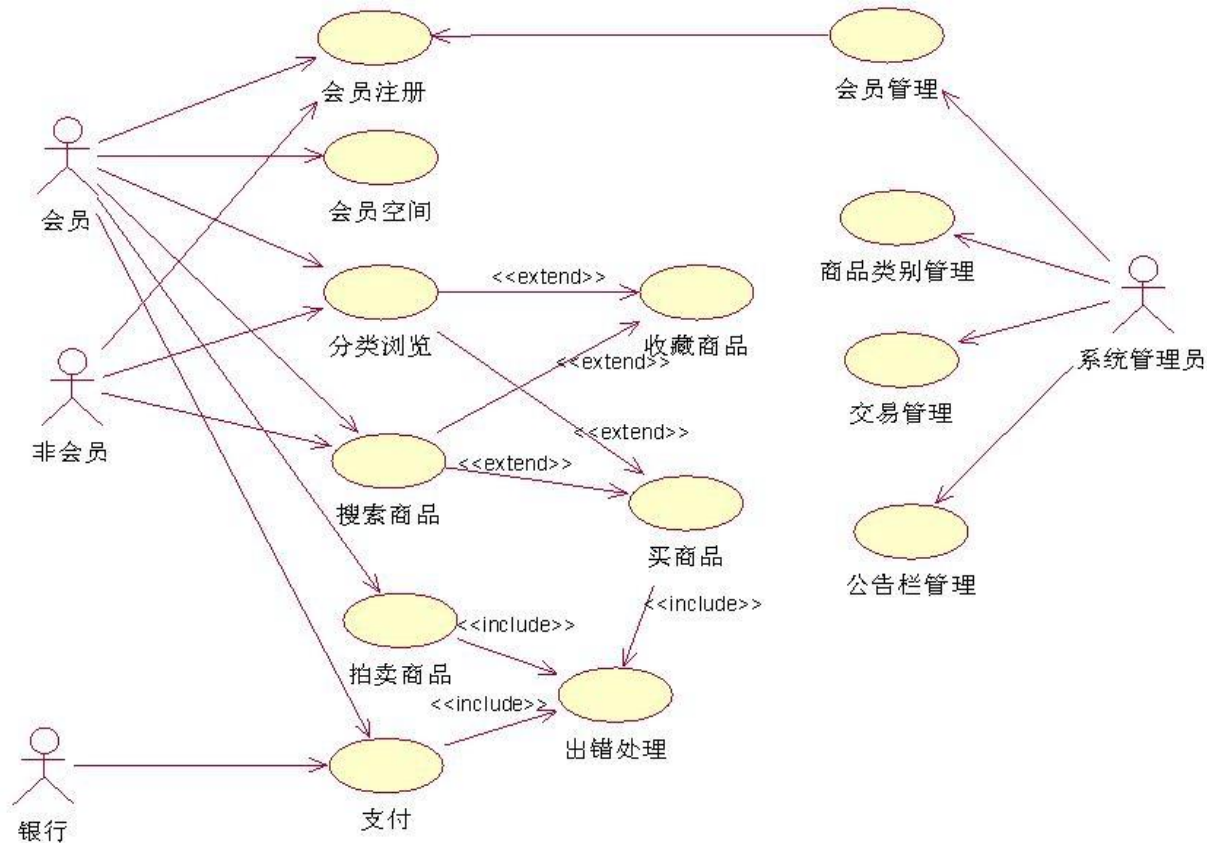


图 2.20 修改后的用例图

软件需求工程,是软件开发人员与用户密切配合,充分交换意见,获得对需求一致意见的过程。

在开发者一方,参与工作的主要人员是系统分析员和系统工程师等,负责沟通用户和开发人员的认识和见解,起着桥梁作用。

需求工程阶段的最终任务是要完成目标系统的需求规格说明,确定系统的功能、非功能需求和性能,为后阶段的开发打下基础。

本阶段常用的有 SA 法,原型法,OOA 法等。

## 小结

需求工程是软件开发的重要阶段,软件需求无疑是当前软件工程中的关键问题,没有需求就没有软件。本章对软件需求的任务、软件的功能需求、非功能需求等基本概念进行了介绍。需求工程的主要活动包括需求的获取、需求分析与建模、需求规格说明、需求确认和需求管理。对其中的关键问题,需求的获取技术和需求分析与建模技术进行了重点讨论,并强调在需求过程中最终用户参与的重要性。此外,在需求阶段,软件需求的变更是不可避免的,也是必须解决的难题。

最后通过 SA 法和 OOA 法的两个案例,对如何进行软件需求作了讨论。

## 习题二

### 一、 选择题

1. 需求工程的主要目的是( )。
  - A) 系统开发的具体方案
  - B) 进一步确定用户的需求
  - C) 解决系统是“做什么的问题”
  - D) 解决系统是“如何做的问题”
2. 需求分析的主要方法有( )。
  - A) 形式化分析方法
  - B) PAD 图描述
  - C) 结构化分析(SA)方法
  - D) OOA 法
3. 面向对象的分析方法主要是建立三类模型, 即( )。
  - A) 系统模型、ER 模型、应用模型
  - B) 对象模型、动态模型、应用模型
  - C) E-R 模型、对象模型、功能模型
  - D) 对象模型、动态模型、功能模型
4. S A法的主要描述手段有( )。
  - A) 系统流程图和模块图
  - B) D F D图、数据词典、加工说明
  - C) 软件结构图、加工说明
  - D) 功能结构图、加工说明
5. 画分层 D F D图的基本原则有( )。
  - A) 数据守恒原则
  - B) 分解的可靠性原则
  - C) 子、父图平衡的原则
  - D) 数据流封闭的原则
6. 在 E-R 模型中, 包含以下基本成分( )。
  - A) 数据、对象、实体
  - B) 控制、联系、对象
  - C) 实体、联系、属性
  - D) 实体、属性、联系
7. 用例驱动的需求方法的主要优点是( )。
  - A) 作为需求分析阶段用户与开发者之间交流信息的工具
  - B) 对系统的数据结构进行描述
  - C) 对目标系统的层次结构进行描述
  - D) 作为分析和设计的工具
8. 数据字典是数据流图中所有元素的定义的集合, 一般由以下四类条目组成( )。
  - A) 数据说明条目、控制流条目、加工条目、数据存储条目
  - B) 数据流条目、数据项条目、文件条目、加工条目
  - C) 数据源条目、数据流条目、数据处理条目、数据文件条目
  - D) 数据流条目、数据文件条目、数据池条目、加工条目
9. 在需求分析阶段主要采用图形工具来描述的原因是( )。
  - A) 图形的信息量大, 便于描述规模大的软件系统
  - B) 图形工具能够极好地概括描述一个系统的信息, 比文字叙述能够更好地表达重要的细节
  - C) 图形能够更加直观地描述目标系统, 便于用户理解和交流, 有利于开发者与用户之间达成一致的需求。
  - D) 图形比文字描述简单、形象。

## 二、判断题



1. 在进行了可行性分析后，需求分析就只需要解决目标系统的设计方案。( )
2. S A法是面向数据流，建立在数据封闭原则上的需求分析方法。( )
3. 需求管理主要是对需求变化的管理，即如何有效控制和适应需求的变化。( )
4. 在面向对象的需求分析方法中，建立动态模型是最主要的任务。( )
5. 加工小说明是对系统流程图中的加工进行说明。( )
6. 判定表的优点是容易转换为计算机实现，缺点是不能够描述组合条件。( )
7. 需求分析的主要方法有 SD 法、OOA 法及 HIPO 法等。( )
8. 分层的 DFD 图可以用于可行性分析阶段，描述系统的物理结构。( )
9. 信息建模方法是从数据的角度来建立信息模型的，最常用的描述信息模型的方法是 E-R 图。( )
10. 用于需求分析的软件工具，应该能够保证需求的正确性，即验证需求的一致性、完整性、现实性和有效性。( )

### 三、问答题

1. 需求工程包括哪些基本活动？各项基本活动的主要任务是什么？
2. 简述抽取需求的主要方法，并比较它们的特点。
3. 客户的需要总是不断(连续)地增长，但是一般的软件开发又总是落后于客户需求的增长，如何管理需求的进化(变化)就成为软件进化的首要问题。请说明需求变更的管理过程。
4. M 公司的软件产品以开发实验型的新软件为主。用瀑布模型进行软件开发已经有近十年了，并取得了一些成功。若你作为一名管理员刚加入 M 公司，你认为快速原型法对公司的软件开发更加优越，请向公司副总裁写一份报告阐明你的理由，切记：副总裁不喜欢报告长度超过一页(B5)。
5. 如何画分层数据流图？有哪些基本原则？
6. 加工小说明有哪些描述方法？它们各有何优缺点？为什么不采用自然语言进行描述？
7. 考察下图子图、父图的平衡。

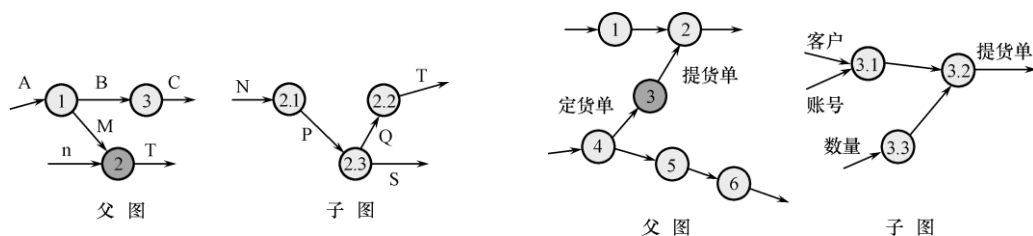


图 1

图 2

8. 画出银行取款过程的 Use Case 图。问题描述为：储户用存折取款，首先填写取款单，根据“账卡”中的信息检验取款单与存折，如有问题，将问题反馈给储户，否则，登录“储户存款 数据库”，修改相应数据，并更新“账卡”，同时发出付款通知，出纳向储户付款。