

第3章

软件需求分析

一、概述

软件需求分析是软件生存周期中重要的一步,也是最关键的一步。只有通过软件需求分析,才能把软件功能和性能的总体概念描述为具体的软件需求规格说明,进而建立软件开发的基础。本章主要介绍需求分析的任务和步骤、需求分析方法和需求分析规格说明。

二、教学重点与难点

重点:

- (1) 需求分析的过程,能够用数据流图、E-R图、数据字典和简洁的算法描述所定义系统的逻辑模型。
- (2) 会应用结构化分析方法。
- (3) 关注需求规格说明书的编写。

难点:

数据流图、E-R图、数据字典的应用。

3.1 需求分析概述

软件需求是软件开发人员在开发软件产品之前应该做的事情。如果没有一个清晰和有效的用户需求,就不可能进行软件开发,即使进行开发,投入也是徒劳的。软件的功能必须与用户的需求一致。很多基于计算机的信息系统的失败都是由于不能正确获取用户的需求而造成的。如果修改一个已经完成的产品来满足用户最新的需求,所付出的努力、花费的代价都会相当高。

Standish Group的一项研究表明,导致软件开发失败的因素中有三条最重要的原因,它们所占失败因素比例超过了三分之一。分别是:

- (1) 缺少用户的输入:占软件失败因素 13%。
- (2) 不完整的需求和规格说明书:占软件失败因素 12%。
- (3) 需求和规格说明书的变更:占软件失败因素 12%。

Davis(1993)指出,如果开发周期内发现需求错误的时间太晚,花在弥补错误上的代价

会很大。图 3-1 指出了在软件开发周期中各个阶段发现需求错误时改正错误所花费的代价。该图表明,除非在软件开发早期发现需求错误,否则修正错误的代价几乎按指数级增加。这种现象表明,在需求分析阶段仔细确认用户的需求是非常重要的。

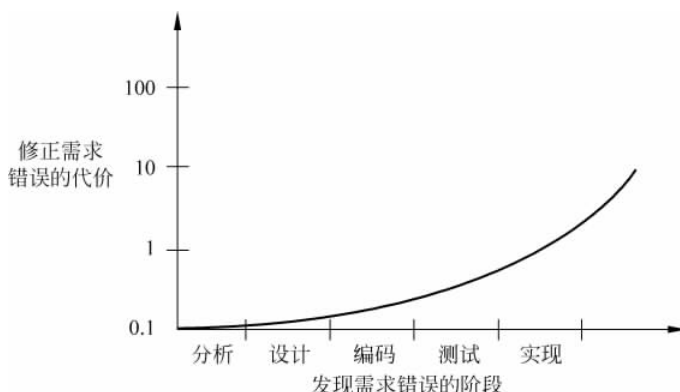


图 3-1 修正需求错误的代价

3.1.1 软件需求的定义

软件需求表达了对解决现实世界中某类问题的产品的要求和约束。

软件需求的目的是解决现实世界的问题,这是它最基本的特点。因此,软件需求是用来解决某个具体问题的。使用该软件的人可能会遇到包括公司或组织的业务流程问题,改正软件中存在的缺点,控制某种设备等问题。用户的功能、业务流程、设备等都是非常复杂的。因此,软件需求是一个非常复杂的需求组合,这些需求来自一个组织中不同层次的人,同时还要考虑软件的运行环境。

软件需求的一个重要特性就是它们是可验证的。验证某个软件的需求非常困难,代价很大。例如,开发呼叫中心模拟软件就必须验证吞吐量需求。软件需求和软件质检人员必须保证软件是可验证的。

需求除了表现出来的行为属性外,还有其他属性。常见的例子包括优先级,它使得资源有限的情况下保证开发的正常进行,使项目进展能被监测。一般的软件需求应该非常明确,以便于判断是否符合软件的要求,方便软件开发周期中的管理。

3.1.2 产品和过程需求

产品参数和过程参数有明显的区别。产品参数是待开发软件的要求。例如,软件可以验证一个学生在选一门课之前是否满足选课条件。

过程参数本质上是对软件开发的一个限制,例如,要求一个软件使用 Ada 语言编写。这有时被称为过程需求。许多软件都暗含过程需求。验证技术的选择就是一个例子。另一个例子是使用严谨的分析方法(如形式验证方法),它可以用来减少导致低可靠性的错误。过程需求可能是直接由开发组织、客户或者第三方(如安全管理者)提出的。

3.1.3 功能性和非功能性需求

功能需求用来描述系统应该做什么,即为用户和其他系统完成的功能、提供的服务。

例如,客户登录、邮箱网站的收发邮件、论坛网站的发帖留言等。

非功能性需求是指必须遵循的标准、外部界面的细节、实现的约束条件、质量属性等。非功能需求限制了选择解决问题方案的范围,如运行平台、实现技术、编程语言和工具等。

这些需求的例子有:

- (1) 硬件、软件和将遵照的通信接口。
- (2) 必须服从公司标准的用户界面。
- (3) 将被坚持的报告格式。
- (4) 过程限制,比如 ISO 9000 等。
- (5) 基础设施造成的硬件限制。

3.1.4 可量化的需求

应该尽可能清楚地陈述软件要求。要避免主观评断的、含糊的和不能验证的软件需求(软件应该是可靠的、软件应该是用户友好的),这对非功能性需求特别重要。两个可量化的需求如下:呼叫中心的软件必须增加 20% 的吞吐量;在任何营业时间出现一个致命错误的可能性应低于 1×10^{-8} 。吞吐量需求是一个非常高的水平,并且常常源自一些细节的需求。可靠性需求严格约束着系统的体系结构。

3.1.5 系统需求和软件需求

系统意味着为了完成某一目标而相互作用的元素的组合。这些元素包括硬件、软件、固件、人、信息、技术、设施、服务和其他系统工程、国际委员会定义的元素。

系统需求是系统的整体需求。在一个包含软件组件的系统中,软件需求来源于系统需求。在有些文献中也称系统需求为用户需求。系统需求包括用户需求、其他投资者的需求(如认证机构)和无法确定的人力资源的需求。

需求分析的任务不是确定系统如何完成工作,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。

(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 确定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。确定系统性能要求,如系统所需要的存储容量、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,系统在多长时间内有反应并输出结果,这对于实时系统来讲是关系到系统能够被用户接受的重要因素)等。

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、无歧义的书面的描述。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,用以验收软件产品。

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的信息域,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。在分析过程中可用层次的方式来细分这三个模型,以得出软件实现的具体细节。

3.2 需求分析的执行步骤

遵循科学的需求分析步骤可以使需求分析工作更高效。需求分析的一般步骤如图 3-2 所示。

1. 需求分析涉及的方面

需求分析涉及的方面有很多,主要有以下几方面:

(1) 在功能方面,需求包括系统要做什么,相对于原系统目标系统需要进行哪些修改,目标用户有哪些,以及不同用户需要通过系统完成何种操作等。

(2) 在性能方面,需求包括用户对于系统执行速度、响应时间、吞吐量和并发度等指标的要求。

(3) 在运行环境方面,需求包括目标系统对于网络设置、硬件设置、温度和湿度等周围环境的要求,以及操作系统、数据库和浏览器等软件配置的要求。

在界面方面,需求涉及数据的输入/输出格式的限制及方式、数据的存储介质和显示器的分辨率要求等问题。

2. 需求分析的一般步骤

(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。例如,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,如问卷调查、访谈、实地操作、建立原型系统等。

① 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



图 3-2 需求分析的一般步骤

的调查问卷进行汇总、统计和分析,开发人员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。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。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。

② 访谈通过开发人员与特定的用户代表进行座谈,进而了解用户的意见,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,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。

③ 为了深入地了解用户需求,有时候开发人员还会以用户的身份直接参与到现有系统的使用过程中,在亲身实践的基础上,更直接地体会现有系统的弊端,以及新系统应该解决的问题,这种需求获取方法就是实地操作。通过实地操作得到的信息会更加准确和真实,但是这种方法比较费时间。

④ 当用户本身对需求的了解不太清晰的时候,开发人员通常采用建立原型系统的方法,对用户需求进行挖掘。原型系统就是目标系统的一个可操作的模型。在初步获取需求后,开发人员会快速地开发一个原型系统。通过对原型系统进行模拟操作,开发人员能及时获得用户的意见,从而对需求进行明确的分析。利用原型系统获取需求的方法的示意图如图 3-3 所示。

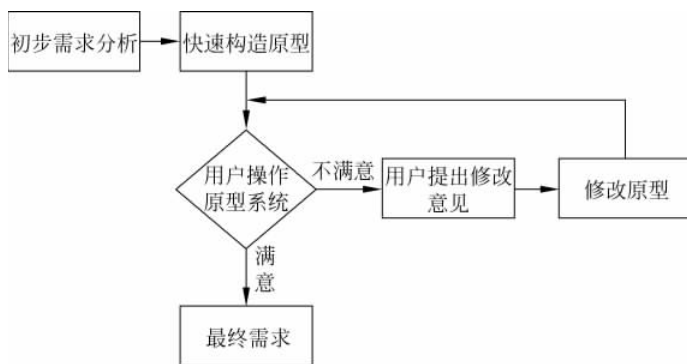


图 3-3 利用原型系统获取需求

(2) 分析需求,建立目标系统的逻辑模型。

在获得需求后,开发人员应该对问题进行分析抽象,并在此基础上从高层建立目标系统的逻辑模型。模型是对事物高层次的抽象,通常由一组符号和组织这些符号的规则组成常用的模型图有数据流图、E-R图、用例图和状态转换图等,不同的模型从不同的角度或不同的侧重点描述目标系统。绘制模型图的过程,既是开发人员进行逻辑思考的过程,也是开发人员进一步认识目标系统的过程。

(3) 将需求文档化。

获得需求后要将其描述出来,即将需求文档化。对于大型的软件系统,需求阶段一般会输出三个文档:

- ① 系统定义文档(用户需求报告)。
- ② 系统需求文档(系统需求规格说明书)。
- ③ 软件需求文档(软件需求规格说明书)。

对于简单的软件系统而言,需求阶段只需要输出软件需求文档(即软件需求规格说明书)即可。软件需求规格说明书主要描述软件的需求,从开发人员的角度对目标系统的业务模型、功能模型和数据模型等内容进行描述。作为后续的软件设计和测试的重要依据。需求阶段的输出文档应该具有清晰性、无二义性和准确性,并且能够全面和确切地描述用户需求。

(4) 需求验证。

需求验证是对需求分析的成果进行评估和验证的过程。为了确保需求分析的正确性、一致性、完整性和有效性,提高软件开发的效率,为后续的软件开发做好准备,需求验证的工作非常必要。

在需求验证的过程中,可以对需求阶段的输出文档进行多种检查,如一致性检查、完整性检查和有效性检查等。同时,需求评审也是在这个阶段进行的。

3. 需求分析的原则

需求分析的原则如下:

(1) 必须能表达和理解问题的数据域和功能域。其中,数据域包括数据流、数据内容和数据结构。

(2) 自顶向下逐层分解问题。

(3) 要给出系统的逻辑视图和物理视图。

① 逻辑视图给出软件要达到的功能和要处理数据之间的关系,而不是实现的细节。

② 物理视图给出处理功能和数据结构的实际表示形式。

3.3 结构化分析方法

结构化分析方法(Structured Method,结构化方法)是强调开发方法的结构合理性以及所开发软件的结构合理性的软件开发方法。

结构是指系统内各个组成要素之间的相互联系、相互作用的框架。结构化分析方法提出了一组提高软件结构合理性的准则,如分解与抽象、模块独立性、信息隐蔽等。针对软件生存周期各个不同的阶段,有结构化分析(SA)和结构化程序设计(SP)等方法。

结构化分析方法给出一组帮助系统分析人员产生功能规约的原理与技术。它一般利用图形表达用户需求,使用的手段主要有数据流图、数据字典、结构化语言、判定表以及判定树等。

结构化分析的步骤如下:

(1) 分析当前的情况,做出反映当前物理模型的 DFD。

(2) 推导出等价的逻辑模型的 DFD。

(3) 设计新的逻辑系统,生成数据字典和基元描述。

- (4) 建立人机接口,提出可供选择的目標系统物理模型的 DFD。
- (5) 确定各种方案的成本和风险等级,据此对各种方案进行分析。
- (6) 选择一种方案。
- (7) 建立完整的需求规约。

3.3.1 数据流图

数据流图(Data Flow Diagram,DFD)从数据传递和加工角度,以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程,是结构化分析方法的主要表达工具及用于表示软件模型的一种图示方法。

数据流图是结构化分析方法中使用的工具,它以图形的方式描绘数据在系统中流动和处理的过程。由于它只反映系统必须完成的逻辑功能,所以它是一种功能模型。在结构化分析方法中,数据流图是需求分析阶段产生的结果。数据流图从数据传递和加工的角度,以图形的方式刻画数据流从输入到输出的移动变换过程。

数据流程图中有以下几种主要元素:

(1) →: 数据流。数据流是数据在系统内传播的路径,由一组成分固定的数据组成。如订票单由旅客姓名、年龄、单位、身份证号、日期、目的地等数据项组成。由于数据流是流动中的数据,所以必须有流向,除了与数据存储之间的数据流不用命名外,数据流应该用名词或名词短语命名。

(2) □: 数据源或宿(“宿”表示数据的终点)。代表系统之外的实体,可以是人、物或其他软件系统。

(3) ○: 对数据的加工(处理)。加工是对数据进行处理单元,它接收一定的数据输入,对其进行处理,并产生输出。

(4) =: 数据存储。表示信息的静态存储,可以代表文件、文件的一部分、数据库的元素等。

数据流图基本组成符号如图 3-4 所示。

在单张数据流图时,必须注意以下原则:

(1) 一个加工的输出数据流不应与输入数据流同名,即使它们的组成成分相同。

(2) 保持数据守恒。也就是说,一个加工所有输出数据流中的数据必须能从该加工的输入数据流中直接获得,或者说通过该加工能产生的数据。

(3) 每个加工必须既有输入数据流,又有输出数据流。

(4) 所有的数据流必须以一个外部实体开始,并以一个外部实体结束。

(5) 外部实体之间不应该存在数据流。

例如,学生信息管理业务活动数据流程图如图 3-5 所示。

分层数据流图的设计方法如下:

(1) 画出系统的输入/输出,即先画顶层数据流图。顶层流图只包含一个加工,用以表示被开发的系统,然后考虑该系统有哪些输入数据流、输出数据流。顶层图的作用在于表明

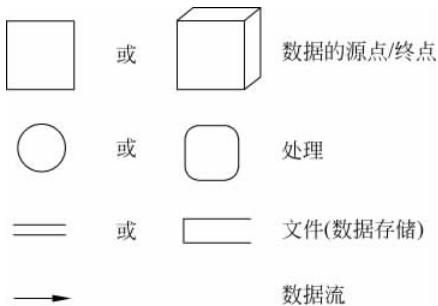


图 3-4 数据流图的基本组成符号

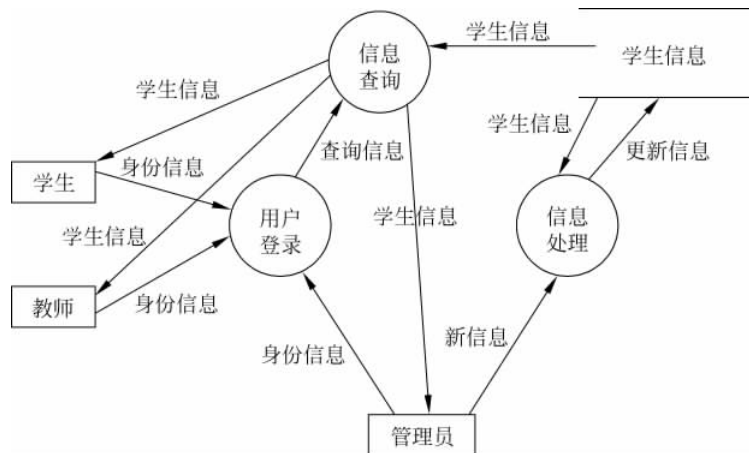


图 3-5 学生信息管理业务活动数据流程图

被开发系统的范围以及它和周围环境的数据交换关系。

(2) 画出系统内部,即画下层数据流图。不再分解的加工称为基本加工。一般将层号从 0 开始编号,采用自顶向下、由外向内的原则。画 0 层数据流图时,分解顶层流图的系统为若干子系统,决定每个子系统间的数据接口和活动关系。

可以用下述方法来确定加工:

在数据流的组成或值发生变化的地方应该画出一个加工,这个加工的功能就是实现这一变化,也可以根据系统的功能决定加工。

确定数据流的方法:用户把若干数据当作一个单位来处理(这些数据一起到达、一起处理)时,可以把这些数据看成一个数据流。

关于数据存储:对于一些以后某个时间要使用的数据,可以组织成为一个数据存储来表示。

(3) 画出加工的内部,把每个加工看作一个小系统,把加工的输入/输出数据流看成小系统的输入/输出流。于是可以像画 0 层图一样画出每个小系统的加工的数据流图。

(4) 画子加工的分解图,对第(3)步分解出来的数据流图中的每个加工,重复第(3)步的分解过程,直到图中尚未分解的加工都是足够简单的(即不可再分解)。至此,得到一套分层数据流图。

(5) 对数据流图和加工编号,对于一个软件系统,其数据流图可能有许多层,每一层又有许多张图。为了区分不同的加工和不同的数据流图子图,应该对每张图进行编号,以便于管理。

- ① 顶层图只有一张,图中的加工也只有一个,所以不必为其编号。
- ② 0 层图只有一张,图中的加工号分别是 0.1、0.2……或者 1、2……
- ③ 子图就是父图中被分解的加工号。
- ④ 子图中的加工号由图号、圆点和序号组成,如 1.12、1.3 等。

(6) 注意事项。

- ① 命名。不论数据流、数据存储还是加工,恰当的命名都可以使人们易于理解其含义。
- ② 画数据流而不是控制流。数据流反映系统“做什么”,不反映“如何做”,因此,箭头上

的数据流名称只能是名词或名词短语,整个图中不反映加工的执行顺序。

③ 一般不画物质流。数据流反映能用计算机处理的数据,并不是实物,因此对目标系统的数据流图一般不画物质流。

④ 每个加工至少有一个输入数据流和一个输出数据流,反映出此加工数据的来源与加工的结果。

⑤ 编号。如果一张数据流图中的某个加工分解成另一张数据流图时,则上层图为父图,直接下层图为子图。子图及其所有的加工都应编号。

⑥ 父图与子图的平衡。子图的输入/输出数据流同父图相应加工的输入/输出数据流必须一致,此即父图与子图的平衡。

⑦ 局部数据存储。如果某层数据流图中的数据存储不是父图中相应加工的外部接口,而只是本图中某些加工之间的数据接口,则称这些数据存储为局部数据存储。

⑧ 提高数据流图的易懂性。注意合理分解,要把一个加工分解成几个功能相对独立的子加工,这样可以减少加工之间输入/输出数据流的数目,增加数据流图的可理解性。

【实例 3-1】 营销系统的数据流图。分析一家公司的营销系统。其采购部门每天需要按销售部门提供的订货单(需订的货物)向供应商采购货物。每种货物的数量都存放在数据存储货物库存中,销售和采购使每种货物数量发生的变化能够在此数据存储中及时被反映出来。而资金的汇总、核对等工作由其会计部门处理。这样此系统的顶层结构就大致可以分析出来。绘出系统的第一层数据流图,如图 3-6 所示。

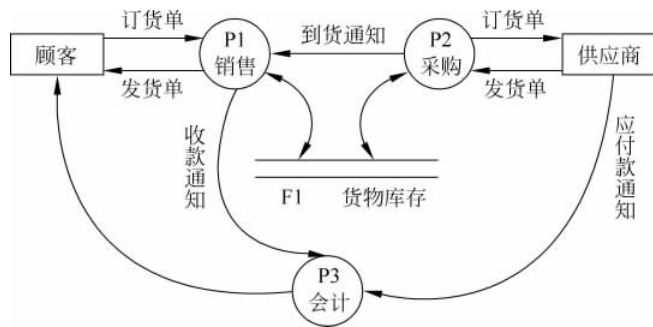


图 3-6 第一层数据流图

为了更加清晰地描述系统,可把顶层数据流图的三个主要加工销售、采购、会计再进行逐步分解,形成第二层数据流图。

首先分析销售加工。先根据顾客的订货单和货物目录确定订货,在这期间要修改和维护货物目录和顾客两个数据存储。对于正当的订货,目前有货可发的则直接产生发货单准备发货;而如果暂时缺货则产生暂存订货单,等采购到所需的货物再产生发货单。按顾客要求发货后,要修改货物库存、销售历史和应收款账目这三个数据存储。对于库存和销售历史的变化要分别编写库存检索和库存销售报表提供给经理。其数据流图如图 3-7 所示。

再来分析采购加工。首先根据暂存订货单和货物的库存情况确定需订的货物,产生需订货物的数据存储。对需订货物按供应商汇总,产生采购订货单,分别发给不同的供应商。

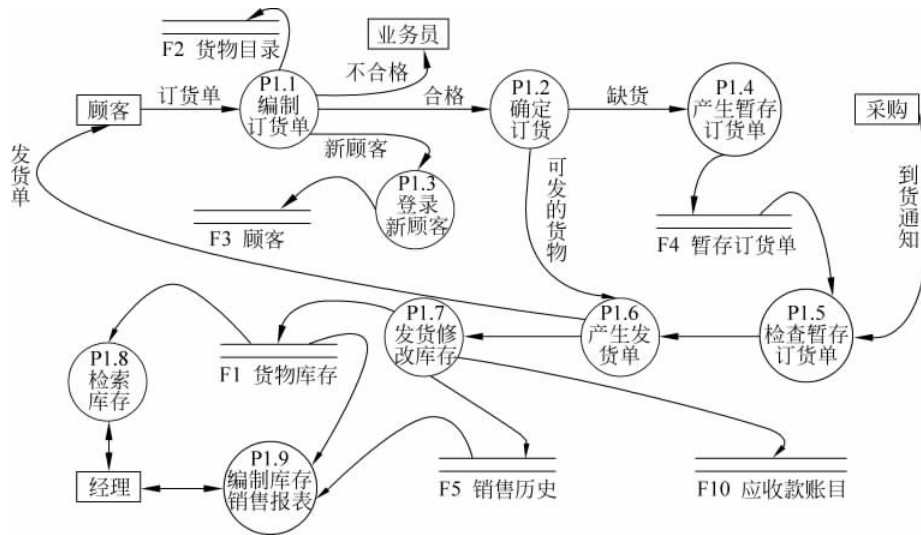


图 3-7 销售系统数据流图

如果供应商发出的发货单正确无误,提货后向销售部门发出到货通知并修改库存、应付款账目和需订货物数量。这样为以后的订货提供依据。其数据流图如图 3-8 所示。

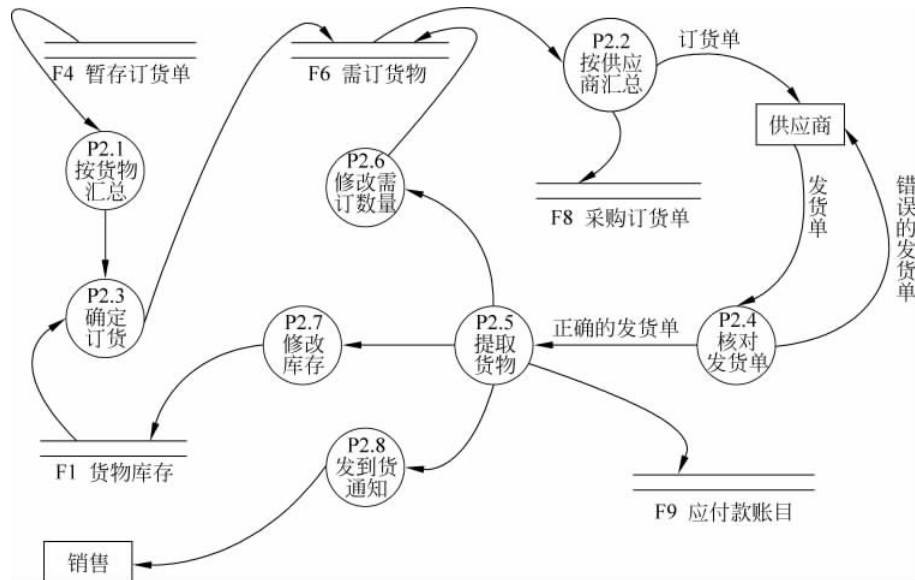


图 3-8 采购系统数据流图

最后分析会计加工。顾客付款后应得到收据,收款处理还应该修改数据存储:应收款账目。而对供应商的应付款通知进行核对后要进行付款处理。这个操作要修改另一数据存储:应付款账目。这时要根据应收款账目和应付款账目的修改情况进行修改总账目的处理,最后把总账目的变化情况编制成会计报表提交给经理。其数据流图如图 3-9 所示。

以上两层四张数据流图一起组成了这家公司营销系统的分层数据流图。第二层的加工比第一层要细,并且大都为足够简单的“基本加工”,不必再进行分解。

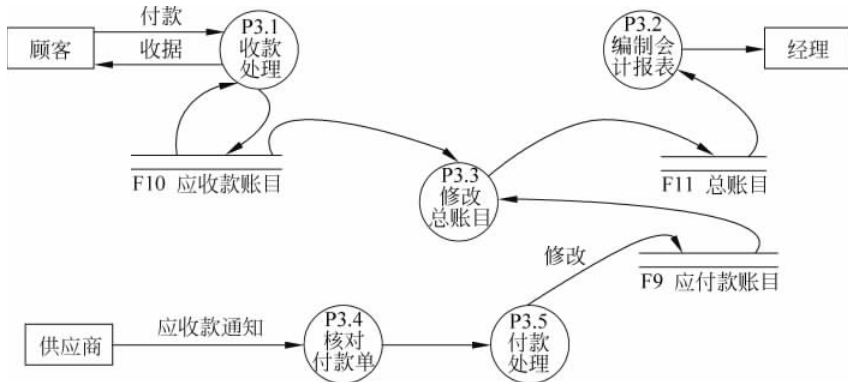


图 3-9 会计系统数据流图

3.3.2 数据字典

数据字典(Data Dictionary)是对于数据模型中的数据对象或者项目的描述的集合,这样做有利于程序员和其他需要参考的人。分析一个用户交换的对象系统的第一步就是辨别每一个对象,以及其与其他对象之间的关系。这个过程称为数据建模,结果产生一个对象关系图。当每个数据对象和项目都给出一个描述性的名字之后,对它的关系再进行描述(或者是成为潜在描述关系的结构中的一部分),然后描述数据的类型(如文本还是图像,或者是二进制数值),列出所有可能预先定义的数值,以及提供简单的文字性描述。这个集合被组织成书的形式用来参考,称为数据字典。

当开发用到数据模型的程序时,数据字典可以帮助理解数据项适合结构中的哪个地方,它可能包含什么数值,以及数据项基本上表示现实世界中的什么意思。例如,一家银行或者是一个银行组织可能对客户银行业涉及的数据对象进行建模。它们需要给银行程序员提供数据字典。这个数据字典就描述了客户银行业中的数据模型每一个数据项(例如,“账户持有人”和“可用信用”)。

数据字典最重要的作用是作为分析阶段的工具。任何字典最重要的用途都是供人查询对不了解的条目的解释,在结构化分析中,数据字典的作用是给数据流图上每个成分加以定义和说明。换句话说,数据流图上所有的成分的定义和解释的文字集合就是数据字典,而且在数据字典中建立的一组严密一致的定义很有助于改进分析员和用户的通信。

数据库数据字典不仅是每个数据库的中心,而且对每个用户而言也是非常重要的信息。用户可以用 SQL 语句访问数据库数据字典。

数据字典的组成:

(1) 数据项: 数据流图中数据块的数据结构中的数据项说明。

数据项是不可再分的数据单位。对数据项的描述通常包括以下内容:

数据项描述 = {数据项名, 含义说明, 别名, 数据类型, 长度, 取值范围, 取值含义, 与其他数据项的逻辑关系}

其中,“取值范围”“与其他数据项的逻辑关系”定义了数据的完整性约束条件,是设计数

据检验功能的依据。若干数据项可以组成一个数据结构。

(2) 数据结构：数据流图中数据块的数据结构说明。

数据结构反映了数据之间的组合关系。一个数据结构可以由若干数据项组成,也可以由若干数据结构组成,或由若干数据项和数据结构混合组成。对数据结构的描述通常包括以下内容:

数据结构描述 = {数据结构名, 含义说明, 组成: {数据项或数据结构}}

(3) 数据流：数据流图中流线的说明。

数据流是数据结构在系统内传输的路径。对数据流的描述通常包括以下内容:

数据流描述 = {数据流名, 含义说明, 数据流来源, 数据流去向, 组成: {数据结构}, 平均流量, 高峰期流量}

其中,“数据流来源”是说明该数据流来自哪个过程,即数据的来源。“数据流去向”是说明该数据流将到哪个过程去,即数据的去向。“平均流量”是指在单位时间(每天、每周、每月等)里的传输次数。“高峰期流量”是指在高峰时期的数据流量。

(4) 数据存储：数据流图中数据块的存储特性说明。

数据存储是数据结构停留或保存的地方,也是数据流的来源和去向之一。对数据存储的描述通常包括以下内容:

数据存储描述 = {数据存储名, 含义说明, 编号, 流入的数据流, 流出的数据流, 组成: {数据结构}, 数据量, 存取方式}

其中,“数据量”是指每次存取多少数据,每天(或每小时、每周等)存取几次等信息。“存取方法”包括是批处理,还是联机处理;是检索还是更新;是顺序检索还是随机检索等。另外,“流入的数据流”要指出其来源,“流出的数据流”要指出其去向。

(5) 处理过程：数据流图中功能块的说明。

数据字典中只需要描述处理过程的说明性信息,通常包括以下内容:

处理过程描述 = {处理过程名, 含义说明, 输入: {数据流}, 输出: {数据流}, 处理: {简要说明}}

其中,“简要说明”中主要说明该处理过程的功能及处理要求。功能是指该处理过程用来做什么(而不是怎么做);处理要求包括处理频度要求,如单位时间里处理多少事务、多少数据量、响应时间要求等,这些处理要求是后面物理设计的输入及性能评价的标准。

【实例 3-2】“进销存管理系统”数据字典。

(1) 数据项,如表 3-1 所示。

表 3-1 数据项

| 数据项名 | 含义说明 | 别名 | 数据类型 | 取值范围 | 取值含义 |
|------|-----------|-----|----------|-----------------------|--|
| Sph | 唯一标识每一个商品 | 商品号 | Char(15) | 0~10 ¹⁴ -1 | 前三位是厂商所在国家的国际代码,它和 4~7 位一起构成厂商识别代码即厂商的注册号,8~12 位是商品项目代码,最后一位是校验码 |

续表

| 数据项名 | 含义说明 | 别名 | 数据类型 | 取值范围 | 取值含义 |
|-------|------------|-------|---------------|------|--|
| Spmc | 标识商品的名称 | 商品名称 | Char(30) | | |
| Jj | 标识商品的进价 | 进价 | Money | | |
| Gysbh | 唯一标识商品的供应商 | 供应商编号 | Char(8) | | 前三位是厂商所在国家的国际代码,它和4~7位一起构成厂商识别代码,即厂商的注册号 |
| kcsl | 标识库存的数量 | 库存数量 | Char(8) | | |
| Gysmc | | 供应商名 | Char(30) | | |
| Dz | | 地址 | Char(30) | | |
| Dh | | 联系方式 | Char(15) | | |
| Lxr | | 联系人 | Char(16) | | |
| Ygbh | | 员工编号 | Char(8) | | 前两位是部门号,3~10是加入日期,11~13是顺序编号 |
| Ygmc | | 姓名 | Char(8) | | |
| Zw | | 职务 | Char(8) | | |
| Bmbh | | 部门编号 | Char(4) | | 顺序编号 |
| Dhdbb | | 订货单编号 | Char(8) | | |
| Dgrq | | 订购日期 | Smalldatetime | | |
| Jhq | | 交货期 | Smalldatetime | | |
| Rkdh | | 入库单号 | Char(8) | | |
| Rksl | | 入库数量 | Char(8) | | |
| Rkri | | 入库日期 | Smalldatetime | | |
| Ckdh | | 出库单号 | Char(8) | | |
| cksl | | 出库数量 | Char(8) | | |
| Ckrq | | 出库日期 | Smalldatetime | | |
| Gkbh | | 顾客编号 | Char(10) | | |
| Sgsp | | 所购商品 | Char(60) | | |
| Zj | | 总价 | Numeric(5,1) | | |
| Rq | | 日期 | Smalldatetime | | |

(2) 数据结构,如表 3-2 所示。

表 3-2 数据结构

| 数据结构名 | 含义说明 | 组成 |
|-------|------------------------------|-------------------------|
| 商品 | 是商品管理子系统的主体数据结构,定义一个商品的有关信息 | 商品编号,商品名称,供应商,单价 |
| 供应商 | 是进货管理子系统的主体数据结构,定义一个供应商的有关信息 | 供应商编号,供应商名称,联系方式,地址,联系人 |
| 员工 | 是员工子系统的主体数据,定义员工的有关信息 | 员工编号,姓名,年龄,职务,所属部门 |
| 顾客 | 是销售子系统的主体数据,定义顾客的有关信息 | 编号,所购商品 |

续表

| 数据结构名 | 含义说明 | 组成 |
|-------|-------------|--------------------------------|
| 订货单 | 是订货子系统的主体数据 | 订货单编号, 供应商编号, 采购员工号, 订购日期, 交货期 |
| 入库单 | 是入库子系统的主体数据 | 入库单号, 商品编号, 入库数量, 入库日期 |
| 出库单 | 是出库子系统的主体数据 | 出库单号, 商品编号, 出库数量, 出库日期 |
| 销售单 | 是销售子系统的主体数据 | 顾客编号, 商品编号, 总价, 日期 |

(3) 数据流, 如表 3-3 所示。

表 3-3 数据流

| 数据流名 | 含义说明 | 数据流来源 | 数据流去向 | 组成 | 平均流量 | 高峰期流量 |
|------|----------|---------|-------|-------------------------------|---------|----------|
| 入库单 | 供应商供应的货物 | 供应商供货处理 | 入库单存储 | 入库单号, 商品编号, 供应商编号, 书库数量, 入库日期 | 每天 20 个 | 每天 100 个 |
| 出库单 | 库中商品出库 | 出库处理 | 出库单存储 | 出库单号, 商品编号, 出库数量, 出库日期 | 每天 20 个 | 每天 100 个 |
| 销售单 | 商店中商品出售 | 销售处理 | 销售单存储 | 顾客编号, 商品编号, 总价, 日期 | 每天 20 个 | 每天 100 个 |

(4) 数据存储, 如表 3-4 所示。

表 3-4 数据存储

| 数据存储名 | 含义说明 | 流入的数据流 | 流出的数据流 | 组成 | 数据量 | 存取方式 |
|-------|------|------------|--------|-----|---------------|------|
| 入库 | 商品入库 | 入库单 | 入库单 | 入库单 | 1 000 000 个记录 | 随机存取 |
| 出库 | 商品出库 | 商品信息, 出库信息 | 出库单 | 出库单 | 1 000 000 个记录 | 随机存取 |
| 销售 | 商品销售 | 商品信息, 销售信息 | 销售单 | 销售单 | 1 000 000 个记录 | 随机存取 |

(5) 处理过程, 如表 3-5 所示。

表 3-5 处理过程

| 处理过程名 | 含义说明 | 输入数据流 | 输出数据流 | 处理 |
|-------|----------|-------|-------|--------------------------|
| 入库 | 商品存入仓库 | 入库单 | 入库单 | 记录入库单号, 商品编号, 入库数量, 入库日期 |
| 出库 | 商品从仓库中取出 | 出库单 | 出库单 | 记录出库单号, 商品编号, 出库数量, 出库日期 |
| 销售 | 商品从商店中售出 | 销售单 | 销售单 | 记录顾客编号, 商品编号, 总价, 日期 |

3.3.3 实体-关系图

实体-关系图即 E-R 图(Entity Relationship Diagram), 是指提供了表示实体型、属性和联系的方法, 用来描述现实世界的概念模型。

E-R 方法是“实体-关系方法”(Entity-Relationship Approach)的简称。它是描述现实世界概念结构模型的有效方法。

实体关系模型或实体联系模式图(ERD)由美籍华裔计算机科学家陈品山(Peter Chen)发明, 是概念数据模型的高层描述所使用的数据库模型或模式图, 它为表述这种实体关系模式

图形式的数据模型提供了图形符号。这种数据模型典型的用在信息系统设计的第一阶段,比如它们在需求分析阶段用来描述信息需求和/或要存储在数据库中的信息的类型。但是,数据建模技术可以用来描述特定论域(即感兴趣的区域)的任何本体(即对使用的术语和它们的联系的概述和分类)。在基于数据库的信息系统设计的情况下,在后面的阶段(通常称为逻辑设计),概念模型要映射到逻辑模型(如关系模型)上,且要在物理设计期间映射到物理模型上。注意,有时这两个阶段被一起称为“物理设计”。

通常,使用 E-R 图来建立数据模型,可把用 E-R 图描绘的数据模型称为 E-R 模型。E-R 图中包含了实体(即数据对象)、关系和属性三种基本成分,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。

人们通常就是用实体、联系和属性这三个概念来理解现实问题的,因此,E-R 模型比较接近人的习惯思维方式。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。

构成 E-R 图的基本要素是实体、属性和联系,其表示方法为:

(1) 实体(Entity): 具有相同属性的实体具有相同的特征和性质,用实体名及其属性名集合来抽象和刻画同类实体。在 E-R 图中用矩形表示实体,矩形框内写明实体名。比如,学生张三、学生李四都是实体。如果是弱实体,则在矩形外面再套实线矩形。

(2) 属性(Attribute): 实体所具有的某一特性,一个实体可由若干属性来刻画。在 E-R 图中用椭圆形表示属性,并用无向边将其与相应的实体连接起来,如图 3-10 所示。比如,学生的姓名、学号、性别都是属性。如果是多值属性,则在椭圆形外面再套实线椭圆。如果是派生属性,则用虚线椭圆表示。

(3) 联系(Relationship): 联系也称关系,信息世界中反映实体内部或实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系;实体之间的联系通常是指不同实体集之间的联系。在 E-R 图中用菱形表示联系,菱形框内写明联系名,并用无向边分别与有关实体连接起来,同时在无向边旁标上联系的类型(1:1、1:n 或 m:n),如图 3-11~图 3-13 所示。比如,老师给学生授课存在授课关系,学生选课存在选课关系。如果是弱实体的联系,则在菱形外面再套菱形。



图 3-10 属性

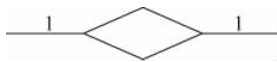


图 3-11 1:1 关系

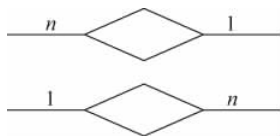


图 3-12 一对多或多对一关系



图 3-13 多对多关系

作图步骤:

(1) 确定所有的实体集合。

- (2) 选择实体集应包含的属性。
- (3) 确定实体集之间的联系。
- (4) 确定实体集的关键字,用下划线在属性上表明关键字的属性组合。
- (5) 确定联系的类型,在用线将表示联系的菱形框联系到实体集时,在线旁注明是 1 或 n (多)来表示联系的类型。

【实例 3-3】

(1) 两个实体集之间的一对一的联系的绘制方法。

假设某学院有若干班级,每个班级只有一个辅导员,每个辅导员只管理一个班级,则辅导员和班级之间是一对一的联系。班级和辅导员的属性分别如下:

辅导员——工号,姓名,性别,年龄,学历

班级——班号,班级名

另外,辅导员和班级之间是管理联系,这种管理用一个属性任职时间来描述,根据描述画出能够反映实体集之间的信息关系的 E-R 图,如图 3-14 所示。

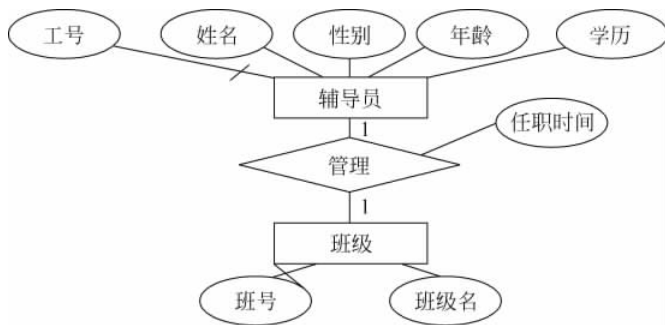


图 3-14 辅导员和班级 E-R 图

(2) 两个实体集之间的一对多的联系的绘制方法。

假设在某出版管理系统中,有两个实体集:出版社和图书。出版社用来出版图书,且规定一本图书只能由一个出版社出版,一个出版社可以出版多本图书。出版社和图书之间是一对多的联系。出版社和图书的属性分别如下:

出版社——社号,地点,电话

图书——书号,书名,价格

另外,在出版联系中要反映出出版图书的数量。根据描述画出能够反映实体集之间的信息关系的 E-R 图,如图 3-15 所示。

(3) 两个实体集之间的多对多的联系的绘制方法。

假设在某图书管理系统中,一名作者可以创作多本书,一本书也可以由多名作者共同创作。图书和作者之间是多对多的联系。作者和图书可用以下属性来描述:

作者——编号,姓名,职称

图书——书号,书名,出版日期

另外,在“创作”联系中应能反映出该书的合同时间和完成时间。根据描述画出能够反映实体之间的信息关系的 E-R 图,如图 3-16 所示。

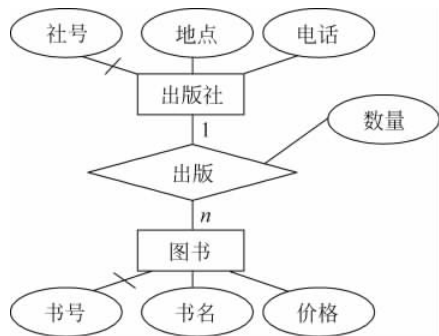


图 3-15 出版社和图书 E-R 图

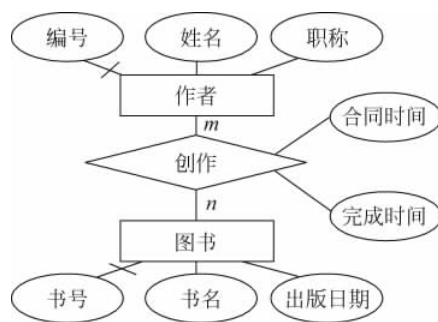


图 3-16 作者和图书 E-R 图

3.3.4 层次方框图

层次方框图用树形结构的一系列多层次的矩形描述数据的层次结构。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,下面的各层矩形框代表这个数据的子集,最底层的各个框代表组成这个数据的实际数据元素(不能再分割的元素)。

例如,某计算机公司全部产品的数据结构如图 3-17 所示。这家公司的产品由硬件、软件和服务三类产品组成,软件产品又分为系统软件和应用软件,系统软件又进一步分为操作系统、编译程序和软件工具等。

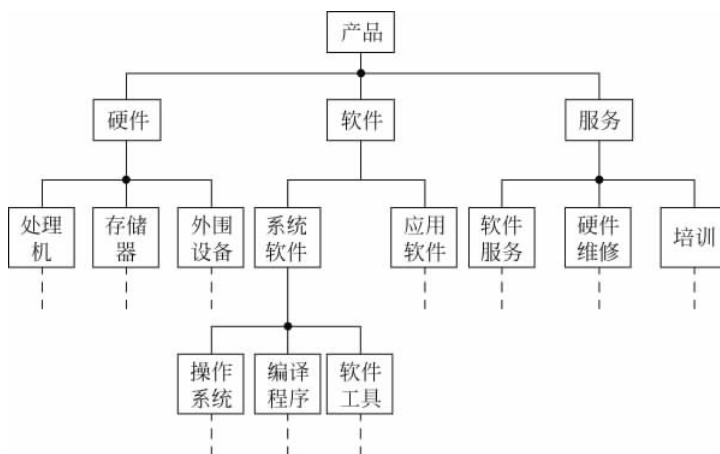


图 3-17 某计算机公司全部产品的数据结构

随着结构的精细化,层次方框图对数据结构也描绘得越来越详细,这种模式非常适于需求分析阶段的需要。系统分析员从对顶层信息的分类开始,沿图中每条路径反复细化,直到确定数据结构的全部细节为止。

3.3.5 Warnier 图

法国计算机科学家 Warnier 提出了表示信息层次结构的另外一种图形工具。与层次方框图相类似,Warnier 图也用树形结构描绘信息,但是这种图形工具比层次方框图提供了更

为详细的描绘手段。

用 Warnier 图可以表明信息的逻辑组织,它不仅可以指出一类信息或一个信息量是重复出现的,也可以表示特定信息在某一类信息中是有条件出现的。因为重复和条件约束是说明软件处理的基础,所以 Warnier 图成为了软件设计的工具。

图 3-18 是用 Warnier 图描绘软件产品的一个例子,说明了这种图形工具的法。图中的花括号用来区分数据结构的层次,在一个花括号中的所有名字都属于同一类信息,例如,操作系统、编译程序和软件工具都在系统软件花括号中,都属于系统软件类,而编辑程序、测试驱动程序和设计辅助工具都在软件工具花括号中,都属于软件工具类;异或信息 \oplus 表明一类信息或者一个数据元素在一定条件下才出现,而且在这个符号上、下方的两个名字所代表的数据只能出现一个,例如,一个软件产品可以是系统软件或应用软件,不可能即是系统软件又是应用软件;在一个名字下面(或右边)的括号中的数字(P1、P2、P3、P4 和 P5 表明了这个名字所代表的信息类(或元素)在这个数据结构中出现的次数)。



图 3-18 软件产品的 Warnier 图

3.3.6 IPO 图

IPO 图是输入/处理/输出图的简称,它是美国 IBM 公司提出的一种图形工具,能够方便地描绘输入数据、处理数据和输出数据的关系。

IPO 图使用的基本符号少而简单,因此很容易掌握。它的基本形式是在左边的框中列出有关的输入数据,在中间的框中列出主要的处理,在右边的框中列出产生的输出数据。处理框中列出了处理的顺序,但是用这些基本符号还不足以精确描述执行处理的详细情况。图 3-19 是一个主文件更新的 IPO 图。

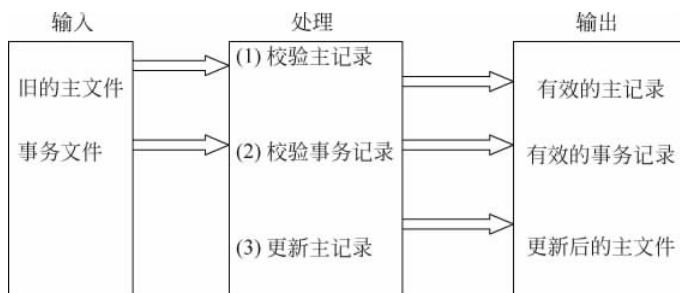


图 3-19 主文件更新的 IPO 图

3.4 软件需求规格说明书

软件需求规格说明书(Software Requirement Specification, SRS)是软件系统的需求规格说明,是开发系统的行为的完整描述,它包含了描述用户与软件交互的用例的集合。用例也称功能性需求。除了用例之外,SRS还包含了非功能的(或补充性的)需求。非功能性需求是指为设计和实现增加限制条件的需求,如性能需求、质量标准或设计上的限制。

规格说明书是对规模、计划、材料和其他需求的详细描述。在软件工程中,它指明了服务的消费者和生产者之间的协议,或者是用户和实现者之间的协议。因此,它可以是需求规格说明书(用户和开发者之间的协议)、设计规格说明书(设计者和实现者之间的协议)、模块规格说明书(写详细设计的设计者和编程者之间的协议)。

1. 软件需求规格说明书的特征

软件需求规格说明书记录用户需求。它的功能如下:

- (1) 标准化开发者的概念,并且简洁有效地表达出来。
- (2) 将对需求的理解传达给赞助商并获得验证。
- (3) 定义一个软件实现和维护的基线。

软件需求规格说明书应具有如下特征:

- (1) 应该只是有固定的、已知的并同意交付的系统的基本特征。
- (2) 应该包含交付什么,而不是如何交付。因而实施细节只能在设计阶段开始着手。
- (3) 应该使用客户能理解的词汇。
- (4) 应该是正确的。例如,它可以说,系统在一个小时内可以处理 50 000 份文档,而实际上它可能无法处理超过 20 000 份文档,这就是一个不正确的案例。
- (5) 应该是准确的。例如,“可以处理大量文档”或者“处理一个文档的时间很短”,这就是不准确的。
- (6) 应该是明确的,也就是说一个语句应该只能表达一个意思。缺乏书面表达能力会导致语句的含义模糊。使用正式规格可以使语句的表达更明确,但这也使得语句变得不容易理解。

下面举一个需求规格的例子:

“每当系主任邀请外部的人时,都要向宾馆经理发送请求,这样的请求必须得到研究所所长的批准。”

第一句赋予了系主任权威;第二句增加了限制条件,但是它并没有说是否所长的批准需要伴随着系主任的邀请出现。因此可能有两种解释:

- ① 如果所长批准则忽略系主任的邀请。
- ② 基于邀请产生一个请求,之后根据所长是否批准进行确认/取消。
- (7) 应该是完整的。语句“如果交易是‘买断型’的则应更新数据库”是不完整的,它必须表明如果交易不是“买断型的”应给采取的行动的类型。
- (8) 应该是可验证的。一旦系统被设计和实施,应该可以验证系统设计/实施是否满足

原始需求(使用分析或正式的方法)。

(9) 应该是能确认的。用户应该能够阅读/理解需求规格说明书并表明说明书在何种程度上反映了他/她的想法。

(10) 应该是一致的。例如,需求规格中某一个地方的语句说:如果库存是负数,则将出现错误信息,并且交易不被处理;另外一个地方却说:即使交易会使得库存量变为负数,但只要交易能使库存达到预期水平,那么交易也应该被执行。这样就是不一致的。

(11) 应该是可更改的。软件需求规划说明书的结构和风格应该保证任何需求的更改都是容易、完整和一致的。因此,它需要大量的清晰准确的内容表、交叉引用、索引和词汇表。

(12) 应该是可以追踪的。需求应该允许设计/实现和需求之间是相互联系的。

2. 软件需求规格说明书的内容

软件需求规格书应该包括如下内容:

1 引言

1.1 编写目的

说明编写这份软件需求说明书的目的,指出预期的读者。

1.2 项目背景

说明:

(1) 待开发的软件系统的名称。

(2) 本项目的任务提出者、开发者、用户及实现该软件的计算中心或计算机网络。

(3) 该软件系统同其他系统或其他机构的基本的相互来往关系。

1.3 定义

列出本文件中用到的专门术语的定义和缩略语的全称。

1.4 参考资料

列出需要的参考资料,如本项目的经核准的计划任务书或合同、上级机关的批文。

2 任务概述

2.1 目标

2.2 运行环境

2.3 条件与限制

3 系统数据描述

3.1 业务流程图描述

3.2 系统的数据需求

3.3 系统流程图

3.4 系统数据字典

3.5 系统的逻辑方案

4 系统的功能要求

4.1 功能划分

4.2 功能描述

5 系统的性能需求

系统处理的准确性和及时性。

6 系统的运行要求

3.5 需求规格案例分析

以“机票预订系统”项目为例说明需求分析的主要过程和要求。

1 引言

1.1 编写目的

开发机票预订系统软件,能够适应现今社会并提高生产效率。使售票员和旅行社更加变得方便和有效率。该系统软件非常容易被接受,具有简单易学性、便于管理等功能。

1.2 项目背景

随着社会发展的不断进步和航空事业的壮大,以及人们消费水平逐渐提高,乘坐民航的消费者越来越多,机票预订系统也开始影响着人们日常生活和出行,并且变得越来越重要。而原有的系统随着航空公司载客量的迅猛增长和人们对便捷性要求的提高,已经无法满足需求。原有的系统不仅效率比较低下,而且在安全性、准确性等方面有很多不足。

为了实现航空公司以及旅游行业的现代化管理,进一步提高工作效率,方便旅客,需要开发一个机票预订系统。该系统需要具有完整的存储、查询、核对、打印机票的功能。

在这个系统中,旅客或工作人员通过机票预订系统查询,为旅客安排航班,打印取票通知和账单,旅客在飞机起飞的前一天凭取票通知和账单交款取票,系统校对无误即可打印机票给旅客。

1.3 定义

1.4 参考资料

2 任务概述

2.1 目标

机票预订系统的总目标:在计算机网络、数据库和先进的开发平台上,利用现有的软件,配置一定的硬件,开发一个具有开放体系结构的、易扩充的、易维护的、具有良好人机交互界面的机票预订系统,实现航空公司的机票销售自动化的计算机系统,为企业的决策层提供准确、精细、迅速的机票销售信息。

本机票预订系统实现后能够大大提高航空公司的机票预订服务效率,降低售票服务中的错误发生率,减少信息交流的烦琐过程及其带来的开销。

2.2 用户特点

使用本系统的最终用户可以定位为所有计算机使用者,尤其以旅游人员为主。由于本软件系统操作简单,界面友好,所以对用户的教育水平和技术水平几乎没有任何要求,只要会用计算机进行常规使用的用户均可使用。本系统将可能会被反复使用于日常的上网过程当中。

2.3 假定和约束

普通管理员,只能对数据库(航班库和客户库)中的信息进行查询操作;系统维护人员,可以根据具体需要进行适当的数据管理(增、删、改、更)。

客户只能对航班信息库中的内容进行查询操作,客户进入页面之后在不进行登录的情

况下只能进行航班信息查询操作,预订机票必须先注册登录提交自己的基本信息;系统会根据管理员和客户的各种操作做出相应的返回信息进行提示。

3 机票预订系统数据描述

3.1 系统业务流程图描述

首先分析本系统总的业务流程图,如图 3-20 所示。机票预订系统的主要业务为订票业务、取票业务和退票业务,其业务流程图分别如图 3-21~图 3-23 所示。

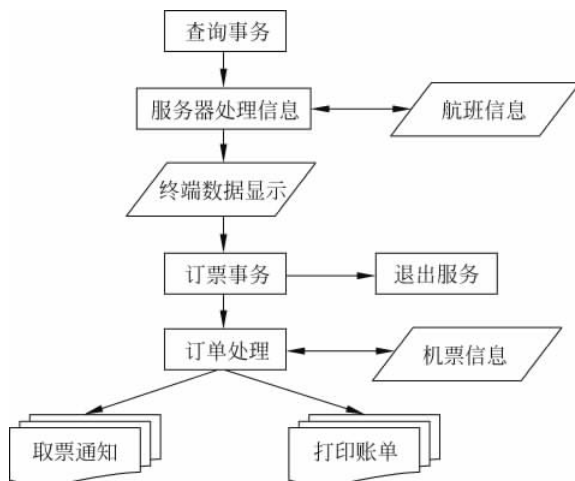


图 3-20 业务流程图

(1) 订票业务。根据旅客提出的要求(航班号、订票数额)查询该航班票额情况。若尚有余票,则为客户办理订票手续,输出座位号;若已满员或余票少于订票额,则登记排队候补。

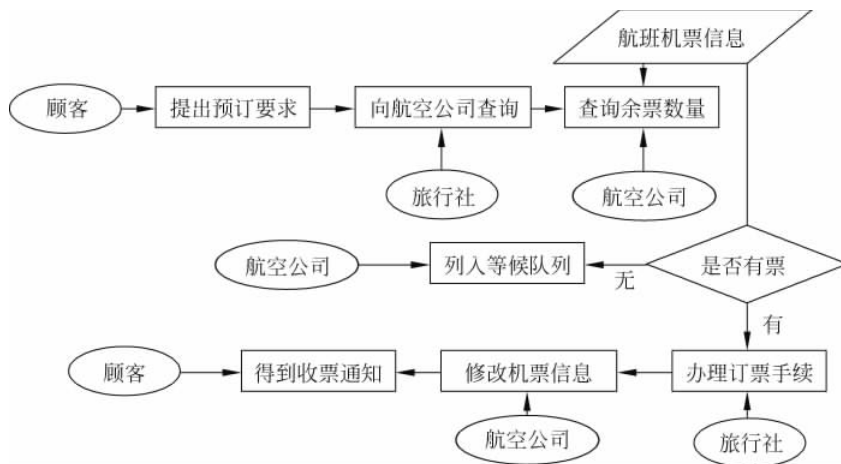


图 3-21 订票业务流程图

(2) 取票业务。根据取票通知书,打印机票,交给顾客。

(3) 退票业务。根据客户提供的情况(日期、航班),为客户办理退票手续,然后查询该

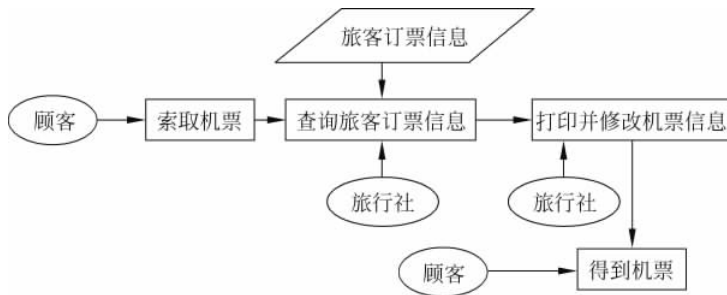


图 3-22 取票业务流程图

航班是否有人排队候补,首先询问排在第一的客户,若所退票额能满足他的要求,则为他办理订票手续,否则依次询问其他候补的客户。

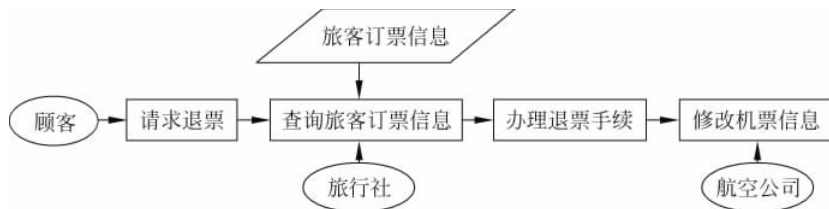


图 3-23 退票业务流程图

3.2 机票预订系统的数据需求

机票预订系统的数据需求包括如下几点:

(1) 数据录入和处理的准确性和实时性。数据的输入是否准确是数据处理的前提,错误的输入会导致系统输出的不正确和不可用,从而使系统的工作失去意义。数据的输入来源是手工输入。手工输入要通过系统界面上的安排。系统具有容错性,并且对操作人员要进行系统的培训。在系统中,数据的输入往往是大量的,因此,系统要有一定的处理能力,以保证迅速地处理数据。

(2) 数据的一致性与完整性。由于系统的数据是共享的,在不同的旅行社中,机票是共享数据,所以如何保证这些数据的一致性,是系统必须解决的问题。要解决这一问题,要有一定的人员维护数据的一致性,在数据录入处控制数据去向,并且要求对数据库的数据完整性进行严格的约束。对于输入的数据,要为其定义完整性规则,如果不能符合完整性约束,则系统应该拒绝该数据。

(3) 数据的共享与独立性。整个机票预订系统的数据是共享的。然而,从系统开发的角度上看,共享会给设计和调试带来困难。因此,应该提供灵活的配置,使各个分系统能够独立运行,而通过人工干预的手段进行系统数据的交换。这样,也能提供系统的健壮性。

3.3 机票预订系统数据流图

首先分析系统总的数据流程图,如图 3-24 所示。

由于系统的数据流程图比较复杂,而系统中主要的是订票、取票和退票三项业务,因此,分别画出这三项业务的数据流程图,如图 3-25~图 3-27 所示。

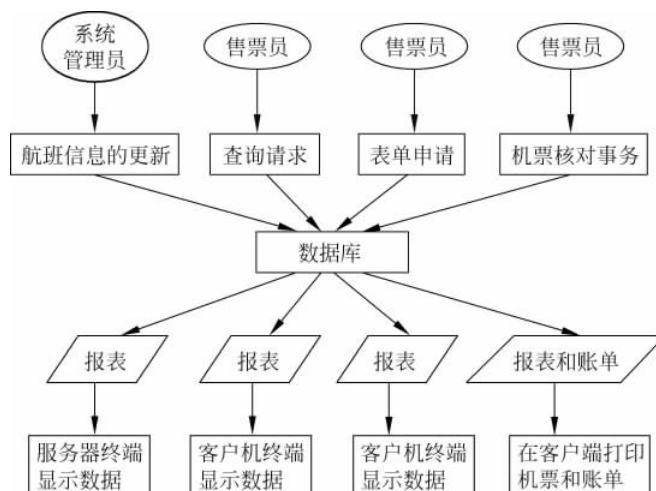


图 3-24 数据流程图

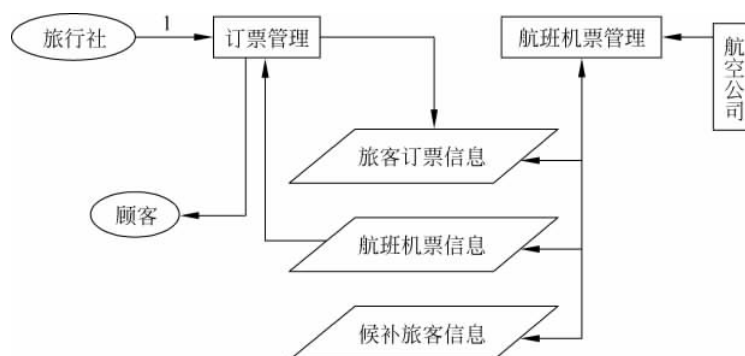


图 3-25 订票业务数据流程图

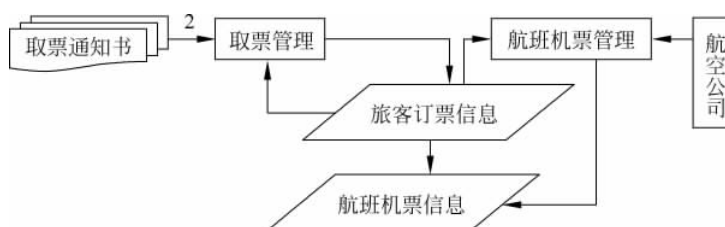


图 3-26 取票业务数据流程图

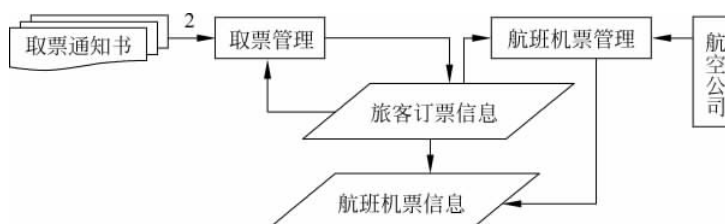


图 3-27 退票业务数据流程图

3.4 机票预订系统数据字典

数据字典是用来规范描述数据具体内容的工具,也是对数据汇总分析的一个总结。一般来说,可为每个数据建立一张二维表。在本系统中,分别为旅客信息、旅客订票信息、候补旅客信息、航班机票信息、取票通知和售出机票信息建立了数据字典,具体如图 3-28~图 3-33 所示。

数据名称: 旅客信息
作用: 记录旅客个人信息,以便对旅客进行确认
来源: 旅客
去向: 旅行社
排列方式: 按照旅客姓名首字母排序
包含信息: 姓名、性别、工作单位、身份证号(护照号码)、旅行时间、旅行始发地和目的地,航班舱位要求

图 3-28 游客信息数据字典

数据名称: 旅客订票信息
作用: 记录旅客的旅行时间和目的地,以便作合适的航班安排
来源: 旅客
去向: 航班机票管理
排列方式: 按照旅客首字母(姓名+起飞时间)排序
包含信息: 旅客姓名、身份证号(护照号码)、航班号、起飞时间、预订票数、舱位等级、机票是否打印

图 3-29 旅客订票信息数据字典

数据名称: 候补旅客信息
作用: 记录旅客的旅行时间和目的地,以便及时调整航班
来源: 航班机票管理
去向: 信息存储/航班机票管理
排列方式: 按照旅客首字母(姓名+起飞时间)排序
包含信息: 旅客姓名、身份证号(护照号码)、航班号、起飞时间、预订票数、舱位等级

图 3-30 候补旅客信息数据字典

数据名称: 航班机票信息
作用: 记录旅客的航班信息,根据旅客的旅行时间和目的地安排航班
来源: 航空公司
去向: 信息存储/航班机票管理
排列方式: 按照机票(飞行日期+航班号)升序排列
包含信息: 航班号、终点站、起飞地点、飞行日期、成员定额、余票量、已订票客户名单(姓名+订票量+舱位等级)、候补客户名单(姓名+所需票量)

图 3-31 航班机票信息数据字典

数据名称: 取票通知
作用: 旅客领取机票的凭证
来源: 航班机票管理系统
去向: 旅客
排列方式: 按照旅客(姓名+飞行日期+航班号)升序排列
包含信息: 旅客姓名、领票时间

图 3-32 取票通知数据字典

数据名称: 售出机票信息
作用: 记录旅客航班机票的信息
来源: 航班机票管理系统
去向: 信息存储/航班机票管理
排列方式: 按照旅客(姓名+飞行日期+航班号)升序排列
包含信息: 旅客的航班机票

图 3-33 售出机票信息数据字典

3.5 机票预订系统的逻辑模型

系统的逻辑方案是指在对现行系统进行分析和优化的基础上,确定新系统的目标、信息流程、总体结构、功能模型以及拟采用的管理模型和信息处理方法等。详细地了解情况和进行系统分析都是为最终建立新系统的逻辑模型做准备。机票预订系统的逻辑模型如

图 3-34 所示。

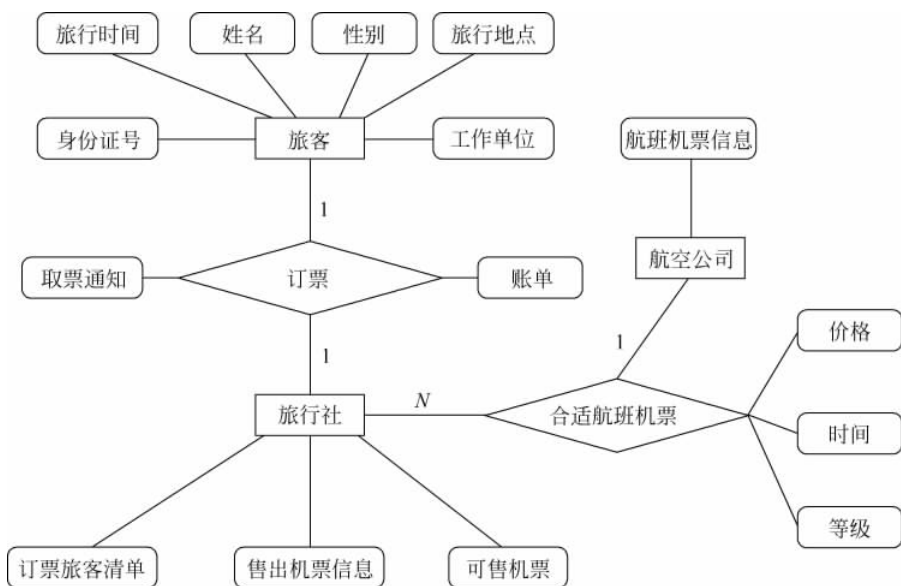


图 3-34 机票预订系统的逻辑模型

4 机票预订系统的功能要求

4.1 功能划分

根据可行性研究的结果和客户的要求,分析现有情况及问题,采用 Client/Server 结构,将机票预订系统划分为两个子系统:客户端子系统、服务器子系统。

4.2 功能描述

下面分析各个子系统的功能需求:

(1) 客户端子系统。

在客户端系统的功能实现上,可以分为以下几个部分:

① 旅客信息的输入和统计。旅行社把旅客要求订票的信息由专人负责输入。这部分功能是客户端子系统的基本部分,这个功能是以以后各个部分的基础。系统要求做到既能够从其他子系统中共享一部分信息,又有方便的操作界面供手工输入旅客信息。这部分要求对输入的数据进行简单的统计,供航空公司进行查询和宏观调控。

② 旅客信息的存储:将旅客的信息存储到旅行社的客户端系统中,以备以后的取票确认以及查询。

③ 机票信息的传递及接收:将旅客所需的机票信息由旅行社客户端由网络传到航空公司的服务器上,并且接受航空公司返回的航班信息,然后存储起来。

④ 取票通知及账单的生成和打印:把已存储的从航空公司返回的航班机票信息打印出来,并且生成账单打印出来一起交给旅客。

印出机票给已经订票的旅客:根据旅客的取票通知及账单,经过确认无误,接受旅客的付款后把机票印出来交给旅客。

⑤ 机票销售情况的核算:这一功能是在上一功能的基础上,对机票销售额进行单项核算,得到该旅行社的销售情况并把核算结果作为企业报表输出。

(2) 服务器端的功能要求。

通过计算机网络将客户端与服务器的数据库相连,将从客户端得到的信息进行处理,实现航班查询、机票生成、销售统计、综合信息查询等子系统。以计算机成本核算为中心,实现销售业务的计算机自动化,为航空公司降低成本、提高销售额、经营决策提供及时精确的依据。

在客户端系统的功能实现上,可以分为以下几个部分:

① 接收由旅行社客户端发回的所需机票信息:通过网络接收机票信息并存入服务器的数据库中。

② 生成航班信息:根据所需机票信息(时间、地点),在数据库中查询并得到正确的航班信息(价格、时间、等级),分配所需的机票数并在数据库中做出已售出的标记。

③ 传递航班信息到客户端(旅行社):把得到的航班信息通过网络传递到旅行社。

④ 接收旅行社的反馈信息:对旅行社的反馈信息进行分析,把已经售出的机票进行统计,对被旅客所退掉的机票要进行数据库的恢复。

⑤ 印出机票给已经订票的旅客:根据旅客的取票通知及账单,经过确认无误,接受旅客的付款后把机票印出来交给旅客。

⑥ 销售额的分析和管理的:要求包括对销售的机票进行分析,这一工作是在前面的基础上,以计算机为工具,对机票预订系统的功能和目标进行扩充。它以财务管理学为理论基础,以辅助决策为目标,以机票销售数据为中心,广泛采用统计学、运筹学等分析方法,对销售信息进行深层加工,建立反映不同航班需求的模型,提供管理所需的各种辅助决策信息和财务信息。这一要求是机票预订系统的最高目标,在通过系统运行后获得的大量销售历史数据基础上实现这一目标。

5 机票预订系统的性能要求

为了保证系统能够长期、安全、稳定、可靠、高效地运行,机票预订系统应该满足以下性能需求。

5.1 系统处理的准确性和及时性

系统处理的准确性和及时性是系统的必要性能。在系统设计和开发过程中,要充分考虑系统当前和将来可能承受的工作量,使系统的处理能力和响应时间能够满足企业对信息处理的需求。

机票预订系统的查询功能对于整个系统的功能和性能完成举足轻重。作为系统的很多数据来源,机票数量和时间又影响企业的决策活动,其准确性很大程度上决定了机票预订系统的成败。在系统开发过程中,必须采用一定的方法保证系统的准确性。

5.2 系统的开放性和系统的可扩充性

机票预订系统在开发过程中,应该充分考虑以后的可扩充性。例如,订票系统中订票方式的改变(网上订票),用户查询的需求也会不断地更新和完善。所有这些,都要求系统提供足够的手段进行功能的调整和扩充。而要实现这一点,应通过系统的开放性来完成,既系统应是一个开放系统,只要符合一定的规范,就可以简单地加入和减少系统的模块,配置系统的硬件。通过软件的修补、替换完成系统的升级和更新换代。

5.3 系统的易用性和易维护性

机票预订系统是直接面对使用人员的,而使用人员往往对计算机并不是非常熟悉。这

就要求系统能够提供良好的用户接口,易用的人机交互界面。要实现这一点,就要求系统应该尽量使用用户熟悉的术语和中文信息的界面;针对用户可能出现的使用问题,要提供足够的在线帮助,缩短用户对系统熟悉的过程。

机票预订系统中涉及的数据是航空公司相当重要的信息,系统要提供方便的手段供系统维护人员进行数据的备份、日常的安全管理、系统意外崩溃时数据的恢复等工作。

5.4 系统的标准性

系统在设计开发使用过程中涉及很多计算机硬件、软件,所有这些都要符合主流国际、国家和行业标准。例如,在开发中使用的操作系统、网络系统、开发工具都必须符合通用标准。如规范的数据库操纵界面、作为业界标准的 TCP/IP 网络协议及 ISO 9002 标准所要求的质量规范等。同时,在自主开发本系统时,要进行良好的设计工作,制订行之有效的软件工程规范,保证代码的易读性、可操作性和可移植性。

5.5 系统的先进性

目前计算系统的技术发展相当快,作为机票预订系统工程,应该保证系统在以后某个时期仍旧是先进的,在系统的生命周期尽量做到系统的先进,充分完成企业信息处理的要求而不至于落后。这一方面通过系统的开放性和可扩充性,不断改善系统的功能完成。另一方面,在系统设计和开发的过程中,应在考虑成本的基础上尽量采用当前主流并先进且有良好发展前途的产品。

5.6 系统的响应速度

机票预订系统在日常处理中的响应速度为秒级,达到实时要求,以及时反馈信息。在进行统计分析时,根据所需数据量的不同而从秒级到分钟级,原则是保证操作人员不会因为速度问题而影响工作效率。

6 机票预订系统的运行要求

机票预订系统中的各个子系统的硬件和软件的配置如下:

(1) 服务器端子系统的运行要求:

系统软件: Windows NT Sever。

数据库管理系统: SQL Sever 2010。

硬件要求: Pentium III 800 以上,256MB RAM,40GB HD。

(2) 客户端子系统的运行要求:

系统软件: Windows NT Workstation 及以上版本。

数据库管理系统: SQL Sever。

硬件要求: Pentium 450 以上,128MB RAM,10GB HD。

3.6 习题

1. 需求分析的任务是什么? 怎样理解分析阶段的任务是决定“做什么”,而不是“怎样做”?
2. 怎样建立目标系统的逻辑模型? 要经过哪些步骤?
3. 需求说明(或需求规格说明书)由哪些部分组成? 各部分之间的关系是什么?
4. 为什么 DFD 要分层? 绘制 DFD 要遵循哪些原则?

5. 选择一个系统(如人事档案管理系统、图书管理系统、医院监护系统、足球俱乐部管理系统、财务管理系统、学生成绩管理系统、飞机订票系统等),用 SA 方法对它进行分析,画出系统的分层 DFD 图,并建立相应的数据词典。

6. 试说明需求分析工具的一般组成。

7. 考察图 5-35 和图 5-36 中父图、子图的平衡关系。

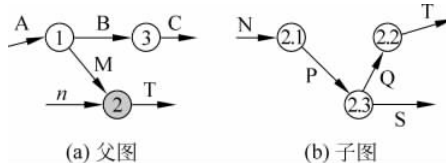


图 3-35 题 7 图 1

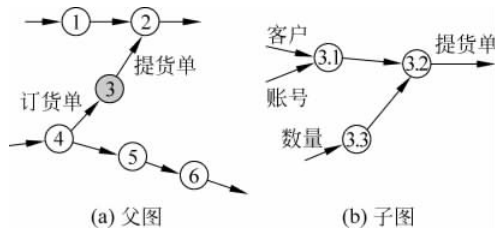


图 3-36 题 7 图 2