

第3章

结构化需求分析

需求分析是必要且十分重要的环节。实践证明,软件分析工作的好坏,在很大程度上决定了软件的成败。本章将从实际案例出发,对结构化方法的可行性分析和需求分析过程进行全面的剖析。在软件开发实践中,经过成功和失败,使人们认识到,为了使开发出来的目标系统能满足实际需要,在着手编程之前,首先必须要有一定的时间用来认真考虑以下问题。

- (1) 需求所要求解决的问题是什么?
- (2) 为解决该问题,系统应该做些什么?
- (3) 系统应该怎么做?

软件分析的任务是:通过计划和需求分析,最终完成系统的逻辑方案。逻辑方案不同于物理方案,前者解决“是什么”的问题,是软件分析的任务;后者解决“如何做”的问题,是软件设计的任务。系统分析过程如图 3.1 所示。

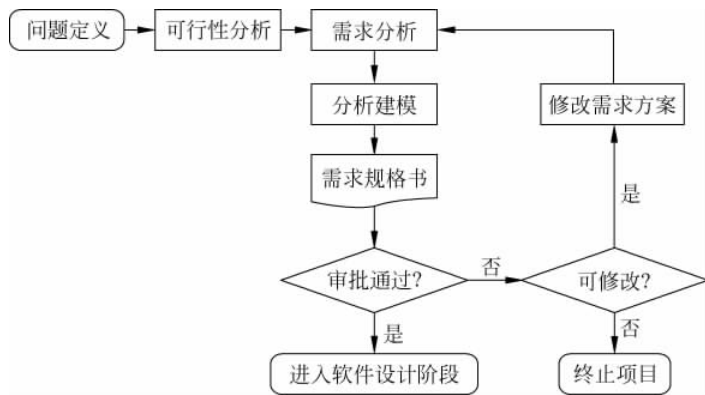


图 3.1 系统分析过程流程图

3.1 可行性研究

3.1.1 问题的定义

问题定义阶段的主要任务是回答“要解决的问题是什么”这一问题。问题定义的内容包括明确问题的背景、开发系统的现状、开发的理由和条件、开发系统的问题要求、总体要求、

问题的性质、类型范围、要实现的目标、功能规模、实现目标的方案、开发的条件、环境要求等,然后写出问题定义报告,以供可行性分析阶段使用。

1. 调研

系统的开发一般都是由用户提出要求开始的。而对于这种开发要求是否具有可行性,以及原有系统是否真是到了必须推倒重来的地步等,都需要在软件开发之前认真考虑。在没有做这些考虑之前提前进入后续任何一项工作都是不明智的。

为了使软件开发工作更加有效地展开,有经验的开发者往往将系统调查分为两步,第一步是初步调查,即先投入少量的人力对系统进行大致的了解,然后再看有无开发的可行性;第二步是详细系统调查,即在软件开发具有可行性并已正式立项后,再投入大量人力展开大规模、全面的系统调查。初步调查的重点如下。

(1) 了解用户与现有系统的总的情况,包括现有系统的规模、系统目标、发展历史、组织结构、人员分工、技术条件、技术水平等。

(2) 现有系统与外部环境的联系,包括现有系统和外部环境有哪些联系,哪些外部条件制约系统的发展等。

(3) 现有系统的现有资源,包括现有系统有哪些资源,系统的状况等。

(4) 用户的需求,包括功能需求,性能需求,资源和环境要求,资金和开发进度等。

通过调查研究,分析员根据软件工作范围,充分理解用户提出的每项功能与要求。同时从软件系统特征、软件开发的全过程,以及可行性分析报告中给出的资源和时间约束来确定软件开发的总策略。只有用户才知道自己需要什么,但是他们并不知道怎样利用软件来实现自己的需求,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,必须通过与用户的沟通获取用户对软件的需求。

对库存管理系统进行初步调查时使用的调查表如表 3.1 和表 3.2 所示。

【案例 3-1】 库存管理系统

案例 3-1“库存管理系统”的调查表如下。

表 3.1 计算机调查表

序号	品牌	型号	数量	配置			软件
				内存	硬盘	显示器	
1	联想	T61	4	2GB	200GB	16 英寸	Windows XP
2	NEC	APC11	3	1GB	100GB	19 英寸	Windows 2007
3	苹果	iPAD2	1	1GB	60GB	14 英寸	Mac

表 3.2 计算机专业人员调查表

	设计与开发人员			实施与管理人员			合计
	高级	中级	初级	高级	中级	初级	
软件							
硬件							

2. 问题定义

在问题定义阶段,分析员要深入现场,阅读用户书写的书面报告,听取用户对开发系统的要求,调查开发系统的背景理由。还要与用户负责人反复讨论,以澄清模糊的地方,改正不正确的地方。最后写出双方都满意的问题定义报告,并确定双方是否可进行深入系统可行性研究的意向。

系统分析过程的第一步涉及对问题的确认。分析员会见客户,客户可能是外面公司、分析员所在公司的市场部门或另一个技术部门的代表,其意图是了解软件开发的目的,并定义满足这一目的所需的具体目标。

问题定义报告的主要内容包括项目名称、背景、项目目标、项目范围、开发条件、环境要求和初步设想等。一旦确定了全部目标,分析员转向可行性评估:建造系统的技术是否存在?将需要什么特殊的开发和制造资源?对成本和进度有什么限制?通过问题的回答以确定目标系统的可执行性。本书以案例 3-1“库存管理系统”为例介绍和描述问题的定义。

案例 3-1“库存管理系统”的问题定义如下。

××公司是小型生产企业,随着改革的深入和经济的发展,该公司的生产任务日益繁重,从而对库存管理的要求也更加严格。在传统的手工管理时期,一种物品由进货到发货,要经过若干环节,且由于物品的规格型号繁多,加之业务人员素质较低等因素,造成物品供应效率低下,严重影响了企业的正常生产。同时,由于库房与管理部门之间的信息交流困难,造成库存严重积压,极大地影响了企业的资金周转速度,另外也使得物资管理、数据汇总成为一大难题。

当今企业的竞争压力越来越大,企业要想生存,就必须在各个方面加强管理,并要求企业有更高的信息化集成,能够对企业的整体资源进行集成管理。现代企业都意识到,企业的竞争是综合实力的竞争,要求企业有更强的资金实力,更快的市场响应速度。这就要求企业各部门之间统一计划,协调生产步骤,汇总信息,调配集团内部资源,实现既要独立,又要统一的资源共享管理。随着信息技术的发展,该厂为了提高库存周转率,加快资金周转速度,决定开发“库存管理系统”。如下所示是库存管理系统的问题定义报告。

(1) 项目: 库存管理系统。

(2) 背景: 由于人工系统业务流程复杂、业务人员素质低,造成工作效率低下;信息交流不畅,造成库存严重积压,极大地影响了企业的资金周转速度;物资管理、数据汇总困难。

(3) 项目目标: 建立一个高效、准确,操作方便,具有查询、更新及统计功能的信息系统,以满足管理人员进行综合的、模糊的查询及更新要求,从而更加方便地管理库存物品。

(4) 项目范围: 硬件利用现有设备,软件开发费用 2 万元。

(5) 开发条件。

系统结构: B/S 结构。

服务器端技术: ASP.NET。

开发语言: C#。

数据库技术: SQL Server 2000。

(6) 环境要求。

服务器端：Windows 2003+IIS 5.1+Visual Studio 2003+SQL Server 2000。

客户端：IE 6.0。

网络：服务器和客户端应有网络连通，配置 TCP/IP。

(7) 初步设想：增加库存查询、库存提示、库存统计等功能。

(8) 可行性研究：建议进行一周，费用 1000 元。

3.1.2 可行性分析的内容

可行性分析又称作可行性研究。可行性分析的任务是以最小的代价、在尽可能短的时间内确定问题是否能够解决，也就是判断原定的目标和规模是否现实。如果在定义阶段较早地识别出一个错误构思的系统，那么，可以避免时间和财物的无谓损失。一般地，在软件项目策划阶段进行的可行性研究应包括经济可行性、技术可行性、法律可行性、方案的选择、可行性分析报告 5 方面的内容，如下所示。

1. 项目背景
 - 1.1 问题描述
 - 1.2 实现环境
 - 1.3 限制条件
2. 管理概要和建议
 - 2.1 重要的研究结果
 - 2.2 说明
 - 2.3 建议
 - 2.4 影响
3. 候选方案
 - 3.1 候选系统的配置
 - 3.2 最终方案的选择标准
4. 系统描述
 - 4.1 系统工作范围的简要说明
 - 4.2 被分配系统元素的可行性
5. 经济可行性(成本-效益分析)
 - 5.1 经费概算
 - 5.2 预期的经济效益
6. 技术可行性(技术风险评价)
 - 6.1 技术实力
 - 6.2 已有工作基础
 - 6.3 设备条件
7. 社会方面的可行性
 - 7.1 系统开发可能导致的侵权、违法和责任
 - 7.2 政策方面的影响

8. 可用性

8.1 用户单位的行政管理、工作制度

8.2 使用人员的素质

9. 其他与项目有关的问题

9.1 其他方案介绍

9.2 未来可能的变化

1. 经济方面的可行性

经济方面的可行性主要是进行成本 / 效益分析。从经济角度判断系统开发是否“合算”。从组织的人力、财力、物力三方面来考察软件开发的可行性。包含在经济可行性研究中的最重要的信息之一是成本/收益分析(对基于计算机的系统项目的经济回报的评估)。成本/收益分析描绘项目开发的成本,并将它们和系统实际的和无形收益相比较。

所谓成本,包括以下几个方面的内容。

- (1) 购置并安装软、硬件及有关设备的费用;
- (2) 系统开发费用;
- (3) 系统安装、运行及维护的费用;
- (4) 人员培训费用。

效益指的是:

- (1) 系统为用户增加的收入或为用户节省的开支,这是有形的效益。
- (2) 给潜在用户心理上造成的影响,这是无形的效益。它可以转化为有形的效益。

成本/收益分析标准随着将被开发的系统的特征、项目的相对规模,以及投资期望回报变化而发生变化。此外,很多收益是无形的,直接的数量比较可能难于实现。

2. 技术方面的可行性

技术可行性分析主要进行技术风险评价。从开发者的技术实力、以往工作基础、问题的复杂性等出发,判断系统开发在时间、费用等限制条件下成功的可能性。分析员需要根据系统的功能、性能需求,建立系统模型。然后对此模型进行一系列的实验、评审和修改。最后由项目管理人员做出是否进行系统开发的决定。如果开发技术风险很大,或者模型演示表明当前采用的技术和方法不能实现系统预期的功能和性能,或者系统的实现不支持各子系统的集成,则项目管理人员可以做出停止系统开发的决定。技术方面可行性的分析包括如下几个方面。

(1) 现有技术的估价:国内外有关技术的发展水平及国家有关技术政策;对目前可利用的技术进行评价(该技术必须是已经广泛应用,有现成产品,而不是待研究或正在研究的)。

(2) 使用现有技术进行软件开发的可行性。

(3) 对技术发展可能产生的影响进行预测。

(4) 对关键技术人员的数量和水平估价。

(5) 计算机硬件的可行性,包括各种外围设备、通信设备、计算机设备的性能是否能满

足软件开发的要求,以及这些设备的使用、维护及其充分发挥效益的可行性。

(6) 计算机软件的可行性,包括各种软件的功能能否满足软件开发的要求,软件系统是否安全可靠,本单位对使用、掌握这些软件技术的可行性。

3. 社会方面的可行性

社会方面的可行性主要是指一些社会因素对系统的影响。例如,新系统开发是否会引起侵权或其他法律责任问题,新系统开发是否符合政府法规或行业正常要求,外部环境的可能变化对新系统的开发影响如何等。

4. 方案的选择

方案的选择评价系统或产品开发的几个可能的候选方案,最后给出结论意见。分析员考虑问题解决的方案,一般采用将一个大而复杂的系统分解为若干个子系统的办法来降低解的复杂性。如何进行系统分解,如何定义各子系统的功能、性能和界面,实现方案可能有多种。可以采用折中的方法,反复比较各个方案的成本/效益,选择可行的方案。

5. 编写可行性研究报告

可行性研究的结果要用可行性研究报告的形式编写出来。

可行性研究的结论应明确指出以下内容。

- (1) 系统具备立即开发的可行性,可进入软件工程的下一个阶段;
- (2) 可行性分析结果完全不可行,软件开发工作必须放弃;
- (3) 某些条件不具备,可以创造条件,增加资源或改变新系统的目标后,再重新进行可行性论证。

可行性研究报告的形式可以有多种,但其中应包含前述可行性分析报告中的主要内容。

可行性研究报告首先由项目负责人审查(审查内容是否可靠),再上报给上级主管审阅(估价项目的地位)。从可行性研究应当得出“行或不行”的决断。

案例 3-1“库存管理系统”的可行性研究报告目录如下所示。

1. 引言

1.1 摘要

1.1.1 系统名称: 库存管理系统

1.1.2 目标: 建立一个高效、准确,操作方便,具有查询、更新及统计功能的信息系统

1.1.3 功能: 系统管理、出入库处理、库存报警、库存查询、库存计划

1.2 背景

1.2.1 系统开发的组织单位: ××软件公司

1.2.2 系统的服务对象: 系统管理员、计划员、采购员、库管员、车间、供应商、其他相关人员

1.2.3 本系统和其他系统或机构的关系和联系: 无

1.3 参考和引用的资料(略)

- 1.4 专门术语和缩写词(略)
 2. 系统开发的必要性和意义(略)
 3. 现有系统调查与分析
 - 3.1 组织机构
 - 3.1.1 供应部:主要负责计划、采购、库存管理和供应
 - 3.1.2 车间:物资消耗部门
 - 3.2 业务流程:(略)
 - 3.3 费用(略)
 - 3.4 计算机应用(略)
 - 3.5 现有系统存在的问题(略)
 4. 新系统的几种方案介绍(略)
 5. 新系统方案比较(略)
 6. 结论
-

3.2 需求分析的任务

通过问题定义和可行性研究,回答了上面的第一个问题,即:需求所要求解决的问题是什么。而第二个问题的解决,即:为解决该问题,软件系统应该做些什么,就是需求分析的任务了。而第三个问题则留给软件设计阶段去解决。

软件需求分析是软件生存周期中重要的一步,是软件定义阶段的最后一个阶段,是关系到软件开发成败的关键步骤。软件需求分析过程就是对可行性研究确定的系统功能进一步具体化,并通过分析员与用户之间的广泛交流,最终形成一个完整、清晰、一致的软件需求规格说明书的过程。通过需求分析能把软件功能和性能的总体概念描述为具体的软件,从而奠定软件开发的基础。总的来说,软件需求分析过程实际上是一个调查研究、分析综合的过程。它准确回答了“系统该做什么”的问题。

软件需求分析的目标是深入描述软件的功能和性能,确定软件设计的约束和软件同其他系统元素的接口细节,定义软件的其他有效性需求。

要解决“系统应该做些什么”的问题,分析员必须与用户密切协商,这是软件分析工作的特点之一。根据现有系统的特点,认真调查和分析用户需求。弄清哪些工作交由计算机完成,哪些工作仍由人工完成,以及计算机可以提供哪些新功能。这样就可以在逻辑上规定目标系统的功能,而不涉及具体的物理实现,也就解决了“系统应做些什么”的问题。

通常软件开发项目是要实现目标系统的物理模型。作为目标系统的参考,需求分析的任务就是借助于现有系统的概念模型导出目标系统的概念模型,解决目标系统“做什么”的问题。其实现步骤如图 3.2 所示。

一个好的设计过程通常总是从现有系统的物理模型出发,导出现有系统的概念模型,经过对现有系统的分析,找出问题和不足之处,设想目标系统的概念模型,最后根据目标系统的概念模型设计新系统的物理模型。下面通过案例 3-1“库存管理系统”需求分析的过程进行相关的讨论。

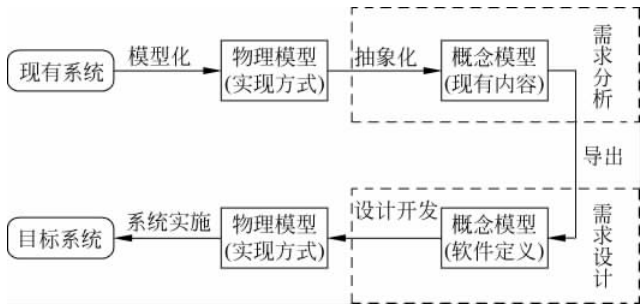


图 3.2 参考现有系统建立目标系统模型

1. 了解现有系统的物理模型

现有的系统是信息的主要来源。显然,如果目前有一个正在使用的系统,那么这个系统一定能完成某些有用的工作,因此,新的目标系统必须也要完成这些基本的工作;另一方面,现有的系统必然存在某些问题或不足,才导致了新系统问题的提出。新的系统必须能够解决现有系统存在的这些问题或不足。此外,运行和使用现有系统所需要的费用是一个重要的经济指标,如果新系统不能增加收入或减少使用费用,那么从经济角度看新系统就不如当前的系统,甚至要考虑新系统开发的必要性。

只要了解现有系统能做什么,而不必清楚是怎样做到的。特别要了解和记录现有系统和其他系统的接口情况,这是设计新系统时的重要约束条件。根据上述了解画出现有系统的物理模型图。

案例 3-1“库存管理系统”现有系统的描述如下。

各车间向物品供应部门提出对某种物品的需求计划,仓库将相应的商品发放给各车间,一般要经过计划、库房管理等过程。系统的物理模型图如图 3.3 所示。

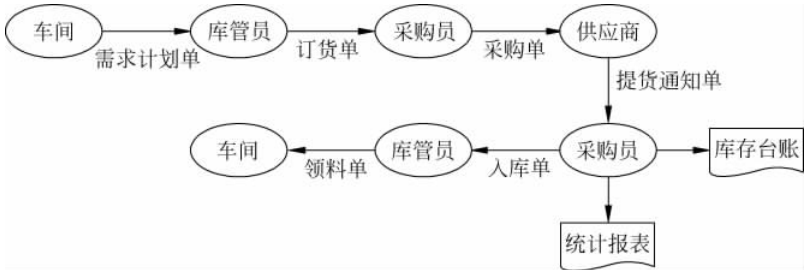


图 3.3 现有系统的物理模型

2. 抽象出现有系统的概念模型

应该仔细阅读和分析现有系统的有关文档和所有资料,当然更要实地考察现有系统。理解现有系统都做了什么,能做什么,为什么这样做,使用现有系统所付出的代价。为了了解上述这些问题,必须访问相关的人员,这就是了解现有系统的过程。

在理解物理模型的基础上,抽取其做什么的本质,从而从现有系统的物理模型抽象出现

有系统的概念模型。该概念模型应能描述出现有系统对相关业务的处理过程。要注意的是千万不要花费太多的经历去分析和描述现有系统的实现细节。

应与用户座谈、沟通,开展调研,了解现有系统存在哪些问题和不足之处。例如,经过对当前运行的“库存管理系统”的调研,发现案例 3-1 存在以下问题。

案例 3-1“库存管理系统”现有系统存在的问题和不足如下。

由于采用的是手工管理,账目繁多,加之几个仓库之间距离较远,库管员、计划员和有关领导相互之间的信息交流困难,使得物资供应效率低下,影响生产。同时每月的月末报表会耗费大量的人力,且由于手工处理容易造成失误,从而影响了数据的效率和准确率,造成了不必要的损失。因此,该厂必须建立相应的库存管理系统,使其能根据市场情况,及时合理地采购所需商品,同时又能科学地对商品进行管理,统筹安排人力、物力、财力,有效地改善当前管理的混乱状况。

根据对该厂的库存管理情况所做的调查和参考有关资料,发现目前该厂在库存管理方面存在着如下问题。

(1) 不能及时获得库存信息。在企业运作过程中,管理人员必须获知各种商品当前的库存量,在库存数量小于商品的最低库存限度的时候,向供应商进行订货;在库存数量大于商品的最高库存限度的时候,即商品积压的时候,应该停止商品的进货活动。但在实际操作中,由于商品的种类多、数量大,需要进行仔细核算,这不仅费时,而且易出错,从而影响企业快速有效地运转。

(2) 库存信息不够准确。仓库管理员根据各种入库单、需求计划单和领料单进行商品的入库、出库操作后,要随时修改商品的库存信息和出库、入库信息,以便反映库存状况。工作中的主要问题是:由于商品种类多、数量大、出库入库操作频繁等原因,造成库存记录 and 实际库存量通常达不到严格一致,因而需要通过盘点来纠正差错,这既耽误时间,又增加了工作量。

(3) 无法及时了解车间对库存商品的需求情况。在需求计划单下达后,由于库存商品与车间的关系复杂,根据送料员的个人经验给各车间分配车间所需商品时,常缺少入库、出库信息和相关信息,经常出现车间缺少该商品的时候才知道该产品需要情况,此时如果库存量不足,将会导致车间的停产。无法及时了解车间对库存商品的需求情况会使企业的生产和销售环节发生混乱,使企业无法正常运作。

(4) 市场需求日益多样化和个性化。产品更新换代的周期越来越短,这就要求企业必须改变库存管理现状,以适应时代的要求。

3. 建立目标系统的概念模型

通过前面对现有系统的了解,分析员就清楚了新系统应该具有哪些基本功能、应该增加哪些功能,及其所受的约束和限制条件。描绘出目标系统的业务处理的过程,从而表达出对目标系统的设想。目标系统的系统流程图实际上是对目标系统的一个高度的抽象模型,对具体功能、性能、数据,以及各种约束和限制条件的描述还有待于在需求分析与设计的过程中通过软件建模来实现。在本书案例 3-1 的“库存管理系统”中,对目标系统新增加的功能的描述如下。也可以通过目标系统的系统流程图进行说明。

4. 对目标系统概念模型进行补充,以求完整地描述

因为传统企业库存管理存在以上的问题,难以适应现代库存管理要求,所以现代企业库存管理系统要具有以下特点。

(1) 科学的库存管理流程。存货的种类不同,所涉及的业务环节及它们所组成的业务流程也各有差异。一般而言,库存业务包括入库处理、货物保管和出库处理三个主要部分。通畅的业务流程是保障高效库存管理的基础,应具备优化、无冗余、并行作业的基本属性。企业库存管理系统对企业的业务流程进行流程再造,使其更加通畅,提高企业在同行业中的竞争力。

(2) 商品代码化管理。代码问题,严格地说是一个科学管理的问题,设计出一个好的代码方案对于系统的开发工作是一件极为有利的东西。代码设计得好可以使很多机器处理变得十分方便,还可以把一些现阶段计算机很难处理的工作变成很简单的工作。

(3) 由于库存商品种类繁多,在库存管理过程中极易发生混乱的问题。IT技术与层次编码技术的结合为商品的高效管理提供了可能。这种编码技术对所有库存商品按照层次和类别赋予唯一的编码。它是区分不同商品的最主要的标准,具有易读和易记的特点,使得管理者只需知道商品的编码,就可以了解该商品的有关信息。

(4) 库存异常报警。当库存数量小于商品的最低库存限度的时候,系统发出警报,提醒管理人员应该向供应商进行订货;在库存数量大于商品的最高库存限度的时候,即商品积压的时候,系统也会发出警报,提醒管理人员应该停止商品的进货活动。也就是说,企业库存管理系统既能防止商品供应滞后于车间对他们的需求,也能防止商品过早地生产和进货,以免增加库存。

软件需求分析阶段研究的对象是软件项目的用户要求,如何准确表达用户的要求,怎样与用户共同明确将要开发的是一个什么样的系统,是需求分析要解决的主要问题。一方面,必须全面理解用户的各项要求,但又不能全盘接受所有的要求;另一方面,要准确地表达被接受的用户要求。只有经过确切描述的软件需求才能成为软件设计的基础。也就是说,需求阶段的任务并不是确定系统怎样完成工作,而仅仅是确定系统必须完成哪些工作,即对目标系统提出完整、准确、清晰、具体的要求。需求分析阶段所要完成的任务是在可行性研究成果的基础上,针对目标模型,通过分析综合建立软件系统的分析模型,编制出软件需求规格说明书。

在需求分析阶段,分析员要对收集到的大量资料和数据进行分析综合,透过现象看本质,看到事物的内在联系及矛盾所在;同时,对于那些非本质的东西,找出解决矛盾的办法;最后,通过“抽象”建立起描述软件需求的一组模型,即分析模型。分析模型应该包含系统以下的需求内容。

- (1) 功能要求;
- (2) 性能要求(如响应时间、吞吐量、处理时间、对内外存的限制等的要求);
- (3) 接口要求;
- (4) 约束与限制条件要求(安全性、保密性、可靠性);
- (5) 运行要求(如对硬件、支撑软件、数据通信接口等的要求);
- (6) 异常处理等对系统的综合需求,以及对于系统信息处理中数据元素的组成、数据的

逻辑关系、数据字典和数据模型等系统的数据要求。

这些是形成软件需求规格说明书、进行软件设计与实现的基础。需求规格说明书是软件分析阶段的最后结果,它通过一组图表和文字说明描述了目标系统的概念模型。设计概念模型是软件分析工作的另一个特点。概念模型包括数据流图、数据字典、基本加工说明等。它们不仅在逻辑上表示目标系统所具备的各种功能,而且表达了输入、输出、数据存储、数据流程和系统环境等。概念模型只告诉人们目标系统要“做什么”,而暂不考虑系统怎样来实现的问题。

对需求规格说明书的复审是由软件开发者和客户一起进行的,因为需求规格说明书构成了设计和以后的软件工程活动的基础,在进行复审时必须给予特别的重视。

简单说来,需求分析阶段是将新系统目标具体化为用户需求,再将用户需求转换为新系统的概念模型,系统的概念模型是用户需求明确、详细的表示。

3.3 获取需求的方法

需求获取是软件开发工作中最重要的一个环节之一。其工作质量对整个软件系统的开发建设的成败具有决定性影响。需求获取工作量大,所涉及的过程、人员、数据、信息非常多,因此要想获得真实、全面的需求必须要有正确的方法。需求获取技术包括以下两方面的工作。

- (1) 建立获取用户需求的方法的框架;
- (2) 支持和监控需求获取的过程的机制。

获取用户需求的主要方法是调查研究。

1. 了解系统的需求

软件开发常常是系统开发的一部分。仔细分析研究系统的需求规格说明,对软件的需求获取是很有必要的。应将用户日常业务中所用的计划、原始凭据、单据和报表等的格式或样本统统收集起来,以便进行分类研究。

2. 市场调查

了解市场和用户对待开发软件有什么样的要求;了解市场上有无与待开发软件类似的系统。如果有,在功能上、性能上、价格上情况如何。也可以采用书面调查的方法,根据系统特点设计调查表,用调查表向有关单位和个人征求意见和收集数据。如果企业已经具有网络设施,可通过 Internet 和局域网发电子邮件进行调查,还可以利用打电话和召开电视会议进行调查,但只能作为补充手段,因为许多资料需要亲自收集和整理。这可以大大节省时间、人力、物力和成本。

在做调查研究时,可以采取如下调查方式。

- (1) 制定调查提纲,向不同层次的用户发调查表,如表 3.3 所示。
- (2) 按用户的不同层次,分别召开调查会,了解用户对待开发系统的想法和建议。
- (3) 向用户领域的专家或在关键岗位上工作的人员个别咨询。
- (4) 实地考察,跟踪现场业务流程。

- (5) 查阅与待开发系统有关的资料。
- (6) 使用各种调查工具，如数据流图、任务分解图、网络图等。

表 3.3 问卷调查表

序号	问 题	回答
1	你的工作岗位是什么	
2	你的工作性质是什么	
3	你的工作任务是什么(收集或绘制业务功能图)	
4	你每天的工作时间安排(绘制工作安排表)	
5	你的工作同前/后续工作如何联系(绘制工作流程)	
6	如何建立计算机系统,你愿意学习操作吗	
7		
8		

×××先生/女士：

您好！请您抽空准备一下，我们将于×日与您会面。谢谢！

××课题组

3. 访问用户和用户领域的专家

开调查会有助于大家的见解互相补充，以便形成较为完整的印象。但是由于时间限制等其他因素，不能完全反映出每个与会者的意见，因此，往往在会后根据具体需要再进行个别访问。把从用户那里得到的信息作为重要的原始资料进行分析；访问用户领域的专家所得到的信息将有助于对用户需求的理解。

4. 考察现场

如果条件允许，亲自参加业务实践是了解现有系统的最好方法。通过实践加深了开发人员和用户的思想交流和友谊，通过了解用户实际的操作环境、操作过程和操作要求，对照用户提交的问题陈述，对用户需求可以有更全面、更细致的认识。这将有利于下一步的系统开发工作。

为了能够有效地获取和理清用户需求，应当打破用户(需方)和开发者(供方)的界限，共同组成一个联合小组，发挥各自的长处，协同工作。

3.4 结构化需求分析方法

3.4.1 需求分析的原则

一般来说，现有系统是杂乱无章的，并且用户群体中的各个用户会从不同角度提出对原始问题的理解及对用计算机要实现管理的软件系统的需求，但并非所有的用户提出的要求都是合理的，所以必须全面理解用户的各项要求，不可能接受所有的要求。所以要经过一个认清问题、分析资料、建立分析模型的过程。

需求分析的方法很多,但是总的来说任何分析方法都是为了把软件系统要做的事情描述清楚,所以有其共同的、可适用的基本原则。

1. 必须能够清楚地表达和理解问题的功能域和数据域

软件的定义以及开发工作基本上都是要解决数据处理问题,即:从一种形式转换为另一种形式。转换过程大致都要经历数据的输入、数据的加工和对结果数据的输出等几个步骤。

对于软件中的数据,所谓的“数据域”应包含数据流、数据内容和数据结构。

数据流指的是数据通过一个系统时的变化方式。对数据进行转换是软件中常见的功能。而功能与功能之间的数据传递就确定了功能之间的接口。

数据内容指的就是数据项。

数据结构指的是各种数据项的逻辑组织方式,如顺序表、各种树状的链表、二维表等。

2. 以层次化的方式对问题进行分解和不断细化

通常软件要处理的问题,作为一个整体来看规模庞大,过于复杂,很难理解。可以把问题以某种方式分解为几个较易理解的部分,并确定各部分之间的接口,从而实现整体的功能。

在需求分析阶段,软件的功能域和信息域都能做进一步的分解。这种分解可以是同一个层次的,称为横向分解;也可以是多层次的纵向分解。

3. 给出系统的逻辑视图和物理视图

需求分析的方法各有不同,但是无论采用什么方法,必须给出系统的概念模型和物理模型,这对于满足处理需求所提出的逻辑限制条件和系统中其他成分提出的物理限制条件是必不可少的。

软件需求的概念模型给出软件要达到的功能和要处理的数据之间的关系,而不是实现的细节。软件需求中的概念模型是软件设计的基础。

软件需求的物理模型给出处理功能和数据结构的实际表现形式,这往往与设备相关,分析员必须清楚各种相关元素对软件的限制,来正确地进行功能和信息结构的物理表示。

3.4.2 需求分析的过程

在软件需求分析中,必须采用合理的步骤,才能准确地获取软件的需求,产生符合要求的软件需求规格说明书。软件需求可以分为以下4个过程:问题识别与需求获取,分析综合与软件建模,编写软件需求规格说明书,评审与需求验证。下面分别加以介绍。

1. 问题识别与需求的获取

首先系统分析人员要确定对目标系统的综合要求,即软件的需求,并提出这些需求实现条件,以及需求应达到的标准。这些需求包括功能需求、性能需求、环境需求、可靠性需求、安全保密需求、用户界面需求、资源使用需求、软件成本消耗与开发进度需求,并预先估计以后系统可能达到的目标。此外,还需要注意其他非功能性的需求。如针对采用某种开发模

式,确定质量控制标准、里程碑和评审、验收标准、各种质量要求的优先级等,以及可维护性方面的需求。

需求获取通常从分析现有系统包含的数据开始。首先分析现实世界,进行现场调查研究,通过与用户的交流,理解现有系统是如何运行的,了解现有系统的机构、输入输出、资源利用情况和日常数据处理过程,并用一个具体模型反映分析员对现有系统的理解。这就是现有系统的物理模型的建立过程。这一模型应客观地反映现实世界的实际情况。

此外,要建立分析所需要的环境,以保证能顺利地对问题进行分析。分析所需的环境如图 3.4 所示。

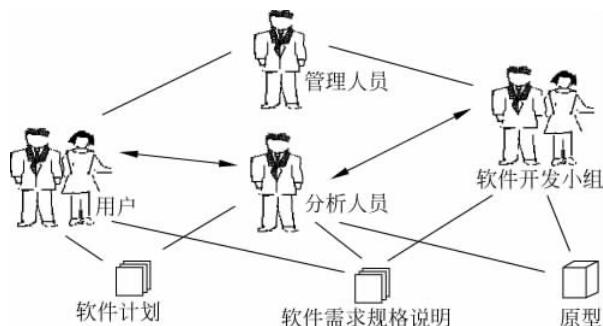


图 3.4 软件需求分析的通信途径

2. 分析综合与软件建模

问题分析和方案的综合是需求分析的第二个方面的工作。分析员必须从信息流和信息结构出发,逐步细化所有的软件功能,找出系统各元素之间的联系、接口特性和设计上的限制,判断是否存在因片面性或短期行为而导致的不合理的用户要求,是否有用户尚未提出的真正有价值的潜在要求。剔除其不合理的部分,增加其需要部分,最终综合成系统的解决方案,给出目标系统的详细概念模型。

分析建模的过程,就是从现有系统的物理模型中,抽象出现有系统的概念模型,再利用现有系统的概念模型,除去那些非本质的东西,抽象出目标系统的概念模型的过程,即对目标系统的综合要求及数据要求的分析综合的过程,是需求分析过程中关键的一步。

首先,在理解现有系统“怎么做”的基础上,抽取其“做什么”的本质,从而从物理模型中抽象出现有系统的概念模型。在物理模型中有许多物理的因素,随着分析工作的深入,需要对物理模型进行分析,区分出本质的和非本质的因素,去掉那些非本质的因素,得出反映系统本质的概念模型。

其次,分析目标系统与现有系统在逻辑上的差别,从现有系统的概念模型导出目标系统的概念模型。从分析现有系统与目标系统变化范围的不同,决定目标系统与现有系统在逻辑上的差别;将变化的部分看作是新的处理步骤,并对数据流进行调整;由外向里对变化的部分进行分析,推断其结构,获取目标系统的概念模型。

最后,补充目标系统的概念模型。为了使已经得出的模型能够对目标系统做完整的描述,还需要从目标系统的人机界面、尚未详细考虑的细节以及其他诸如系统能够满足的性能和限制等方面加以补充。

3. 编写软件需求规格说明书

分析模型建立后,已经确定的目标系统的概念模型应当得到清晰准确的描述。我们把描述目标系统的概念模型的文档称为软件需求规格说明书。软件需求规格说明书是软件需求分析阶段最主要的文档。它是连接计划阶段和开发阶段的桥梁,是软件设计的依据。许多事实表明,在软件需求规格说明书中的任何一个微小错误都有可能导致系统的错误,在纠正时将会付出巨大的代价。同时,为了准确表达用户对软件的输入/输出要求,还需要制定数据要求说明书及编写初步的手册,以及反映目标系统的人机界面和用户使用的具体要求。此外,依据在需求分析阶段对目标系统的进一步分析,可以更准确地估计被开发项目的成本与进度,从而修改、完善并确定软件开发实施计划。

软件需求规格说明书是沟通用户与分析员的媒介,双方要用它来表达对于需要计算机解决的问题的理解。书写时应当易于理解和无二义性,尤其是在描述的过程中最好不使用用户不易理解的专业术语。

为了便于用户,尤其是不熟悉计算机的用户容易理解,软件需求规格说明书应该直观、易读和易于修改,所以应尽量以图文结合的方式,采用自然语言,标准的图形、表格和简单的符号来表示。

4. 评审与需求验证

由分析员提供的软件需求规格说明书的初稿看起来是正确的,但在实现的过程中却会出现各种各样的问题,例如:需求不一致问题、二义性问题等。这些都必须通过需求分析的验证、复审来发现,确保软件需求规格说明书可作为软件设计和最终系统验收的依据。作为需求分析阶段工作的复查手段,应该对功能的正确性、文档的一致性、完备性、准确性和清晰性,以及其他需求给予评价。为保证软件需求定义的质量,评审应以专门指定的人员负责,并按规程严格进行。评审结束应有评审负责人的结论意见及签字。除分析员之外,用户/需求者,开发部门的管理者,软件设计、实现、测试的人员都应当参加评审工作。验证的结果可能会引起修改,必要时要修改软件计划来反映环境的变化。需求验证是软件需求分析任务完成的标志。

3.5 分析建模

为了更好地理解需求获取过程中用户描述的问题,可以采用创建模型的方式来实现。这就是分析建模的过程。所谓模型,就是为了理解事物所做出的一种抽象,是对事物无歧义的书面描述。模型由一组图形符号和组成这些符号的规则所组成。

3.5.1 分析模型

软件的分析模型通常是由一组模型组成,其中包括数据模型、功能模型和行为模型。目前有两种主要的建立分析模型的方法:一种是结构化方法的需求分析模型,这是传统的建模方法,将在本节进行描述。另一种方法是面向对象分析模型,将在后面的章节中详细

介绍。

结构化方法的需求分析模型的组成结构如图 3.5 所示,可以看出模型的核心是 DD (Data Dictionary,数据字典),这是系统所涉及的各种数据对象的总和。



图 3.5 结构化方法的需求分析模型的组成结构

从数据字典出发主要通过以下三种图来构建该模型的三种模型,即:通过 E-R 图进行数据建模,通过 DFD 进行功能建模,通过 STD 进行行为建模。

(1) E-R 图(Entity Relation Diagram,实体联系图):用于描述数据对象间的关系,构建软件的数据模型,在实体-关系中出现的数据对象的属性均可用数据对象进行说明描述。

(2) DFD(Data Flow Diagram,数据流图):其主要作用是指明系统中数据是如何流动和变换的,以及描述数据流是如何进行变换的。在 DFD 中出现的每个功能都会写在 PSPEC(Process Specification,加工说明)中,它们构成系统的功能模型。

(3) STD(Status Transfer Diagram,状态转换图):用于指明系统在外部事件的作用下将如何动作,表明系统的各种状态及各种状态间的变迁。所有软件控制方面的附加信息包含在 CSPEC(Control Specification,控制说明)中,它们构成系统的行为模型。

早期的结构化方法的需求分析模型只包括 DD、DFD 和 PSPEC,主要描述软件的数据模型与功能模型。一方面随着软件开发技术的不断发展,特别是数据库技术的发展,软件系统要去满足用户更多、更复杂的数据信息要求。在数据建模时,人们就将用于数据库设计方面的 E-R 图用于结构化方法的需求分析,用来描述包含较复杂数据对象和信息模型。另一方面,随着计算机实时系统应用的不断拓展,在分析建模过程中,由实时发生的事件来触发控制的数据加工,无法用传统的 DFD 来表示。因此在功能模型之外扩充了行为模型,用 CFD(Control Flow Diagram,控制流程图)、CSPEC 和 STD 等工具来描述。

3.5.2 数据建模(E-R 图)

为了把用户的数据要求清楚、准确地描述出来,分析员通常要建立一个概念模型。概念模型是面向问题的数据模型,是按照用户的观点对数据建立模型。在需求分析模型建立过程中,使用 E-R 图来建立数据模型。它描述了从用户的角度看到的数据,反映了用户的现实环境,而与在软件系统中的实现方法无关。在 E-R 模型中,数据模型包括三种互相关联的信息,即数据对象、描述对象的属性、描述对象间相互连接的关系。

E-R 图中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的属性,并用直线把实体与其属性连接起来。例如,图 3.6 是商品和供应商的实体关系图。



图 3.6 商品实体表示方法

人们通常就是用实体、关系和属性这三个概念来理解现实问题的,因此,E-R 图比较接近人们的思维方式,此外,E-R 图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 图可以作为用户与分析员之间一种有效的交流工具。

(1) 数据对象(实体型):是需要被目标系统所理解的复合信息的表示。所谓复合信息是具有若干不同特征或属性的信息。

数据对象可以是外部实体(如显示器)、事物(如报表或显示)、角色(如教师或学生)、行为(如一个电话呼叫)或事件(如单击鼠标左键)、组织单位(如研究生院)、地点(如注册室)或结构(如文件)。

数据对象只封装了数据,没有包含作用于这些数据上的操作。这与面向对象范型中的类和对象不同。具有相同特征的数据对象组成的集合仍然称为数据对象,其中的某一个对象叫作该数据对象的一个实例。例如,商品实体用图 3.6 表示。

(2) 属性:定义了数据对象的特征。它可用来:①为数据对象的实例命名;②描述这个实例;③建立对另一个数据对象的另一个实例的引用。例如,商品实体有商品编号、商品名称、规格、类型等属性。为了唯一地标识数据对象的某一个实例,定义数据对象中的一个属性或几个属性为关键码(key),书写为“_id”。例如,在“商品”数据对象中用“商品编号”作为关键码,它可以唯一地标识一个“商品”数据对象中的实例。

属性是实体的说明。用椭圆形表示实体的属性,并用无向边把实体与其属性连接起来。则其 E-R 图如图 3.7 所示。

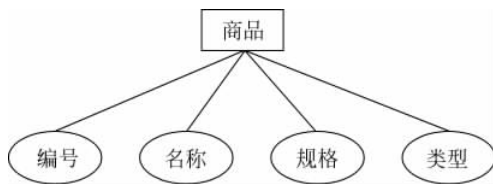


图 3.7 商品实体及其属性

(3) 关系:数据对象及其关系可用 E-R 图表示。各个数据对象的实例之间有关联。如一名学生“张鹏”选修两门课程“软件工程”与“计算机网络”,学生与课程的实例通过“选修”关联起来。实例的关联有以下三种。

① 一对一(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,则部门与经理的关系就是一对一的。

② 一对多(1:m)。例如,某校教师与课程之间存在一对多的关系“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任课。

③ 多对多($n:m$)。例如,商品与供应商之间的关系是多对多的,即一种商品可以有多个供应商,而每个供应商可以有多种商品,如图 3.8 所示。

(4) 实体关系图:实体间的联系是两个或两个以上实体类型之间的有名称的关联。实体间的联系用菱形表示,菱形内要有联系名,并用无向边把菱形分别与有关实体相连接,在无向边旁标上联系的类型。例如,可以用 E-R 图来表示商品和供应商关系的概念模型,如图 3.8 所示。

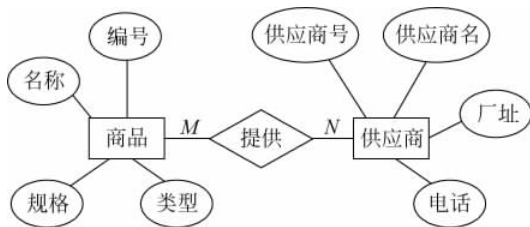


图 3.8 实体、实体属性及实体联系模型图

如果概念模型中涉及的实体带有较多的属性而使实体联系图非常不清晰,可以将实体联系图分成两部分,一部分是实体及其属性图,另一部分是实体及其联系图。

(5) 数据的规范化:信息域分析需要确定数据的内容,每个数据项要用表格列出,最后组织成文件的逻辑结构,即面向应用而不是面向存储的结构。为了便于数据库的设计,常常要对这种结构做一些简化,其中最常见的一种方法就是规范化技术。

“规范化”将数据的逻辑结构归结为满足下列条件的二维表(关系)。

- ① 表中每个信息项必须是一个不可分割的数据项,不可以是组项。
- ② 表格中每一列中所有信息项必须是同一类型,各列的名字(属性名)互异,列的次序可以任意。
- ③ 表格中各行互不相同,行的次序任意。

不满足上述要求的二维表或关系,就叫作非规范化关系。必须将其规范化成单纯的和规范的关系。

规范化的目的就是要消除数据冗余,即消除表格中数据的重复;消除多义性,使关系中的属性含义清楚、单一;使关系的“概念”单一化,让每个数据项只是一个简单的数或字符串,而不是一个组项或重复组;方便操作;使数据的插入、删除与修改操作可行并方便;使关系模式更灵活,易于实现接近自然语言的查询方式。

【案例 3-2】 教学管理系统

案例 3-2“教学管理系统”的数据库概念模型如下。

用三个关系学生、教师、课程保存三个实体型的信息:

学生(学号,姓名,性别,年龄,专业,籍贯)

教师(职工号,姓名,年龄,职称,工资级别,工资)

课程(课程号,课程名,学分,学时,课程类型)

建立两个关系表示实体型之间的联系:

选课(学号,课程号,听课出勤率,作业完成率,分数)

教课(职工号,课程号)

这 5 个关系就组成了数据库的概念模型。在每个关系中,属性名下加下画线指明关键字,并规定关键字能唯一地标识一个元组。关键字可以由一个或一组属性组成。

关系规范化的程度,通常按属性间的依赖程度来区分,并以范式 NF (Normal Form) 来表达。常用的范式分为第一范式(1NF)、第二范式(2NF)和第三范式(3NF)。

判断规范化程度的条件如下。

- (1) 关系中所有属性都是“单纯域”,即不出现“表中有表”;
- (2) 非主属性完全函数依赖于关键字;
- (3) 非主属性相互独立,即任何非主属性间不存在函数依赖。

如果一个关系连条件(1)都不满足,则这个关系是非规范化的;如果一个关系仅满足条件(1),则这个关系满足第一范式(1NF);如果一个关系满足条件(1)、(2),但不满足(3),则这个关系满足第二范式(2NF);如果一个关系同时满足条件(1)、(2)和(3),则这个关系满足第三范式(3NF)。当数据模型达到 3NF 时,一般情况下就能满足数据库应用的需要。

3.5.3 功能建模(数据流图)

数据流图是结构化方法的需求分析最基本的工具,数据流图从数据传递和加工的角度,以图形化的方式刻画数据流从输入到输出的移动和变换过程。在数据流图中具体的物理元素都已去掉,只剩下数据的存储、流动、加工和使用情况。这种抽象性能使我们总结出信息处理的内部规律性。由于数据流图是用图形来表示逻辑系统,即使不是专业的计算机人员也能比较容易地理解数据流图,因此数据流图成为一种极好的交流工具。

1. 数据流图的基本表示符号

1) 数据流图的 4 种基本表示符号

图 3.9 是描述储户携带存折去银行办理取款手续的数据流图。

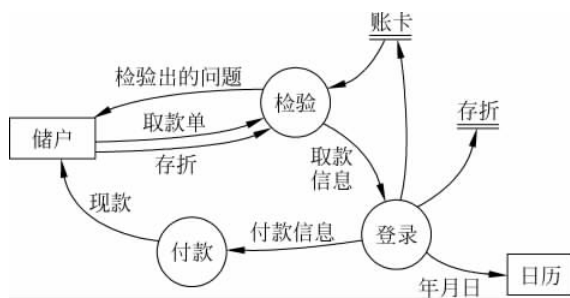


图 3.9 办理取款手续的数据流图

从图中可以看到,数据流图的基本图形元素有 4 种,如图 3.10 所示。

在数据流图中,如果有两个以上数据流指向一个加工,或是从一个加工中引出两个以上的数据流,这些数据流之间往往存在一定的关系。为表达这些关系,对这些数据流的加工可以标上不同的标记符号。所用符号及其含义在图 3.11 中给出。

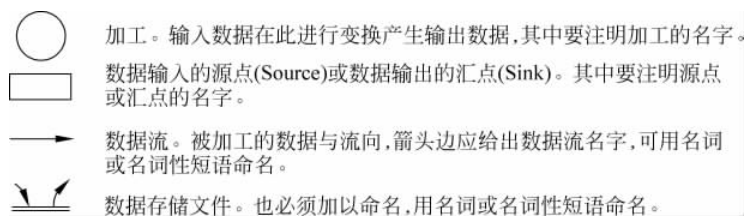


图 3.10 DFD 的基本图形符号

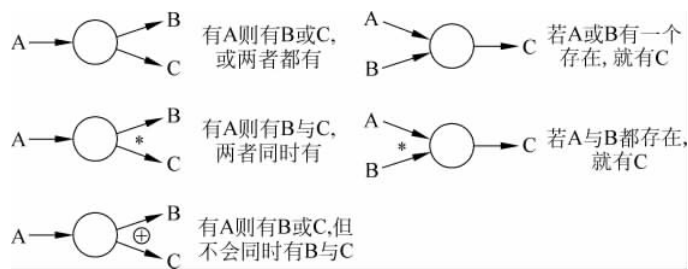


图 3.11 表明多个数据流与加工之间关系的符号

2) 分层数据流图的表示

为了表达数据处理过程的数据加工情况,用一个数据流图是不够的。稍复杂的实际问题,在数据流图上常常出现十几个甚至几十个加工。这样的数据流图看起来很不清楚。层次结构的数据流图能很好地解决这一问题。按照系统的层次结构进行逐步分解,并以分层的数据流图反映这种结构关系,能清楚地表达和容易理解整个系统。

图 3.12 给出分层数据流图的示例。数据处理 S 包括三个子系统 1、2、3。顶层 LO 下面的第一层数据流图 DFD/L1。第二层数据流图 DFD/L2.1、DFD/L2.2 及 DFD/L2.3 分别是子系统 1、2 和 3 的细化。对任何一层数据流图来说,我们称它的上层图为父图,在它下一层的图则称为子图。

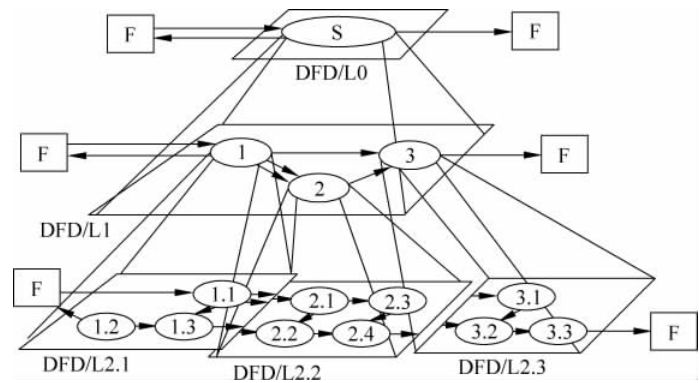


图 3.12 分层数据流图

3) 数据流图 4 种基本符号的另一种表示方法

为了使数据流图便于在计算机上输入和输出,免去画曲线、斜线、圆的困难,常常使用如

图 3.13 所示的另一套符号,这一套符号与图 3.10 中给出的符号是完全等价的。

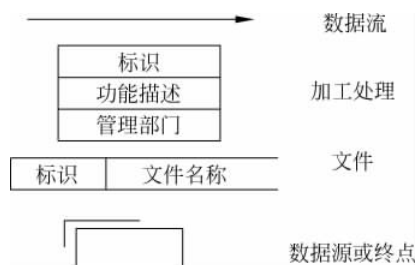


图 3.13 数据流图的基本符号

图 3.14 是一个简单的数据流图,它表示数据 X 从源 S 流出,经 P1 加工转换成 Y,接着经 P2 加工转换为 Z,在加工过程 P2 中从 F 中读取数据。

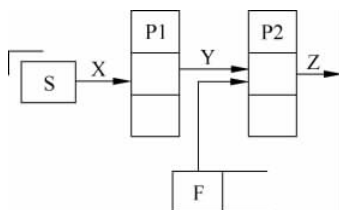


图 3.14 数据流图举例

2. 数据流图中的主要元素

1) 数据流

数据流由一组确定的数据组成。例如,“发票”为一个数据流,它由品名、规格、单位、单价、数量等数据组成。数据流用带有名字的具有箭头的线段表示,名字称为数据流名,表示流经的数据,箭头表示流向。数据流表明了数据的流动方向及其名称,它是数据载体的表现形式。在数据流的上方写上数据流的名称。数据流可以从加工流向加工,也可以从加工流进、流出文件,还可以从源点流向加工或从加工流向终点。图 3.15 是常见的几种可能的数据流。

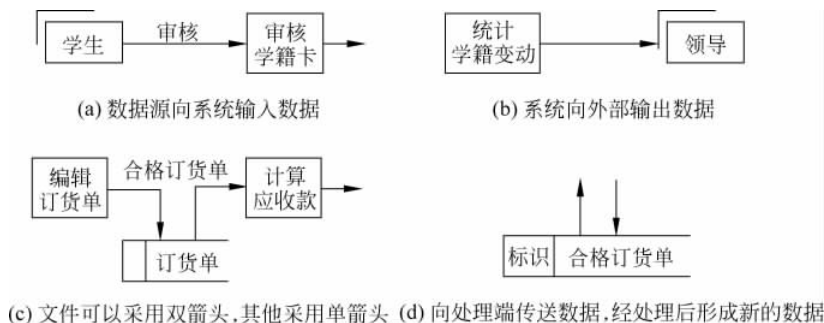


图 3.15 常见的几种可能的数据流

对数据流的表示有以下约定。

(1) 对流进或流出文件的数据流无须标注名字,因为文件本身就足以说明数据流。而其他数据流则必须标出名字,名字应能反映数据流的含义。

(2) 数据流不允许同名。

(3) 两个数据流在结构上相同是允许的,但必须体现人们对数据流的不同理解。

(4) 两个加工之间可以有几股不同的数据流,这是由于它们的用途不同,或它们之间没有联系,或它们的流动时间不同。

(5) 数据流图描述的是数据流而不是控制流。

2) 加工处理

加工处理是对数据流进行的操作,它把流入的数据流转换为流出的数据流。每个加工处理都应取一个名字表示它的含义,并规定一个编号用来标识该加工在层次分解中的位置。名字中必须包含一个动词,例如“计算”“打印”等。

对数据流加工转换的方式有以下两种。

(1) 改变数据流的结构,例如将数组中各数据重新排序。

(2) 产生新的数据流,例如对原来的数据进行汇总计、求平均值等。

加工规格说明用来说明 DFD 中的数据加工的加工细节。加工规格说明描述了数据加工的输入、实现加工的算法以及产生的输出。另外,加工规格说明指明了加工(功能)的约束和限制、与加工相关的性能要求,以及影响加工的实现方式的设计约束。必须注意,编写加工规格说明的主要目的是要表达“做什么”,而不是“怎样做”。因此它应描述数据加工实现加工的策略而不是实现加工的细节。

目前,用于描写加工规格说明的工具具有结构化英语、判定表和判定树。

3) 数据存储

文件是存储数据的工具。文件名应与它的内容一致,写在开口长条内。从文件流入或流出数据流时,数据流方向是很重要的。如果是读文件,则数据流的方向应从文件流出,写文件时则相反;如果是又读又写,则数据流是双向的。在修改文件时,虽然必须首先读文件,但其本质是写文件,因此数据流应流向文件,而不是双向。数据流从文件流出,箭头应指向加工。

4) 数据源或终点

数据源和终点表示数据的外部来源和去处。它通常是系统之外的人员或组织,不受系统控制。为了避免在数据流图上出现线条交叉,同一个源点、终点或文件均可在不同位置多次出现,这时要在源(终)点符号的右下方画小斜线,或在文件符号左边画竖线,以示重复。

由图 3.14 可见,数据流图可通过基本符号直观地表示系统的数据流程、加工、存储等过程。但它不能表达每个数据和加工的具体、详细的含义,这些信息需要在“数据字典”和“加工说明”中表达。

3. 数据流图的画法

画数据流图的基本步骤概括地说,就是自外向内,自顶向下,逐层细化,完善求精。先确定系统的边界或范围,再考虑系统的内部,先画加工的输入和输出,再画加工的内部。即:

- (1) 识别系统的输入和输出。
- (2) 从输入端至输出端画数据流和加工,并同时加上文件。
- (3) 加工“由外向里”进行分解。
- (4) 数据流的命名,名字要确切,能反映整体。
- (5) 各种符号布置要合理,分布均匀,尽量避免交叉线。

对于不同的问题,数据流图可以有不同的画法。具体实行时,可按如下案例的步骤进行。

案例 3-1“库存管理系统”的顶层数据流图如下。

- (1) 识别系统的输入和输出,画出顶层图。

即确定系统的边界。在需求阶段,系统的功能需求等还不很明确,为了防止遗漏,不妨先将范围定得大一些。系统边界确定后,那么越过边界的数据流就是系统的输入或输出,将输入与输出用加工符号连接起来,并加上输入数据来源和输出数据去向就形成了顶层图。

画出库存管理系统顶层图的主要目的是识别系统的输入和输出,如图 3.16 所示。

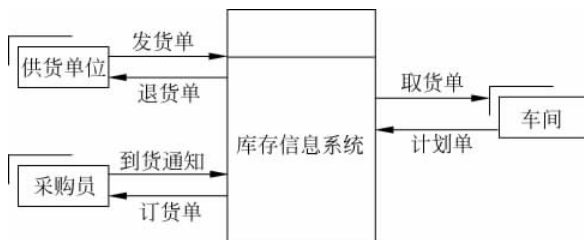


图 3.16 库存管理系统的顶层数据流图

- (2) 画出系统内部的数据流、加工与文件,画出一级细化图。

从系统输入端到输出端(也可反之),逐步用数据流和加工连接起来,当数据流的组成或值发生变化时,就在该处画一个“加工”符号。画数据流图时还应同时画上文件,以反映各种数据的存储位置,并表明数据流是流入还是流出文件。

最后,再回过头来检查系统的边界,补上遗漏但有用的输入输出数据流,删去那些没被系统使用的数据流。

- (3) 加工的进一步分解,画出二级细化图。

同样运用“由外向里”的系统方式对每个加工进行分析,如果在该加工内部还有数据流,则可将该加工分成若干个子加工,并用一些数据流把子加工连接起来,即可画出二级细化图。二级细化图可在一级细化图的基础上画出,也可单独画出该加工的二级细化图,二级细化图也称为该加工的子图。需要注意编号问题,即对上一级加工的分解,应以上一级编号为开始,加一个“.”并依次编号。例如,对 1 号加工的分解,编号为 1.1、1.2、...,以此类推。

- (4) 其他注意事项。

一般应先给数据流命名,再根据输入/输出数据流名的含义为加工命名。名字含义要确切,要能反映相应的整体。若碰到难以命名的情况,则很可能是分解不恰当造成的,应考虑重新分解。

从左至右画数据流图。通常左侧、右侧分别是数据源和终点,中间是一系列加工和文件。正式的数据流图应尽量避免线条交叉,必要时可用重复的数据源、终点和文件符号。此

外,数据流图中各种符号布置要合理,分布应均匀。

案例 3-1“库存管理系统”数据流图的一级细化图,如图 3.17 所示。

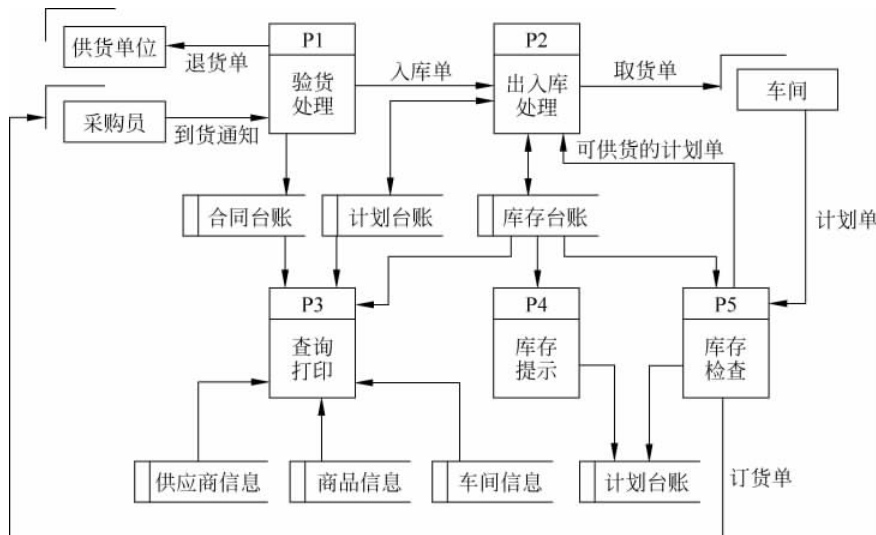


图 3.17 库存管理系统的数据流图的一级细化图

画数据流图是一项艰巨的工作,要做好重画的思想准备,重画是为了消除隐患,有必要不断改进。

因为作为顶层加工处理的改变域是确定的,所以改变域的分解是严格按自顶向下分解的。由于目标系统目前还不存在,因此分解时开发人员还需凭经验进行,这是一项创造性的劳动。同时,在建立目标系统数据流图时,还应充分利用本章讲过的各种方法和技术,例如,分解时尽量减少各加工之间的数据流;数据流图中各个成分的命名要恰当;父图与子图间要注意平衡等。

当画出分层数据流图,并为数据流图中各个成分编写词典条目或加工说明后,就获得了目标系统的初步概念模型。

4. 检查和修改数据流图应遵循的基本原则

- (1) 数据流图上所有图形符号只限于前述 4 种基本图形元素。
- (2) 顶层数据流图必须包括前述 4 种基本元素,缺一不可。
- (3) 顶层数据流图上的数据流必须封闭在外部实体之间。
- (4) 每个加工至少有一个输入数据流和一个输出数据流。
- (5) 在数据流图中,需按层给加工框编号。编号表明该加工处在哪一层,以及上下层的父图与子图的对应关系。
- (6) 规定任何一个数据流子图必须与它上一层的一个加工对应,两者的输入数据流和输出数据流必须一致。即:父图与子图的平衡。
- (7) 可以在数据流图中加入物质流,帮助用户理解数据流图。
- (8) 图上每个元素都必须有名字。数据流和数据文件的名字应当是“名词”或“名词性短语”,表明流动的数据是什么。加工的名字应当是“名词+宾语”,表明做什么事情。

(9) 数据流图中不可夹带控制流。

(10) 初画时可以忽略琐碎的细节,以集中精力于主要数据流。

5. 绘制分层数据流图时应注意的问题

1) 合理编号

分层数据流图的顶层称为 0 层,称它是第 1 层的父图,而第 1 层既是 0 层图的子图,又是第 2 层图的父图,以此类推。由于父图中有的加工可能就是功能单元,不能再分解,因此父图拥有的子图数少于或等于父图中的加工个数。为了便于管理,应按下列规则为数据流图中的加工编号。

(1) 子图中的编号由父图号和子加工的编号组成。

(2) 子图的父图号就是父图中相应加工的编号。

为简单起见,约定第 1 层图的父图号为 0,编号只写加工编号 1、2、3、...,下面各层由父图号 1、1.1 等加上子加工的编号 1、2、3、...组成。按上述规则,图的编号既能反映出它所属的层次以及它的父图编号的信息,还能反映子加工的处理信息。例如,1 表示第 1 层图的 1 号加工处理,1.1、1.2、1.3、...表示父图为 1 号加工的子加工,1.3.1、1.3.2、1.3.3、...表示父图号为 1.3 加工的子加工。

为了方便,对数据流图中的每个加工,可以只标出局部号,但在加工说明中,必须使用完整的编号。例如,图 3.18 可表示第 1 层图的 1 号加工的子图,编号可以简化成图中的形式。

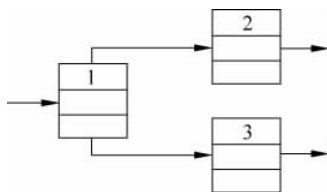


图 3.18 简化子图编号示例

2) 注意子图与父图的平衡

子图与父图的数据流必须平衡,这是分层数据流的重要性质。这里的平衡指的是子图的输入、输出数据流必须与父图中对应加工的输入、输出数据流相同,如图 3.19 所示。

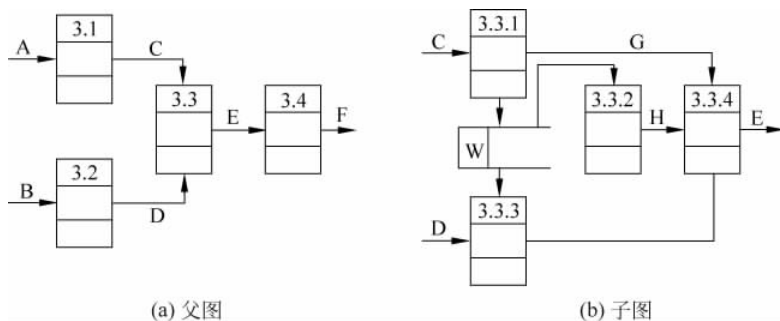


图 3.19 数据流图中的局部文件

3) 局部文件

图 3.19 中的父图和子图是平衡的,但子图中的文件 W 并未在父图中出现。这是由于

对文件 W 的读、写完全局限在加工 3.3 之内,在父图中各个加工之间的界面上不出现,该文件是子图的局部文件或为临时文件。

应当指出的是,如果一个临时文件在某层数据流图中的某些加工之间出现,则在该层数据流图中就必须画出这个文件。一旦文件被单独画出,那么也需画出这个文件同其他成分之间的联系。

4) 分解的程度

对于规模较大的系统的分层数据流图,如果一下子把加工直接分解成基本加工单元,一张图上画出过多的加工将使人难以理解,也增加了分解的复杂度。然而,如果每次分解产生的子加工太少,会使分解层次过多而增加作图的工作量,阅读也不方便。经验表明,一般说来一个加工每次分解量以最多不要超过 7 个为宜。同时,分解时应遵循以下原则。

(1) 分解应自然,概念上要合理、清晰。

(2) 上层可分解得快些(即分解成的子加工个数多些),这是因为上层是综合性描述,对可读性的影响小;而下层应分解得慢些。

(3) 在不影响可读性的前提下,应适当地多分解成几部分,以减少分解层数。

(4) 一般说来,当加工用一页纸可以明确地表述时,或加工只有单一输入/输出数据流时(出错处理不包括在内),就应停止对该加工的分解。另外,对数据流图中不再分解的加工(即功能单元),必须做出详细的加工说明,并且每个加工说明的编号必须与功能单元的编号一致。

6. 数据流图的修改

前面介绍了画数据流图的基本方法。对于一个大型系统来说,由于在软件设计初期人们对于问题理解的深度不够,在数据流图上也不可避免地会存在某些缺陷或错误,因此还需要进行修改,才能得到完善的数据流图。这里介绍如何从正确性和可读性方面对数据流图进行改进。

(1) 正确性: 数据流图的正确性,可以从以下几个方面来检查。

① 文件使用: 在数据流图中,文件与加工之间数据流的方向应按规定认真标注,这样也有利于对文件使用正确性的检查。例如,在图 3.20 中,因为文件 1 和文件 2 是子图的局部文件,所以在子图中应画出对文件的全部引用。但子图中文件 2 好像一个“渗井”,数据只有流进,没有流出,显然是一个错误。

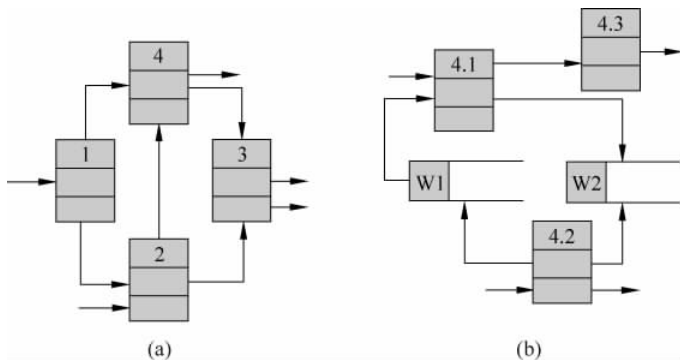


图 3.20 局部文件使用错误

② 子图与父图间的平衡：造成子图与父图不平衡的一个常见原因是在增加或删除一个加工时，忽视了对相应父图或子图的修改。在检查数据流图时应注意这一点。

③ 加工与数据流的命名：加工和数据流的名字必须体现被命名对象的全部内容而不是一部分。对于加工的名字，应检查它的含义与被加工的输入/输出数据流是否匹配。

一个加工的输出数据流仅由它的输入数据流确定，这个规则绝不能违背。数据不守恒的错误有两种，一是漏掉某些输入数据流，二是某些输入数据流在加工内部没有被使用。虽然有时后者并不一定是个错误，但也要认真考虑，对于确实无用的数据就应该删去，以简化加工之间的联系。在检查数据流图时，应注意消除控制流。

(2) 可读性：数据流图的可读性，可以从以下几个方面来提高。

① 简化加工之间的联系：各加工之间的数据流越少，各加工的独立性就越高。因此应当尽量减少加工之间数据流的数目，必要时可采用后面要介绍的步骤对数据流图重新分解。

② 分解应当均匀：在同一张数据流图上，应避免出现某些加工已是功能单元，而另一些加工却还应继续分解好几层的情况出现，否则应考虑重新分解。

③ 命名应当恰当：理想的加工名由一个具体的动词和一个具体的宾语(名词)组成。数据流和文件的名字也应具体、明确。

(3) 数据流图重新分解的步骤。

有时需要对做出的部分或全部数据流图做重新分解，可按以下步骤进行。

① 把需要重新分解的所有子图连成一张；

② 根据各部分之间联系最少的原则，把图划分成几部分；

③ 重建父图，即把第二步所得的每一部分画成一个圆圈，各部分之间的联系就是加工之间的界面；

④ 重建各张子图，只需把第二步所得的图按各自的边界剪开即可；

⑤ 为所有加工重新命名、编号。

例如，图 3.21(a) 中加工 2 和其他加工的联系太复杂以致很难独立理解，所以其结构不太合理。将它们的父图加工分成两个更为合适，如图 3.21(b) 所示。

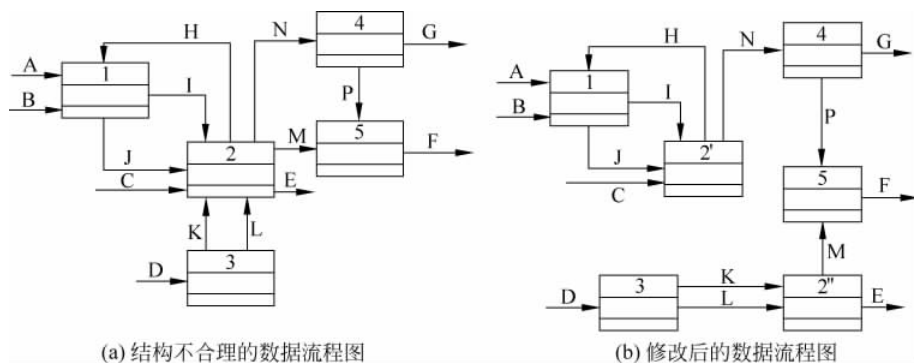


图 3.21 结构不合理的数据流图及其修改

根据上述规则检查图 3.17 发现案例 3-1“库存管理系统”存在如下问题。

(1) 一层细化图缺少了一个输入数据流。应在供应商和处理 P1 之间加一个数据流“发

货单”，数据从供应商流入处理 P1。

(2) 文件使用错误：供应商信息、车间信息和商品信息三个文件在一层细化图中只有流进，没有流出。应在顶层数据流图中添加“供应商信息”“车间信息”和“商品信息”三个数据存储，系统读取三个文件，因此数据从文件流入“库存管理系统”处理。

3.5.4 行为建模(状态迁移图)

行为建模给出需求分析方法的所有操作原则，但只有结构化分析方法的扩充版本才提供这种建模的符号。

1. 状态迁移图

利用如图 3.22 所示的状态迁移图(STD)或状态迁移表来描述系统或对象的状态，以及导致系统或对象的状态改变的事件，从而描述系统的行为。

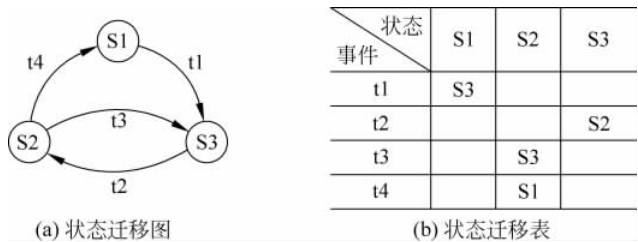


图 3.22 状态迁移图与其等价的状态迁移表示例

每一个状态代表系统或对象的一种行为模式。状态迁移图指明系统的状态如何响应外部的信号(事件)进行推移。在状态迁移图中，用圆圈“○”表示可得到的系统状态，用箭头“→”表示从一种状态向另一种状态的迁移。在箭头上要写上导致迁移的信号或事件的名字。如图 3.22(a)所示，系统中可取得的状态 = S1, S2, S3，事件 = t1, t2, t3, t4。事件 t1 将引起系统状态 S1 向状态 S3 迁移，事件 t2 将引起系统状态 S3 向状态 S2 迁移，等等。图 3.22(b)就是与图 3.22(a)等价的状态迁移表。

另外，状态迁移图指明了作为特定事件的结果(状态)。在状态中包含可能执行的行为(活动或加工)。

如果系统比较复杂，可以把状态迁移图分层表示。例如，在确定了如图 3.23 所示那样的状态 S1, S2, S3 之后，接下来就可把状态 S1, S2, S3 细化。在该图中对状态 S1 进行了细化。此外，在状态迁移图中，由一个状态和一个事件所决定的下一状态可能会有多个。实际会迁移到哪一个是由更详细的内部状态和更详细的事件信息来决定的。此时，可采用状态迁移图的一种变形，如图 3.24 所示，使用加进判断框和处理框的记法。

2. Petri 网

Petri 网，简称 PNG (Petri Net Graph)，它适用于描述相互独立、协同操作的处理系统，即并发执行的处理系统。在软件需求分析与设计阶段都可以使用。

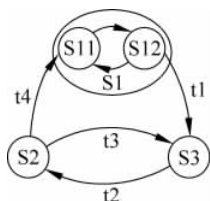


图 3.23 状态迁移图的网

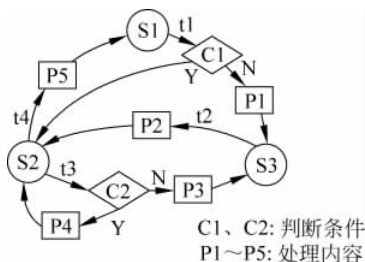


图 3.24 状态迁移图的变形

Petri 网是一种有向图,它有两种结点:“○”表示系统的状态;“—”或“|”表示系统中的事件。图中的有向边表示对事件的输入,或从事件输出:“→”表示对事件的输入;“└→”表示事件的结果,即从事件的输出。

图 3.25 用 Petri 网描述了一个多任务系统中的两个进程 PR1 和 PR2 使用一个公共资源 R 时,利用原语 LOCK(对资源加锁)和 UNLOCK(对资源解锁)控制 R 的使用,保证进程间的同步的例子。

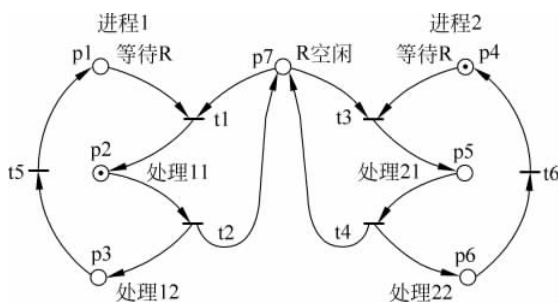


图 3.25 进程同步机制的 PNG

图中每个进程是一个数据对象,它有三个状态:等待资源(p1 或 p4),占用资源执行的执行(p2 或 p5),不占用资源执行的执行(p3 或 p6)。另外,系统有一个状态:资源空闲(p7)。在有的状态中有一个黑点“●”,称为标记或令牌,表明系统或对象当前正处于此状态。当作为一个事件的输入的所有状态都得到或保有令牌时,才能引起该事件“激发”。使得系统和对象的状态向前推移,完成系统和对象的某些行为。

3. 控制规格说明

控制规格说明从两个方面给出系统的行为。其一是状态迁移图(STD),它是行为的“顺序规格说明”;其二是加工激活表(PAT),它是行为的“组合规格说明”,表明当事件激发时,数据流图中的哪些加工要被激活。控制规格说明仅描述了系统的行为,不提供被激活的加工的内部的细节。

3.5.5 数据字典

分析模型中包含对数据对象、功能和行为的表示。在每一种表示中,数据对象和行为项

都扮演一定的角色。为表示每个数据对象和行为项的特性,建立了数据字典。

数据字典精确地、严格地定义了每一个与系统相关的数据元素,并以字典式顺序将它们组织起来,使得用户和分析员对所有的输入、输出、存储成分和中间计算有共同的理解。

在数据流图的基础上,还需对其中的每个数据流、文件和数据项加以定义,我们把这些定义所组成的集合称为数据字典。数据流图是系统的大框架,而数据字典以及下面将要介绍的加工说明则是对数据流图中每个成分的精确描述。它们有着密切的联系,必须结合使用。以下是有关词条的描述方法。

在数据字典的每一个词条中应包含以下信息。

- (1) 名称: 数据对象或控制项、数据存储或外部实体的名字。
- (2) 别名或编号。
- (3) 分类: 数据对象? 加工? 数据流? 数据文件? 外部实体? 控制项(事件 / 状态)?
- (4) 描述: 描述内容或数据结构等。
- (5) 何处使用: 使用该词条(数据或控制项)的加工。

在数据字典的编制中,分析员最常用的描述内容或数据结构的符号如表 3.4 所示。

表 3.4 数据字典定义式中的符号

符号	含 义	解 释
=	被定义为	
+	与	例如, $x=a+b$, 表示 x 由 a 和 b 组成
[...,...]	或	例如, $x=[a, b]$, $x=[a b]$, 表示 x 由 a 或由 b 组成
[... ...]	或	
{ ... }	重复	例如, $x=\{a\}$, 表示 x 由 0 个或多个 a 组成
$m\{...\}n$	重复	例如, $x=3\{a\}8$, 表示 x 中至少出现 3 次 a , 至多出现 8 次 a
(...)	可选	例如, $x=(a)$, 表示 a 可在 x 中出现, 也可不出现
“...”	基本数据元素	例如, $x=“a”$, 表示 x 为取值为 a 的数据元素
..	连接符	例如, $x=1..9$, 表示 x 可取 1~9 之中的任一值

【案例 3-3】 数据文件“存折”

案例 3-3“数据文件‘存折’”在数据字典中的定义格式

在图 3.9 表示的取款数据流图中,数据文件“存折”的格式如图 3.26 所示,它在数据字典

户名	所号	账号
开户日	性质	印密
日期 年月日	摘要	支出
	存入	余额
	操作	复核

图 3.26 存折格式

典中的定义格式如表 3.5 所示。

表 3.5 数据文件“存折”在数据字典中的定义格式

名称	定 义	注 释
存 折	户名+所号+账号+开户日+性质+(印密)+1{存取行}50	
户 名	2{字母}24	
所 号	“001”.. “999”	
账 号	“00000001”... “99999999”	储蓄所编码,规定 3 位数字
开户日	年+月+日	账号规定由 8 位数字组成
性 质	“1”...“6”	
印 密	“0”	
存取行	日期+(摘要)+支出+存入+余额+操作+复核	“1”普通户,“5”工资户等
日 期	年+月+日	印密在存折上不显示
.....		
字 母	[“a”.. “z” “A”.. “Z”]	

在数据字典中有三种类型的条目：数据流条目、文件条目和数据项条目。下面分别进行举例说明。

1. 数据项条目

数据项条目用来给出数据项的定义。由于数据项是数据的最小单位,是不可分割的,因此数据项条目只包含名称、代码、类型、长度和值的含义内容等。对于那些足以从名称看出其含义的“自说明”型的数据项,则不必在条目中再解释其含义。

案例 3-1“库存管理系统”中的部分数据项条目如表 3.6 所示。

表 3.6 管理系统数据项条目

名 称	说 明
数据项编号:	1-01
数据项名称:	商品编号
别 名:	无
简 述:	某种商品的编号
类 型:	字符型
长 度:	8 字节
取值范围:	数字 + 英文字母
数据项编号:	1-02
数据项名称:	单价
别 名:	购入单价
简 述:	某种商品的购入单价
类 型:	数值型
长 度:	10 位,小数位两位
取值范围:	0.00~9999999.99
数据项编号:	1-03
数据项名称:	库存数量

续表

名 称	说 明
别 名:	实际库存数量
简 述:	某种商品的库存数量
类 型:	数值型
长 度:	5 位整数
取值范围:	0~99999

2. 数据流条目

数据流条目对每个数据流进行定义,它通常由数据流名、别名、组成、注释、流入、流出和流量等部分组成。其中,别名是前面已定义的数据流的同义词;组成栏是定义的主要部分,通常是列出该数据流的各组成数据项;注释栏用于记录其他有关的信息,例如该数据流在单位时间内传输的次数等。

如果数据流的组成很复杂,则可采用“自顶向下,逐步分解”的方式来表示。例如,“供应商信息”数据流可写成:

供应商信息 = 供应商基本信息 + 供应商通信信息 + 供应商经营信息

案例 3-1“库存管理系统”数据流条目如表 3.7 所示。

表 3.7 库存管理系统数据流条目

名 称	说 明
数据流名称:	入库单
编 号:	F1
简 述:	采购人员填写的商品入库凭单
数据流来源:	采购人员
数据流去向:	登记库存台账
数据流组成:	日期 + 入库单编号 + 商品编号 + 购入数量
流 通 量:	25 份/天
高峰流通量:	50 份/天
数据流名称:	发货单
编 号:	F2
简 述:	供应商填写的商品发货凭单
数据流来源:	供应商
数据流去向:	登记合同台账
数据流组成:	日期 + 发货单编号 + 供应商编号 + 商品编号 + 发货数量
流 通 量:	25 份/天
高峰流通量:	50 份/天

3. 文件条目

文件条目用来对文件(或数据库)进行定义。它由 5 部分组成:文件名,编号,组成,结构和注释。其中,组成栏的定义方法与前面的数据流条目相同;结构栏用于说明重复部分

的相互关系,比如指出是顺序或索引存取。

数据字典明确地定义了各种信息项。随着系统规模的扩大,数据字典的规模和复杂性将迅速增加。

案例 3-1“库存管理系统”文件条目如表 3.8 所示。

表 3.8 库存管理系统文件条目

名 称	说 明
文件名称:	库存台账
文件编号:	D1
简 述:	记录商品的编号、名称、单价与库存数量等信息
组 成:	商品编号 + 购入单价 + 库存数量
关 键 字:	商品编号
文件名称:	合同台账
文件编号:	D2
简 述:	记录合同的编号、供应商编号、货物编号、单价与购入数量等信息
组 成:	合同编号 + 供应商编号 + 商品编号 + 单价 + 购入数量 + 日期 + 合同状态
关 键 字:	合同编号
文件名称:	计划台账
文件编号:	D3
简 述:	记录计划的编号、车间编号、商品编号、数量等信息
组 成:	计划编号 + 供应商编号 + 商品编号 + 数量 + 日期 + 计划状态
关 键 字:	计划编号

3.5.6 加工说明的描述工具

由于自然语言不够精确、简练,不适合编写加工说明。目前有许多适用加工说明的描述工具。下面介绍三种最常用的工具:结构化语言、判定表和判定树。

1. 结构化语言

自然语言的优点是容易理解,但是它不精确,可能有多义性。程序设计语言的优点是严格精确,但它的语法规则太死板,使用不方便。结构化语言(Structured Language)则是介于自然语言和程序设计语言之间的一种语言,它是带有一定结构的自然语言。在我国,通常采用较易为用户和开发人员双方接受的结构化汉语。

在用结构化语言描述问题时只允许使用基本逻辑结构、顺序结构、选择结构和循环结构这三种结构。配合这三种结构所使用的词汇主要有三类:陈述句中的动词;在数据字典中定义的名词;某些逻辑表达式中的保留字、运算符、关系符等。后面还会具体说明这三种语句的使用方式。

案例 3-1“库存管理系统”中关于库存提示的加工说明如表 3.9 所示。

表 3.9 库存提示的加工说明

名 称	说 明
加 工 号：	库存提示
加工编号：	RS4
输 入：	功能代号 4
加工逻辑：	IF 库存数量>库存最高值 OR 库存数量<库存最低值 THEN 库存提示
输 出：	ELSE ...
注 释：	计划台账 按周期自动执行或不定期手工执行

为了降低复杂性,便于人们理解,编写加工说明需要注意以下几点。

- (1) 避免结构复杂的长句。
- (2) 所用名词必须在数据字典中有定义。
- (3) 不要用意义相同的多种动词,用词名应始终如一。例如,“修正”“修改”“更改”含义相同,一旦确定使用其中一个以后,就不要再其余两个。
- (4) 为提高可读性,书写时可采用“阶梯形”格式。
- (5) 嵌套使用各种结构时,应避免嵌套层次过多而影响可读性。

2. 判定表

对于具有多个互相联系的条件和可能产生多种结果的问题,用结构化语言描述则显得不够直观和紧凑,这时可以用以清楚、简明为特征的判定表(Decision Table)来描述。判定表采用表格形式来表达逻辑判断问题,表格分成 4 个部分:左上角为条件说明;左下角为行动说明;右上角为各种条件的组合说明;右下角为各条件组合下相应的行动。下面用例子来说明如何使用判定表。

表 3.10 是使用判定表描述订货折扣的政策问题的例子。其中,C1~C3 为条件,A1~A4 为行动,1~8 为不同条件的组合,Y 为条件满足,N 为不满足,X 为该条件组合下的行动。例如,条件 4 表示若交易额在 50 000 元以上、最近三个月中有欠款且与本公司交易在 20 年以下,则可享受 5%的折扣率。

表 3.10 判定表描述的折扣政策

		不同条件组合							
		1	2	3	4	5	6	7	8
条件	C1: 交易额在 50 000 元以上	Y	Y	Y	Y	N	N	N	N
	C2: 最近三个月无欠款单据	Y	Y	N	N	Y	Y	N	N
	C3: 与本公司交易 20 年以上	Y	N	Y	N	Y	N	Y	N
行动	A1: 折扣率 15%	√	√						
	A2: 折扣率 10%			√					
	A3: 折扣率 5%				√				
	A4: 无折扣率					√	√	√	√

判定表是根据条件组合进行判断的,表 3.10 中每个条件只存在“Y(是)”和“N(非)”两

种情况,所以三个条件共有 $2^3=8$ 种可能性。在实际使用中,有的条件组合可能是矛盾的,需要剔除,有的则可以合并。因此需在原始判定表的基础上进行整理和综合,才能得到简单明了且实用的判定表。同时,在整理过程中还可能对用户的原有业务过程进行改进和提高。表 3.11 是通过表 3.10 合并得到的,其中“—”表示“Y”或“N”均可。

表 3.11 合并整理后的判定表

		不同条件组合			
		1(1,2)	2(3)	3(4)	4(5,6,7,8)
条件	C1: 交易额在 50 000 元以上	Y	Y	Y	N
	C2: 最近三个月无欠款单据	Y	N	N	—
	C3: 与本公司交易 20 年以上	—	Y	N	—
行动	A1: 折扣率 15%	√			
	A2: 折扣率 10%		√		
	A3: 折扣率 5%			√	
	A4: 无折扣率				√

判定表的内容十分丰富,除了以上介绍的有限判定表(Limited Entry Table),根据表中条件取值的状态不同,还有扩展判定表(Extended Entry Table)和混合判定表(Mixed Entry Table)。它们都各有特色,若能合理选择和灵活运用,则可描述、处理更为广泛、复杂的判断过程。详细的内容可参阅有关的书籍。

3. 判定树

判定树(Decision Tree)是用来表示逻辑判断问题的一种图形工具。它用“树”来表达不同条件下的不同处理,比语言、表格的方式更为直观。判定树的左侧(称为树根)为加工名,中间是各种条件,所有的行动都列于最右侧。

表 3.11 描述的折扣政策可以用如图 3.27 所示的判定树来进行描述。



图 3.27 判定树描述的折扣政策

4. 几种表达工具的比较

以上介绍的三种用于描述加工说明的工具各自具有不同的优点和不足,它们之间的比较如表 3.12 所示。

通过比较可以看出它们的适用范围如下。

(1) 结构化语言最适用于涉及具有判断或循环动作组合顺序的问题。

- (2) 判定表较适用于含有 5 或 6 个条件的复杂组合,条件组合过于庞大则将造成不便。
- (3) 判定树适用于行动在 10~15 之间的一般复杂程度的决策。必要时可将判定表上的规则转换成判定树,以便于用户使用。
- (4) 判定表和判定树也可用于软件开发的其它阶段,并被广泛地应用于其他学科。

表 3.12 几种表达工具的比较

比 较 指 标	结构化语言	判 定 表	判 定 树
逻辑检查	好	很好	一般
表示逻辑结构	好(所有方面)	一般(仅是决策方面)	很好(仅是决策方面)
使用方便性	一般	一般	很好
用户检查	不好	不好(除非用户受过训练)	好
程序说明	很好	很好	一般
机器可读性	很好	很好	不好
机器可编辑性	一般(要求句法)	很好	不好
可变性	好	不好(除非是简单的组合变化)	一般

3.6 需求规格说明书

软件需求说明(Software Requirement Specification, SRS, 又称软件规格说明书)是分析员在需求分析阶段需要完成的文档,是软件需求分析的最终结果。它的作用主要是:作为软件人员与用户之间事实上的技术合同说明;作为软件人员下一步进行设计和编码的基础;作为测试和验收的依据。SRS 必须用统一格式的文档进行描述,为了使需求分析描述具有统一的风格,可以采用已有的且能满足项目需要的模板,也可以根据项目特点和软件开发小组的特点对标准进行适当的改动,形成自己的模板。软件需求说明主要包括引言、任务概述、需求规定、运行环境规定和附录等内容。

本节以库存管理系统软件为例来介绍如何撰写需求规格说明书。

案例 3-1“库存管理系统”软件需求规格说明书如下所示。

1. 引言

企业管理信息系统建设有利于企业科学化、合理化、制度化、规范化的管理,使企业的管理水平跨上新台阶,为企业持续、健康、稳定的发展打下基础。××公司为了提高库存周转率,加快资金周转速度,决定开发“库存管理系统”。该软件是以 SQL 作为实现语言,以 ASP.NET 作为主要技术手段。通过操作手册,用户可以了解本软件的工作过程,考虑到不同层次的用户,一方面可以通过简单的鼠标和键盘操作实现库存管理工作中的相关操作,另一方面也可以使有一定计算机基础的用户根据自己的需求对相关的内容进行相应的修改,以便适应本企业的实际情况。

1.1 编写目的

本需求规格说明书的编写目的在于研究库存管理系统软件的开发途径和应用方法。本需求规格说明书的预期读者是与库存系统开发相联系的决策人、开发组人员、辅助开发者、

使用本项目的领导、具体使用者和软件验证者。

1.2 项目范围

1.2.1 项目名称：库存管理系统。

1.2.2 项目开发者：库存管理系统开发小组。

1.2.3 项目用户：采购部门、车间、物资管理部门、供应商。

1.2.4 本产品将采购部门、车间、物资管理部门、供应商等联系到一起,便于进行出入库等部分的管理。

1.3 参考资料

1.3.1 张海藩, 软件工程. 北京: 清华大学出版社, 2009.

1.3.2 黄梯云, 管理信息系统. 北京: 高等教育出版社, 2016.

1.3.3 金雪云, ASP.NET 简明教程(C#篇). 北京: 清华大学出版社, 2006.

1.3.4 李长林, ASP.NET+SQL Server 动态网站开发与实例. 北京: 清华大学出版社, 2006.

2. 任务概述

2.1 产品概述

2.1.1 软件开发意图: 为了使库存管理系统更加完善, 以公司局域网作为基础, 利用计算机系统实现库存管理各个相关部门的联系, 减轻库存管理相关人员的工作负担, 实现对库存管理各种相关事务的统一管理。

2.1.2 该软件的应用目标: 通过本软件使管理人员利用计算机, 快速方便地进行出入库管理等相关事务, 使分散、杂乱的管理变得统一化。

2.1.3 该软件的作用范围: 本软件适应于中小型生产企业, 是比较完善的库存管理软件, 对出入库管理可以进行宏观调整、管理。

2.1.4 该软件开发背景: 为适应该行业的形势, 在外在资源得到充分保障之后, 为了使库存管理体系的管理工作, 特别是信息管理工作科学化、规范化, 以适应日益变化的时代, 公司的领导决定重新设计管理信息系统, 以现阶段的管理软件为基础, 形成一套完善的库存管理体系。

2.2 用户特点

本软件的使用对象是库存管理部门工作人员, 掌握一般计算机的基本操作就可以利用该软件进行相关操作。

2.3 条件与约束

2.3.1 项目的开发经费不超过 2 万元。

2.3.2 项目的开发时间不超过半年。

2.3.3 主负责人 1 名, 开发小组 3 人, 其他人员 2 名。

2.3.4 在管理方针、硬件限制、并行操作安全和保密方面有一定限制。

2.4 预计不良后果

假设开发经费不到位, 管理不完善, 数据处理不规范, 各部门需求分析调查不细致, 本项目的开发会受到很大影响。

3. 需求规定

3.1 对功能的规定

3.1.1 外部功能：该软件具有输入、输出和查询等功能。

3.1.2 内部功能：该软件集命令、编程和编辑于一体，完成过滤、定位显示等功能。

3.2 对性能的规定

3.2.1 精度：在精度需求上，根据需要使用，在各项数据的输入/输出及传输过程中，可以满足各种精度的需求。

3.2.2 时间特性要求：在软件、响应时间、更新处理时间方面满足用户要求。

3.2.3 灵活性：当用户需求，如操作方式、运行环境、结果精度、数据结构与其他软件接口等发生变化时，设计的软件能做适当调整，满足不同用户的要求。

3.3 输入/输出要求(略)

3.4 数据管理能力要求(略)

3.5 故障处理要求(略)

3.6 其他专门要求(略)

3.7 保密性(略)

4. 运行环境规定

4.1 设备

4.1.1 最低配置：600MHz Pentium III 处理器，160MB 内存，5GB 可用硬盘空间，DVD-ROM 驱动器。

4.1.2 开发环境：一台服务器作为数据服务器，一台高性能 PC 作为 Web 服务器和开发用机，硬件防火墙。

4.2 支持软件(略)

4.3 接口

4.3.1 用户接口：本产品的用户一般需要通过终端进行操作，进入主界面后单击相应的窗口，分别进入相对应的界面。不同部门人员的操作权限不同。用户对程序的维护要有备份。

4.3.2 软件接口：Windows XP 及以上操作系统，IE 6.0 及以上浏览器。

4.4 控制

本软件是以 SQL 语言来控制软件运行的。

3.7 需求验证

软件需求分析阶段的结果是软件开发项目的重要根据，大量统计数字表明，软件系统中约 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，对目标系统提出一组要求后，必须严格验证需求的正确性。这个环节的参与者有用户、管理部门、软件设计、编码和测试人员。一般说来，应该从下述几个方面进行验证。

1. 一致性

所有的软件需求都必须是统一的，任何一条需求都不能和其他需求矛盾。用自然语言

书写的软件需求规格说明书是难于验证的,特别是在目标规模大、软件需求规格说明书较长的时候,人工复审没有更好的方法进行测试。一些没有保证的、冗余的、遗漏和不一致的问题可能不容易被发现而被保留下来,为以后的软件设计留下隐患。为了克服这些困难,人们使用形式化的语言书写软件需求规格说明书,可以用软件工具验证需求的一致性。

2. 现实性

指定的需求应该是用现有的硬件技术和软件技术能够基本实现的。如果超出了现有的技术基础,就会增加软件实现的难度,提高软件开发成本,甚至导致软件开发的失败。由此,验证时,应该参照以往开发系统的经验,分析现有的软、硬件实现目标系统的可行性。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. 完整性与有效性

需求必须是完整的,软件需求规格说明书中必须包括用户需要的每个功能或性能。需求必须是正确有效的,确实能解决用户的实际问题。只有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。检验需求的完整性与有效性必须在用户的合作下才能完成。而大多用户并不能清楚地说明自己的需求,也不能根据软件需求规格说明书确认是否满足自己的实际需求。使用快速原型方法是比较现实的解决方案。让用户试用一段时间原型系统,让他们能够认识到真正需要的是什么。对比现有的需求分析,可以提出更符合实际的要求。同时,软件设计、编码及测试人员的参与,更进一步加深与用户的沟通,理解用户的真实需求,有益于软件开发的各个环节的联系,保证目标系统的完整性与有效性。

小结

本章首先对系统分析做了基本介绍,详细介绍了问题定义、可行性分析和需求分析的过程。尤其对结构化需求分析方法进行了详细的讨论,包括需求的获取方法、利用实体关系图进行数据建模、利用数据流图进行功能建模和利用状态迁移图进行行为建模的方法。在讨论中给出了实际软件系统应用的案例,同时给出可行性分析报告和软件需求规格书的文档编写规范和主要内容。

习题

一、判断题

1. 需求规格书描述的是软件如何实现。
2. 在 E-R 图中,实体与实体之间的连接是通过主键和外键进行的。
3. 在结构化分析方法中,用以表达系统内数据的运行情况的工具是功能结构图。
4. 各种需求方法都有它们共同适用的方法。
5. 数据流图的基本成分有 6 种。

6. 软件需求的逻辑视图描述的是软件要达到的功能和要处理的信息之间的关系。
7. 软件需求的逻辑视图没有描述实现的细节。
8. 软件需求的物理视图给出的是处理功能和信息结构的实际表现形式。
9. 软件需求的物理视图需考虑实际的环境和具体的设备。
10. 数据流图的主图必须含有 4 种元素,缺一不可。
11. 数据流图的主图必须封闭在外部实体之间,实体可以有多个。
12. 数据流图中包含控制流。
13. 数据项是数据处理中基本的不可分割的逻辑单位。

二、选择题

1. 软件需求分析阶段的工作,可以分为以下 4 个方面:对问题的识别、分析与综合、编写需求分析文档以及()。
A. 总结
B. 阶段性报告
C. 需求分析评审
D. 以上答案都不正确
2. 各种需求方法都有它们共同适用的()。
A. 说明方法
B. 描述方式
C. 准则
D. 基本原则
3. 在结构化分析方法中,用以表达系统内数据的运行情况的工具有()。
A. 数据流图
B. 数据字典
C. 结构化英语
D. 判定表与判定树
4. 在结构化分析方法中用状态迁移图表达系统或对象的行为。在状态迁移图中,由一个状态和一个事件所决定的下一状态可能会有()。
A. 1 个
B. 2 个
C. 多个
D. 不确定
5. 软件需求分析的任务不应包括()。
A. 问题分析
B. 信息域分析
C. 结构化程序设计
D. 确定逻辑模型
6. 进行需求分析可使用多种工具,但()是不适用的。
A. 数据流图
B. 判定表
C. PAD
D. 数据字典
7. 在需求分析中,分析员要从用户那里解决的最重要的问题是()。
A. 要让软件做什么
B. 要给该软件提供哪些信息
C. 要求软件工作效率如何
D. 要让软件具有什么样的结构
8. 需求规格说明书的内容不应当包括()。
A. 对重要功能的描述
B. 对算法的详细过程性描述
C. 软件确认准则
D. 软件的性能
9. 需求规格说明书在软件开发中具有重要的作用,但其作用不应当包括()。
A. 软件设计的依据
B. 用户和开发人员对软件要“做什么”的共同理解
C. 软件验收的依据
D. 软件可行性分析的依据

三、填空题

1. 在实体-关系图中,表达对象的实例之间的关联有三种类型:一对一联系、(A)联系、多对多联系。
2. 需求分析的重点是(A)、(B)、(C)、(D)。
3. 获取需求的常用方法有(A)、(B)、(C)、(D)。
4. 数据流图的基本成分有(A)、(B)、(C)、(D)。
5. 在数据字典的每一个词条中应包含①(A)、②(B)、③(C)、④(D)、⑤(E)。

四、简答题

1. 可行性研究主要研究哪些问题? 试说明。
2. 需求获取为什么难?
3. 需求分析的原则有哪些?
4. 需求分析的任务有哪些?
5. 数据流图的作用是什么?
6. 数据字典的作用是什么?