

软件需求分析

本章要点：

- (1) 需求分析的任务；
- (2) 需求分析的步骤；
- (3) 需求分析的各种描述工具；
- (4) 需求分析的文档。

学完本章之后，你将能够：

- (1) 理解需求分析的任务；
- (2) 掌握需求分析的步骤；
- (3) 理解结构化分析方法的基本思想；
- (4) 掌握数据流图、实体—关系图的绘制方法和数据字典的用法；
- (5) 掌握软件需求规格说明书的内容。

3.1 需求分析的任务

需求分析是研究用户要求，得到目标系统的需求定义的过程。

需求分析的基本任务是软件人员和用户一起完全弄清用户对系统的确切要求。需求分析的结果是否正确，关系到软件开发的成败，正确的需求分析是整个系统开发的基础。需求分析是理解、分析和表达“系统必须做什么”的过程。

其中，理解就是尽可能准确地了解用户当前的情况和需要解决的问题。需求分析阶段并不马上进行具体的系统设计和需求实现，而是对用户提出的要求反复多次地细化，从而充分理解用户的需求。通过分析得出用户对系统完整、准确、清晰、具体的要求。表达是通过建模、规格说明和复审，说明“系统必须做什么”的过程。

建立模型就是描述用户需求，可使用的工具有实体—关系图、数据流图、数据字典、层次图、IPO 图等。

下面介绍需求分析阶段的具体任务。

1. 确定目标系统的具体要求

需求分析阶段要确定目标系统的具体要求。

(1) 系统的运行环境要求

系统运行时的硬件环境要求,如外存储器种类、数据输入方式、数据通信接口等;软件环境要求,如操作系统、汉字系统、数据库管理系统等。

(2) 系统的性能要求

系统的性能要求,如系统所需的存储容量、安全性、可靠性、期望的响应时间(即从终端输入数据到系统后,系统在多长时间内可以有反应,这对于实时系统来讲是关系到系统能否被用户接受的问题)。

(3) 系统功能

确定目标系统必须具备的所有功能,系统功能的限制条件和设计约束。

(4) 接口需求

接口需求描述系统与其环境通信的格式。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、通信接口需求等。

2. 建立目标系统的逻辑模型

需求分析实际上就是建立系统模型的活动。

模型是为了理解事物而对事物做出一种抽象、无歧义的书面描述。模型由一组图形符号和组成图形的规则组成。建模的基本目标如下。

(1) 描述用户需求;

(2) 为软件的设计奠定基础;

(3) 定义一组需求,用于验收软件产品。

模型分为数据模型、功能模型和行为模型。为了理解和表示问题域的信息建立数据模型;为了定义软件的功能建立功能模型;为了表示软件的行为建立行为模型。可用层次的方式来细分数据模型、功能模型和行为模型,在分析过程中得出软件实现的具体细节。

3.2 需求分析的步骤

遵循科学的需求分析步骤可以使需求分析工作更高效。需求分析的步骤如下:获取需求,识别问题;分析和描述系统的逻辑模型;需求分析的复审。

1. 获取需求,识别问题

开发人员从功能、性能、界面和运行环境等多个方面识别目标系统要解决哪些问题,要满足哪些限制条件,这个过程就是对需求的获取。开发人员通过调查研究,要理解当前系统的工作模型和用户对新系统的设想与要求。

此外,在获取需求时,还要明确用户对系统的安全性、可移植性和容错能力等其他要求。比如,多长时间需要对系统做一次备份,系统对运行的操作系统平台有何要求,发生错误后重启系统允许的最长时间是多少等。

获取需求是需求分析的基础。为了能有效地获取需求,开发人员应该采取科学的需求获取方法。在实践中,获取需求的方法有很多种,比如问卷调查、访谈、实地操作和研究资料等。

问卷调查法是采用调查问卷的形式来进行需求分析的一种方法。通过对用户填写的调查问卷进行汇总、统计和分析,开发人员便可以得到一些有用的信息。采用这种方法时,调查问卷的设计很重要。一般在设计调查问卷时,要合理地控制开放式问题和封闭式问题的比例。

(1) 开放式问题的回答不受限制,自由灵活,能够激发用户的思维,使他们能尽可能地阐述自己的真实想法。但是,对开放式问题进行汇总和分析的工作会比较复杂。

(2) 封闭式问题的答案是预先设定的,用户从若干答案中进行选择。封闭式问题便于对问卷信息进行归纳与整理,但是会限制用户的思维。

访谈通过开发人员与特定的用户代表进行座谈,进而了解到用户的意见,是最直接的需求获取方法。为了使访谈有效,在进行访谈之前,开发人员要首先确定访谈的目的,进而准备一个问题列表,预先准备好希望通过访谈解决的问题。在访谈的过程中,开发人员要注意态度诚恳,并保持虚心求教的姿态,同时还要对重点问题进行深入的讨论。由于被访谈的用户身份可能多种多样,开发人员要根据用户的身份特点进行提问,并给予启发。当然,进行详细的记录也是访谈过程中必不可少的工作。访谈完成后,开发人员要对访谈的收获进行总结,澄清已解决的和有待进一步解决的问题。

2. 分析和描述系统的逻辑模型

(1) 建立目标系统的逻辑模型

要把来自用户的信息加以分析,通过“抽象”建立目标系统的逻辑模型。如信息处理系统,通常都是把输入数据转变为需要的输出信息,数据决定了所需要的处理和算法。显然,数据是分析的出发点。

为了达到上述目标,可以用3种不同的图形和数据字典进行描述。数据字典用来描述软件使用或产生的所有实体。数据模型用实体—关系图来描述实体之间的关系。功能模型用数据流图来描述,其作用如下:①数据在系统中移动时如何变换;②描绘变换数据流动的功能和子功能。

行为模型可用状态转换图来描绘系统的各种行为模式(状态)和不同状态间的转换。

(2) 沿数据流图回溯

数据流图画好以后,可以通过数据流图回溯的方法进行审查。分析输出数据是由哪些元素组成,每个输出数据元素又是从哪里来的,沿数据流图的输出端向输入端回溯,此时有关的算法也就初步定义了。在沿数据流图回溯时,有的数据元素可能在数据流图中还没有描述或具体算法还没有确定,需要进一步向用户请教或进一步研究算法。通常把在分析过程中得到的数据元素的信息记录在数据字典中,把补充的数据流、数据存储、数据处理添加到数据流图的适当位置上。

3. 需求分析的复审

由系统分析员和用户一起对需求分析结果进行严格的审查,确保软件需求的一致性、完整性和正确性。

审查内容有:实体—关系图、详细的数据流图、数据字典、状态转换图和一些简明的算法描述等。数据是否准确,是否完整,有没有遗漏必要的处理或数据元素,数据元素从

何来,如何处理等,这一切都必须有确切的回答。而这些答案只能来自于系统用户,因而必须请用户对需求分析做仔细的复查。

用户的复查是从数据流图的输入端开始,分析员借助于数据流图和数据字典及简明的算法描述,向用户解释如何将输入的数据一步一步转变为输出数据。用户应该注意倾听分析员的详细介绍,及时进行纠正和补充。在此过程中很可能引出新的问题,此时应及时修正和补充实体—关系图、详细的数据流图、数据字典、状态转换图和一些简明的算法描述。然后再由用户对修改后的系统做复查,如此反复多次地循环进行,才能得到完整、准确的需求分析结果,才能确保整个系统的可靠性和正确性。

需求分析阶段结束时提供的文档包括以下内容:修正后的项目开发计划、软件需求规格说明书、实体—关系图、详细的数据流图、数据字典、状态转换图和一些简明的算法描述、数据要求说明书、初步的测试计划、用户手册和数据要求说明书等。

3.3 结构化分析方法

结构化分析方法(Structured Analysis,SA)是在20世纪70年代末提出的,多年来被广泛应用。

3.3.1 结构化分析方法概述

结构化分析方法是一种面向数据流的需求分析方法,适用于分析大型的数据处理系统。由于利用图形来表达需求,使得文档清晰、简明,易于学习和掌握,是一种传统的、软件分析人员仍在广泛使用的分析方法。

结构化分析方法的基本思想是按照功能分解的原则,自顶向下逐层分解,逐步求精,直到找到满足功能要求的所有可实现的软件元素为止。

结构化分析方法使用的主要工具有数据流图、数据字典、IPO图等。其中,数据流图能够准确、清晰地描述数据的流向,数据字典可以明确地定义数据的组成和数据项的含义,IPO图可以描述对数据的处理。

3.3.2 数据流图

数据流图是一种图形化技术,它描绘信息流和数据从输入移动到输出的过程中所经历的变换。在数据流图中没有任何具体的物理元素,它只是描绘信息在系统中流动和被处理的情况。因为数据流图是系统逻辑功能的图形表示,所以是软件分析人员和用户之间极好的沟通工具。

设计数据流图时,只需考虑软件系统必须完成的基本逻辑功能,完全无须考虑如何具体地实现这些功能。

如图3-1所示,数据流图有4种基本符号。

(1) 正方形或立方体表示数据的源点或终点。

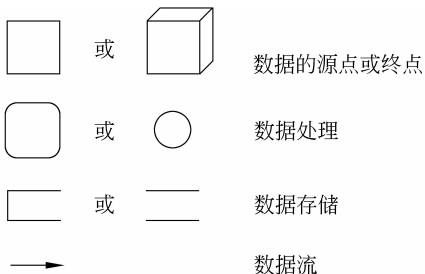


图 3-1 数据流图基本符号

- (2) 圆角矩形或圆形代表数据处理。
- (3) 开口矩形或两条平行横线代表数据存储。
- (4) 箭头表示数据流,即数据流动的方向。

画数据流图的目的是让用户明确系统中数据流动和处理的情况,即系统的基本逻辑功能。对于一个大型系统来说,数据流图的表示方法不是唯一的。较好的方法是分层次地描述系统。顶层数据流图描述系统总体概貌,表明系统关键功能,然后分别把每个关键功能适当地详细描述。这样分层次描述,便于用户逐步深入地了解一个复杂的系统。下面介绍画数据流图的步骤。

(1) 画顶层数据流图

列出系统的全部数据源点和数据终点,将系统加工处理过程作为一个整体,就可得顶层数据流图。

(2) 画各层的数据流图

对系统处理过程自顶向下,逐步分解,每层画出数据流图。

(3) 画总的数据流图

这一步对了解整个系统很有好处,但也要根据实际情况来决定总图的布局,不要把数据流图画得太复杂。

为了对数据流图有更直观的认识,以一个仓库系统为例介绍数据流图的绘制步骤。

某工厂有一材料仓库。仓库保管员每天接收入库单,将材料入库;接收车间出库单,将材料出库,仓库设有库存清单,记录材料编码、名称、规格、型号、平均单价、库存量和库存临界量等信息。保管员每天更新一次库存清单,当材料的库存量等于或少于临界值时向采购部门报告采购单。仓库系统的数据流图如图 3-2 所示。

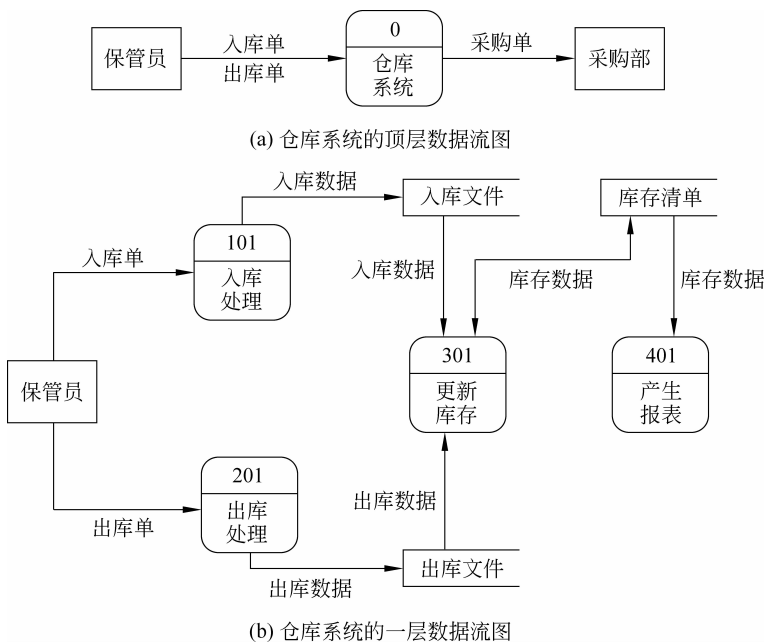


图 3-2 仓库系统的数据流图

在绘制数据流图的过程中,要注意以下几点。

- (1) 数据的处理不一定是一个程序或一个模块,也可以是一个连贯的处理过程。
- (2) 数据存储是指输入或输出文件,但它不仅可以是文件,还可以是数据项或用来组织数据的中间数据。
- (3) 数据流和数据存储是不同状态的数据。数据流是流动状态的数据,而数据存储是指处于静止状态的数据。

(4) 当目标系统的规模较大时,为了描述的清晰和易于理解,通常采用逐层分解的方法,画出分层的数据流图。在分解时,要考虑到自然性、均匀性和分解度几个概念:①自然性是指概念上要合理和清晰;②均匀性是指尽量将一个大问题分解为规模均匀的若干部分;③分解度是指分解的维度,一般每一个加工每次分解最多不宜超过7个子加工,应分解到基本的加工为止。

(5) 数据流图分层细化时必须保持信息的连续性,即细化前后对应功能的输入和输出数据必须相同。

3.3.3 数据字典

数据字典是所有与系统相关的数据元素的有组织的列表,并且包含了对这些数据元素的精确、严格的定义,从而使得用户和系统分析员双方对输入/输出、存储的成分甚至中间计算结果有共同的理解。简而言之,数据字典是描述数据信息的集合,是对系统中使用的所有数据元素的定义的集合。数据字典的作用是在软件分析和设计过程中提供数据描述,是数据流图必不可少的辅助资料。只有数据流图和对数据流图中每个元素的确切定义合起来才构成完整的系统规格说明。

一般来说,数据字典包含以下信息。

- (1) 名字——数据、控制项、数据存储或外部实体的名称。
- (2) 别名——第一项中对象的其他名字。
- (3) 使用地点与方式——使用数据或控制项的处理的列表,以及使用这些对象的方式(例如作为处理的输入、从处理输出、作为数据存储、作为外部实体)。
- (4) 内容描述——描述数据或控制项内容的符号。
- (5) 补充信息——关于数据类型、预置值、限制等的其他信息。

为了规范数据字典的说明内容和格式,在结构化方法中给出了一些标准符号,见表3-1。

表 3-1 数据字典中使用的标准符号

符号	说 明	例 子
=	等价于或定义为	标识符=字母字符+字母数字串
+	和,连接两个数据元素	
[]	或,从方括号内选择一个	教师职称=[助教 讲师 副教授 教授]
{ }	对{ }中的内容可重复使用,重复次数的下限和上限写在括号左右	学生成绩=学号+姓名+1{课程号+成绩}3,表示最多有3门、最少有一门课程的考试成绩
()	可选	曾用名=(姓名),表示曾用名可有可无

例如,学生信息的数据字典可以定义为:

名字: 学生信息
别名: 无
描述: 记录学生的基本信息
定义: 学生信息=学号+姓名+性别+出生日期+籍贯
位置: 数据库的学生信息表

3.3.4 实体—关系图

为了理解和表示问题域的信息,需要建立数据模型。数据模型包含3种相互关联的信息:数据对象、描述数据对象的属性及数据对象彼此间相互连接的关系。

1. 数据对象

数据对象也称为数据实体,是必须被软件理解的复合信息的表示。复合信息是指具有一系列不同性质或属性的事物,因此,只有单个值的事物不是数据对象。复合信息独立描述了外界实体,软件用它来代表客观的实体。

例如,“序号”不是一个合适的数据对象,但是由“序号”、“姓名”、“性别”、“年龄”等信息描述的一个“人”则可以被定义成一个数据对象。

数据对象可以是外部实体(如产生或使用信息的任何事物)、事物(如报告或屏幕显示)、事件(如响警报)、角色(如售货员)、单位(如人事部门、销售中心)、地点(如实验室、仓库)或结构(如文件、目录)等。

教师、学生、课程、教室等都可以认为是数据对象,因为它们都可以由一组属性来定义。

数据对象之间是有关联的。例如,教师和学生的关联是通过课程建立的,教师“教”某门课程,学生“学”某门课程,教或学的关系表示教师和课程或学生和课程之间的一种特定的连接。

又如,在售票系统中,客户、订票单、票单都可以认为是数据对象,客户和订票单之间存在关联,即客户要订票,同时,客户和票单之间也存在关联,因为一张票单唯一地属于一个客户。关系是双向的,即可以从不同的数据对象来看待这种关联。

数据对象只定义了数据,没有定义对数据的操作,这是数据对象与面向对象方法中的类或对象的显著区别。

2. 属性

属性就是数据对象的特征。数据对象常有一组特征,这些特征按照一定的规则组织起来,每组特征的取值表示数据对象的一个实例。例如,在售票系统中,客户是一个数据对象,可用一组数据来描述客户的特征,这组数据包括客户名称、性别、出生日期、送票地址、邮编、联系电话等。每一个客户都有一组属性值,而每组属性值都代表了客户这个群体中的一员。

属性可以用来标识一个数据对象。经常用主关键字或唯一关键字来标识一个数据对

象实例。例如,用学号来标识一个学生。属性还可以用来建立对另一个数据对象的引用,也就是常说的外部关键字。例如,在学生管理系统中,成绩单这个数据对象的属性中有“学号”这个属性值,它标识了对“学生”这个数据对象的引用。如果要了解“学生”的信息,就需要根据“学号”属性值到“学生”数据对象里找到这个“学号”所对应的学生。当然,在“学生”数据对象里“学号”必须是能够唯一标识一个学生的属性,也就是它必须是唯一关键字,否则就不能确定成绩单到底是哪个学生的。

理论上,一个系统数据对象的一组属性对于本系统来说应该是必要且充分的。也就是说,如果缺一个就不能完整地描述这个数据对象的特征;同时,如果多一个,对于描述这个数据对象的特征又是冗余的。但是在实际系统中往往是做不到的,常为了查询的快捷而允许适当的冗余。

3. 关系

数据对象彼此之间相互连接的方式称为关系或联系。

客观世界中的事物之间往往是有关系的。例如,教师与课程之间存在“教”这种关系,而学生与课程之间则存在“学”这种关系。关系可分为 3 类。

(1) 一对一关系(1 : 1)

数据对象 A 的一次出现只能关系到数据对象 B 的一次出现;反之亦然。一个部门有一个经理,而每个经理只在一个部门任职。

(2) 一对多关系(1 : N)

数据对象 A 的一次出现能关系到数据对象 B 的一次或多次出现,但是,数据对象 B 的一次出现只能关系到数据对象 A 的一次出现。例如,一个客户可有一个或多个订票单,但是一个订票单只能属于一个客户。

(3) 多对多关系(M : N)

数据对象 A 的一次出现能关系到数据对象 B 的一次或多次出现,同样,数据对象 B 的一次出现也能关系到数据对象 A 的一次或多次出现。例如,学生与课程之间的关系是多对多,一个学生可以学多门课程,每门课程也可以有多个学生学。

关系也可能有属性。例如:学生“学”某门课程所取得的成绩,既不是学生的属性,也不是课程的属性。成绩既依赖于某个特定的学生,又依赖于某门特定的课程,所以是学生和课程之间的关系“学”的属性。

4. 实体—关系图

通常,使用实体—关系图来建立数据模型,可以把实体—关系图简称为 E-R 图,相应地,用 E-R 图描绘的数据模型也可以称为 E-R 模型。

E-R 图中包含了实体(即数据对象)、关系和属性 3 种基本成分,其表示方法如下。

(1) 实体:用矩形框表示实体。

(2) 关系:用菱形框表示实体之间的关系。

(3) 属性:用椭圆形或圆角矩形表示实体(或关系)的属性。

图 3-3 是学生管理系统中教师和学生的实体—关系图。分析该系统共有教师、学生

和课程 3 类实体。教师的属性有教工号、姓名、性别、职称、职务。学生的属性有学号、姓名、性别、系、班级。课程的属性有课程号、课程名、学时、学分。“教师”教“课程”，“学生”学“课程”，相互之间建立关系。某“学生”学某“课程”后，得到“成绩”，成绩是关系“学”的属性。课程与学生是多对多的关系($M:N$)。每门课程由一位教师担任，而每位教师可以担任多门课程的教学工作，因而教师与课程是一对多的关系($1:N$)。

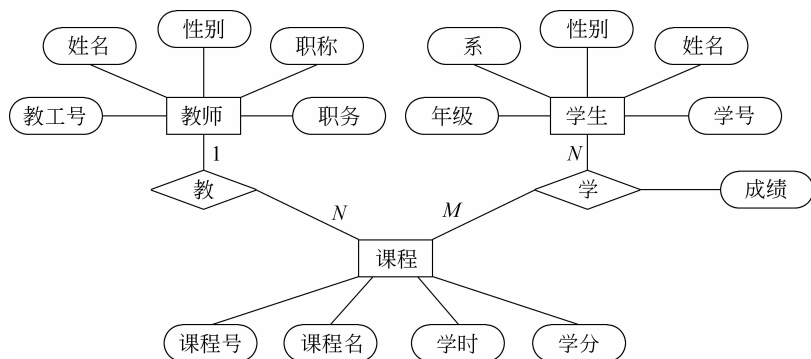


图 3-3 教师与学生的实体—关系图

人们通常就是用实体、属性和关系这 3 个概念来理解现实问题的，因此，E-R 模型比较接近人的思维方式。此外，E-R 模型使用简单的图形符号表达系统分析员对问题域的理解，不熟悉计算机技术的用户也能理解它，因此，E-R 模型可以作为用户与分析员之间有效的交流工具。

3.4 需求分析图形工具

需求分析阶段除了使用实体—关系图、数据流图和数据字典外，还经常利用其他一些图形工具来描述复杂的数据关系和逻辑处理功能。本节介绍层次图和 IPO 图。

1. 层次图

层次图不仅可以反映系统的功能组成关系，也可以反映现实世界中的信息组成关系。如果严格按照结构化方法的步骤进行，功能层次图应该是在概要设计阶段由数据流图转化而来的软件结构图。在需求分析阶段可以利用层次图帮助理解和分析现实世界的数据关系。

层次图是一种树形结构，树上的每个节点都是一个方框。如图 3-4 所示，如果该图作为软件功能结构方框图，则每个方框表示一个功能模块。功能 A 由子功能 B、C、D 实现，其中功能 C 由 E 和 F 子功能实现，而功能 F 又由 G、H、I、J 实现。

如果图 3-4 作为数据结构方框图，则每个方框表示一个数据结构或数据项。树根 A 是个单独的方框，代表一个完整的数据结构，它的组成由下面一层方框 B、C、D 表示，其中 C 由 E 和 F 组成，而 F 由数据项 G、H、I、J 组成。

2. IPO 图

IPO 图是输入/处理/输出图,是由美国 IBM 公司发展完善起来的图形工具。

IPO 图的基本形式是画 3 个方框,左边框中列出有关的输入参数,中间框中列出主要的处理说明,右边框中列出产生的输出参数。处理框中列出的处理次序是按执行顺序书写的,如图 3-5 所示。

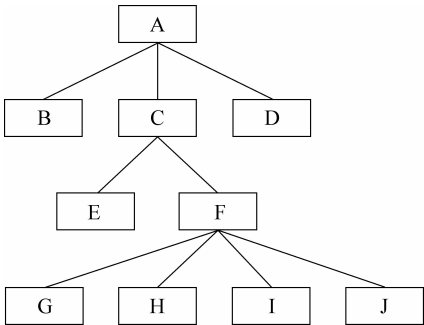


图 3-4 层次图

模块编号：		模块名称：
输入参数	处理说明	输出参数
备注：说明模块的局部变量等		

图 3-5 IPO 图

在实际项目中通常使用 IPO 表,图 3-6 是 IPO 表的形式。

IPO表	
模块编号：_____	作者：_____
系统名称：_____	日期：_____
输入： 	输出：
处理： 	
局部数据元素： 	备注：

图 3-6 IPO 表

3.5 软件需求规格说明书

需求分析阶段除了建立模型之外,还应写出软件需求规格说明书。软件需求规格说明书(Software Requirments Specifications,SRS)精确地阐述一个软件系统提供的功能、性能和必要的限制条件,是系统测试、系统设计和编码的基础。软件需求规格说明书有时附有可执行的原型及初步的用户手册,它是需求分析阶段的最终成果。

一般来说,软件需求规格说明书的格式可以根据项目的具体情况有所变化,没有统一的标准。下面是软件需求规格说明书的编写参考指南。