

第5章

结构化分析



扫码观看视频

本章首先概述需求分析和结构化分析；然后介绍结构化分析的方法，包括功能建模、数据建模、行为建模、数据字典和加工规格说明；接着描述结构化分析的图形工具，包括层次方框图、Warnier 图和 IPO 图。

本章目标

- ☐ 了解需求分析的任务和原则
- ☐ 熟悉进行需求分析的步骤和方法
- ☐ 了解需求管理
- ☐ 熟悉需求分析的常用方法
- ☐ 了解软件原型
- ☐ 掌握结构化分析的几种常用建模方法
- ☐ 掌握结构化分析的几种图形工具

5.1 需求分析

5.1.1 需求分析的任务和原则

1. 为什么需要需求分析

为了开发出真正满足用户需要的软件产品，明确地了解用户需求是关键。虽然在可行性研究中，已经对用户要求有了初步的了解，但是很多细节还没有考虑到。可行性研究的目的是评估系统是否值得去开发，问题是否能够解决，而不是对需求进行定义。如果说可行性分析是要决定“做还是不做”，那么需求分析就是要回答“系统必须做什么”这个问题。

在需求中会存在大量的错误，这些错误若未及时发现和更正，会引起软件开发费用增加、软件质量降低；严重时，会造成软件开发的失败。在对以往失败的软件工程项目进行失

败原因分析和统计的过程中发现,因为需求不完整而导致失败的项目约占 13.1%,缺少用户参与导致项目失败约占 12.5%,需求和需求规格说明书更改约占 8.7%。可见约三分之一的项目失败都与需求有关。要尽量避免需求中出现的错误,就要进行详细而深入的需求分析。可见需求分析是一个非常重要的过程,它完成的好坏直接影响了后续软件开发的质量。

2. 确定系统的运行环境要求

系统运行时的硬件环境要求,如对计算机的 CPU、内存、存储器、输入/输出方式、通信接口和外围设备等的要求;软件环境要求,如操作系统、数据库管理系统和编程语言等的要求。

3. 确定系统的功能性需求和非功能性需求

需求可以分为两大类,功能性需求和非功能性需求,前者定义了系统做什么,后者定义了系统工作时的特性。

功能需求是软件系统的最基本的需求表述,包括对系统应该提供的服务,如何对输入做出反应,以及系统在特定条件下的行为描述。在某些情况下,功能需求还必须明确系统不应该做什么,这取决于开发的软件类型、软件未来的用户以及开发的系统类型。所以,功能性的系统需求,需要详细地描述系统功能特征、输入和输出接口、异常处理方法等。

非功能性需求包括对系统提出的性能需求,可靠性和可用性需求,系统安全以及系统对开发过程、时间、资源等方面的约束和标准等。性能需求指定系统必须满足的定时约束或容量约束,一般包括速度(响应时间)、信息量速率(吞吐量、处理时间)和存储容量等方面的需求。

4. 进行有效的需求分析

一般情况下,用户并不熟悉计算机的相关知识,而软件开发人员对相关的业务领域也不甚了解,用户与开发人员之间对同一问题理解的差异和习惯用语的不同往往会对需求分析带来很大的困难。所以,开发人员和用户之间充分和有效的沟通在需求分析的过程中至关重要。

有效的需求分析通常都具有一定的难度,这一方面是由于交流障碍所引起的,另一方面是由于用户通常对需求的陈述不完备、不准确和不全面,并且还可能在不断地变化。所以开发人员不仅需要在用户的帮助下抽象现有的需求,还需要挖掘隐藏的需求。此外,把各项需求抽象为目标系统的高层逻辑模型对日后的开发工作也至关重要。合理的高层逻辑模型是系统设计的前提。

5. 在需求分析的过程中应该遵守一些原则

首先,需求分析是一个过程,它应该贯穿系统的整个生命周期,而不是仅仅属于软件生命周期早期的一项工作。

其次,需求分析应该是一个迭代的过程。由于市场环境的易变性以及用户本身对于新系统要求的模糊性,需求往往很难一步到位。通常情况下,需求是随着项目的深入而不断地变化的。所以需求分析的过程应该是一个迭代的过程。

最后,为了方便评审和后续的设计,需求的表述应该具体、清晰,并且是可度量的、可实现的。最好能够对需求进行适当的量化。例如:系统的响应时间应该低于 0.5s;系统在同

一时刻最多能支持 30 000 个用户。

6. 需求分析的两个任务

首先,是需求分析的建模阶段,即在充分了解需求的基础上,要建立起系统的分析模型。其次,是需求分析的描述阶段,就是把需求文档化,用软件需求规格说明书的方式把需求表达出来。

7. 软件需求规格说明书

软件需求规格说明书是需求分析阶段的输出,它全面、清晰地描述了用户需求,因此是开发人员进行后续软件设计的重要依据。软件需求规格说明书应该具有清晰性、无二义性、一致性和准确性等特点。同时,它还需要通过严格的需求验证、反复修改的过程才能最终确定。

5.1.2 需求分析的步骤

为了准确获取需求,需求分析必须遵循一系列的步骤。只有采取了合理的需求分析的步骤,开发人员才能更有效地获取需求。一般来说,需求分析分为需求获取、分析建模、需求描述和需求验证与评审 4 步。以下将分步进行介绍。

1. 需求获取

需求获取就是收集并明确用户需求的过程。系统开发方人员通过调查研究来理解当前系统的工作模型、用户对新系统的设想与要求。在需求获取的初期,用户提出的需求一般模糊而且凌乱,这就需要开发人员能够选取较好的需求分析的方法,提炼出逻辑性强的需求。例如,没有经验或只是偶尔使用系统的用户关心的是系统操作的易学性。他们喜欢具有菜单、图形界面、整齐有序的屏幕显示、详细的提示以及使用向导的系统。而熟悉系统的用户会更关心使用的方便性与效率,并看重快捷键、宏、自定义选项、工具栏、脚本功能等。而且不同用户的需求可能会发生冲突,例如,对于同样一个人力资源管理系统,某些用户希望系统反应速度快一些,查找准确;但另一些用户希望做到安全性第一,而对反应速度要求不高。因此,对于发生冲突的需求必须仔细考虑并做出选择。

获取需求的方法有多种,如问卷调查、访谈、实地操作、建立原型等。

(1) 问卷调查是采用让用户填写问卷的形式了解用户对系统的看法。问题应该是循序渐进的,并且可选答案不能太局限,以免限制了用户的思维。回收问卷后,开发人员要对其进行汇总、统计,从而分析出有用信息。

(2) 访谈是指开发人员与特定的用户代表进行座谈的需求获取方法。在进行访谈之前,访谈人员应该准备好问题。一般情况下问题涉及的方面主要是什么时候(When)、在哪里(Where)、做什么(What)、谁(Who)、为什么(Why)、如何(How)。问题可以分为开放性问题 and 封闭式问题。开放性问题是指答案比较自由的问题,例如:对于这个项目,你的看法是什么?而封闭式问题是指答案受限制的问题,例如:对于这个项目,你是赞成还是反对?在访谈中,分析人员应该多提一些开放性问题,这样更容易捕捉到用户的真实想法。由于被访谈的用户身份多种多样,因此在访谈的过程中,访谈人员要根据用户的身份,提不同的问题,这样访谈才能更有效。

(3) 如果开发人员能够以用户的身份参与现有系统的使用过程,那么在亲身实践的基

基础上,观察用户的工作过程,发现问题并及时提问,开发人员就能直接地体会到现有系统的弊端以及新系统应该解决的问题。这种亲身实践的需求获取方法就是实地操作。这种方式可以帮助开发人员获得真实的信息。

(4) 为了进一步挖掘需求,了解用户对目标系统的想法,开发人员有时还采用建立原型系统的方法。在原型系统中,用户更容易表达自己的需求。所谓原型,就是目标系统的一个可操作的模型。原型化分析方法要求在获得一组基本需求说明后,能够快速地对某些重要方面“实现”,通过原型反馈加深对系统的理解,并对需求说明进行补充和优化。利用原型的需求获取过程可以用图 5-1 来表示。

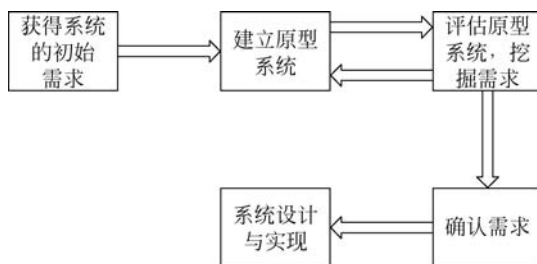


图 5-1 原型化开发过程

针对前述所获取的需求,要进行归纳,形成软件需求。软件需求包括功能需求、性能需求、系统运行环境需求、用户界面需求、软件成本消耗与开发进度需求、资源使用需求、可靠性需求、安全保密要求等。

2. 分析建模

获取需求后,下一步就应该对开发的系统建立分析模型了。模型就是为了理解事物而对事物做出的一种抽象,通常由一组符号和组织这些符号的规则组成。对待开发系统建立各种角度的模型有助于人们更好地理解问题。通常,从不同角度描述或理解软件系统,就需要不同的模型。常用的建模方法有数据流图、实体关系图、状态转换图、控制流图、用例图、类图、对象图等。

3. 需求描述

需求描述就是指编制需求分析阶段的文档。一般情况下,对于复杂的软件系统,需求阶段会产生 3 个文档:系统定义文档(用户需求报告)、系统需求文档(系统需求规格说明书)、软件需求文档(软件需求规格说明书)。而对简单的软件系统而言,需求阶段只需要输出软件需求文档就可以了。软件需求文档主要描述软件部分的需求,简称 SRS (software requirement specification),它站在开发者的角度,对开发系统的业务模型、功能模型、数据模型、行为模型等内容进行描述。经过严格的评审后,它将作为概要设计和详细设计的基础。

对于大型软件项目,在需求分析阶段要完成多项文档,包括可行性研究报告、项目开发计划、软件需求说明、数据要求说明和测试计划。对于中型软件项目,可行性研究报告和项目开发计划可以合并为项目开发计划,软件需求说明和数据要求说明可以合并为软件需求说明。而对于小型软件项目,一般只需完成软件需求说明与开发计划就可以了,如

图 5-2 所示。



图 5-2 文档与软件规模的对应关系

4. 需求验证与评审

需求分析的第四步是验证与评审以上需求分析的成果。需求分析阶段的工作成果是后续软件开发的重要基础,为了提高软件开发的质量,降低软件开发的成本,必须对需求的正确性进行严格验证,确保需求的一致性、完整性、现实性、有效性。确保设计与实现过程中需求的可回溯性,并进行需求变更管理。

需求评审就是通过将需求规约文档发布给利益相关者进行检查,发现需求规约中存在缺陷(如错误、不完整性、二义性等)的过程。对工作产品的评审有两类方式,一类是正式技术评审,也称同行评审,另一类是非正式技术评审。对于任何重要的工作产品,都应该至少执行一次正式技术评审。在进行正式技术评审前,需要有人对要进行评审的工作产品进行把关,确认其是否具备进入评审的初步条件。

需求评审的规程与其他重要工作产品(如系统设计文档、源代码)的评审规程非常相似,主要区别在于评审人员的组成不同。前者由开发方和客户方的代表共同组成,而后者通常来源于开发方内部。

需求分析步骤如图 5-3 所示。

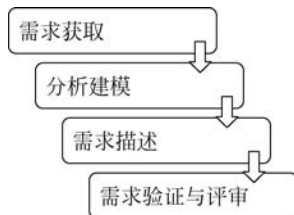


图 5-3 需求分析的步骤

5.1.3 需求管理

为了更好地进行需求分析并记录需求结果,需要进行需求管理。需求管理是一种用于查找、记录、组织和跟踪系统需求变更的系统化方法,可用于:

- 获取、组织和记录系统需求;
- 使客户和项目团队在系统变更需求上达成并保持一致。

有效需求管理的关键在于维护需求的明确阐述、每种需求类型所适用的属性,以及与其他需求和其他项目之间的可追踪性。

软件需求的变更往往贯穿软件的整个生命周期。需求管理是一组活动,用于在软件开发的过程中标识需求、控制需求、跟踪需求,并对需求变更进行管理(需求变更管理活动与“13.7 软件配置管理”中所述的基本相同,因此,具体内容可参见 13.7 节)。

需求管理实际上是项目管理的一个部分,它涉及以下 3 个主要问题。

(1) 识别、分类、组织需求,并为需求建立文档。

(2) 需求变化,是建立对需求不可避免的变化是如何提出、如何协商、如何验证以及如何形成文档的过程。

(3) 需求的可跟踪性,即带有维护需求之间以及与系统的其他制品之间依赖关系的过程。

5.1.4 需求分析的常用方法

需求分析的方法有多种,下面只简单介绍功能分解方法、结构化分析方法、信息建模方法和面向对象的分析方法。

1. 功能分解方法

功能分解方法是将一个系统看成由若干功能模块组成,每个功能又可分解为若干子功能及接口,子功能再继续分解,即功能、子功能和功能接口成为功能分解方法的 3 要素。功能分解方法采用的是自顶向下、逐步求精的理念。

2. 结构化分析方法

结构化分析方法是一种从问题空间到某种表示的映射方法,其逻辑模型由数据流图和数据词典构成并表示。它是一种面向数据流的需求分析方法。它主要适用于数据处理领域问题。本章将详细介绍这种方法。

3. 信息建模方法

模型是用某种媒介对相同媒介或其他媒介里的一些事物进行表现的形式。从建模角度出发,模型就是要抓住事物的最重要方面而简化或忽略其他方面。简而言之,模型就是对现实的简化。建立模型的过程,称为建模。

建模可以帮助理解正在开发的系统,这是需要建模的一个基本理由。并且,人对复杂问题的理解能力是有限的。建模可以帮助开发者缩小问题的范围,每次着重研究一个方面,进而对整个系统产生更加深刻的理解。可以明确地说,越大、越复杂的系统,建模的重要性也越大。

信息建模方法常用的基本工具是 E-R 图,其基本要素由实体、属性和关系构成。它的核心概念是实体和关系,它的基本策略是从现实中找出实体,然后再用属性对其进行描述。

4. 面向对象的分析方法

面向对象的分析方法的关键是识别问题域内的对象,分析它们之间的关系,并建立 3 类模型,它们分别是描述系统静态结构的对象模型、描述系统控制结构的动态模型,以及描述系统计算结构的功能模型。其中,对象模型是最基本、最核心、最重要的。面向对象主要考虑类或对象、结构与连接、继承和封装、消息通信,只表示面向对象的分析中几项最重要特征。类的对象是对问题域中事物的完整映射,包括事物的数据特征(即属性)和行为特征(即服务)。第 8 章将详细介绍这种方法。

5.1.5 软件原型

软件原型是指在项目的前期阶段,系统分析人员根据对客户需求的理解和客户希望实现的结果,快速地给出一个翔实的产品雏形,然后与客户反复协商修改。软件原型是项目需求的部分实现或者可能的实现,可以是工作模型或者静态设计、详细的屏幕草图或者简单草图,最终可以据此形成实际的系统。软件原型的重点在于直观体现产品的主要界面风格及结构,并展示出主要功能模块以及它们之间的相互关系,不断澄清模糊部分,为后期的设计和代码编写提供准确的产品信息。软件原型可以明确并完善需求,减少风险,优化系统的易用性,研究设计选择方案,为最终的产品提供基础。软件原型设计是软件人员与客户沟通的最好工具,主流的软件原型设计工具有: Axure、Balsamiq Mockups、墨刀、Justinmind、iClap 和 Dreamweaver 等。

图 5-4 是使用软件原型设计工具墨刀制作的界面。



图 5-4 用软件原型设计工具墨刀制作的界面

5.2 结构化分析概述

一种考虑数据和处理的需求分析方法被称作结构化分析方法(structured analysis, SA),它是 20 世纪 70 年代由 Yourdon Constaintine 及 DeMarco 等提出和发展,并得到广泛应用的。它基于“分解”和“抽象”的基本思想,逐步建立目标系统的逻辑模型,进而描绘出满足用户要求的软件系统。

“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,可以把大问题分解为若干个小问题,然后再分别解决。图 5-5 所示为对目标系统 X 进行自顶向下逐层分解。

最顶层描述了整个目标系统 X,中间层将目标系统划分为若干个模块,每个模块完成一定的功能,而最底层是对每个模块实现方法的细节性描述。可见,在逐层分解的过程中,起初并不考虑细节性的问题,而是先关注问题最本质的属性,随着分解自顶向下进行,才会逐渐考虑到越来越具体的细节。这种用最本质的属性表示一个软件系统的方法就是“抽象”。

结构化分析方法是一种面向数据流的需求分析方法,其中数据作为独立实体转换,数据建模定义了数据的属性和关系,操作数据的处理建模表明当数据在系统中流动处理时如何转换数据。

结构化分析的具体步骤如下。

(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,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,获得当前系统的“逻辑模型”。

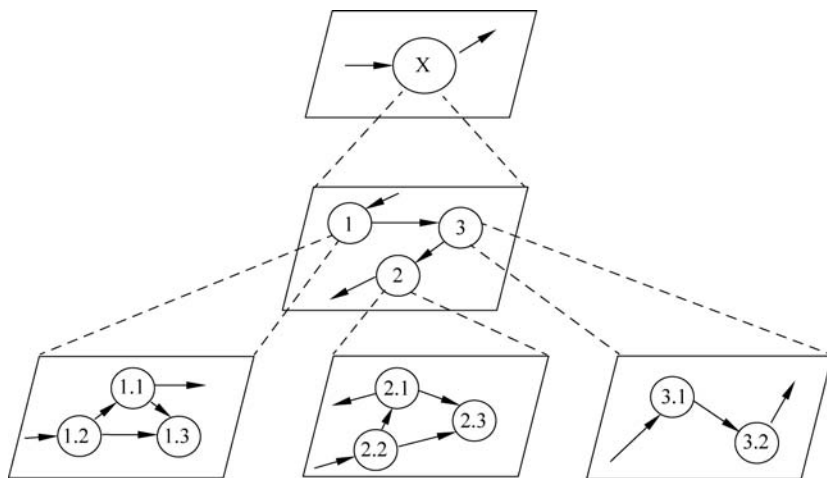


图 5-5 自顶向下逐层分解

(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别，从而进一步明确目标系统“做什么”，建立目标系统的“逻辑模型”。

(4) 为了对目标系统进行完整描述，还需要考虑人机界面和其他一些问题。

5.3 结构化分析方法

结构化分析实质上是一种创建模型的活动，它建立的分析模型如图 5-6 所示。



图 5-6 结构化分析模型

此模型的核心是“数据字典”，它描述软件使用或产生的所有数据对象。围绕着这个核心有 3 种不同的图：“数据流图”指出当数据在软件系统中移动时怎样被变换，以及描绘变换数据流的功能和子功能，用于功能建模；“实体-关系图”(E-R 图)描绘数据对象之间的关系，用于数据建模；“状态转换图”指明作为外部事件结果的系统行为，用于行为建模。

每种建模方法对应其各自的表达方式和规约,描述系统某一方面的需求属性。它们基于同一份数据描述,即数据字典。

结构化分析方法必须遵守下述准则。

- (1) 必须定义软件应完成的功能,这条准则要求建立功能模型。
- (2) 必须理解和表示问题的信息域,根据这条准则应该建立数据模型。
- (3) 必须表示作为外部事件结果的软件行为,这条准则要求建立行为模型。
- (4) 必须对描述功能、信息和行为的模型进行分解,用层次的方式展示细节。
- (5) 分析过程应该从要素信息移向实现细节。

需求分析中的建模过程是使用一些抽象的图形和符号来表述系统的业务过程、问题和整个系统。这种描述较自然语言的描述更易于理解。对模型的描述还是系统分析和设计过程之间的重要桥梁。

不同的模型往往表述系统需求的某一方面,而模型之间又相互关联,相互补充。除了用分析模型表示软件需求之外,还要写出准确的软件需求规格说明。模型既是软件设计的基础,也是编写软件规格说明的基础。

5.3.1 功能建模

功能建模的思想就是用抽象模型的概念,按照软件内部数据传递和变换的关系,自顶向下逐层分解,直到能够构建满足功能要求的可实现的软件为止。功能模型用数据流图来描述。

数据流图(简称 DFD 图)就是采用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,是结构化系统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。

1. 数据流图的表示符号

在数据流图中,存在 4 种表示符号。

- (1) 外部实体:表示数据的源点或终点,它是系统之外的实体,可以是人、物或者其他系统。
- (2) 数据流:表示数据流的流动方向。数据流可以从加工流向加工,从加工流向文件,从文件流向加工。
- (3) 数据变换:表示对数据进行加工或处理,如对数据的算法分析和科学计算。
- (4) 数据存储:表示输入或输出文件。这些文件可以是计算机系统外的外部或者内部文件,也可以是表、账单等。

数据流图主要有 Yourdon 和 Gane 两种表示方法。其符号约定如表 5-1 所示。

表 5-1 数据流图符号约定

表示符号	Yourdon	Gane
外部实体		
数据流		
数据变换		
数据存储		

2. 环境图

环境图(如图 5-7 所示)也称为系统顶层数据流图(或 0 层数据流图),它仅包含一个数据处理过程,也就是要开发的目标系统。环境图的作用是确定系统在其环境中的位置,通过确定系统的输入和输出与外部实体的关系确定其边界。

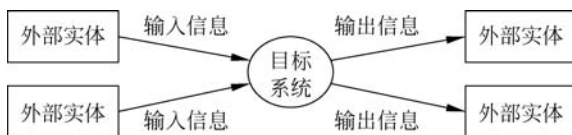


图 5-7 环境图

根据结构化需求分析采用的“自顶向下,由外到内,逐层分解”的思想,开发人员要先画出系统顶层的数据流图,然后再逐层画出低层的数据流图。顶层的数据流图要定义系统范围,并描述系统与外界的数据联系,它是对系统架构的高度概括和抽象。底层的数据流图是对系统某个部分的精细描述。

可以说,数据流图的导出是一个逐步求精的过程。其中要遵守如下一些原则。

- (1) 第 0 层的数据流图应将软件描述为一个泡泡。
- (2) 主要的输入和输出应该被仔细地标记。
- (3) 通过把在下一层表示的候选处理过程、数据对象和数据存储分离,开始求精过程。
- (4) 应使用有意义的名称标记所有的箭头和泡泡。
- (5) 当从一个层转移到另一个层时要保持信息流连续性。
- (6) 一次精化一个泡泡。

图 5-8 是某考务处理系统顶层 DFD 图。其中只用一个数据变换表示软件,即“考务处理系统”;包含所有相关外部实体,即考生、考试中心和阅卷站;包含外部实体与软件中间的数据流,但是不含数据存储。顶层 DFD 图应该是唯一的。

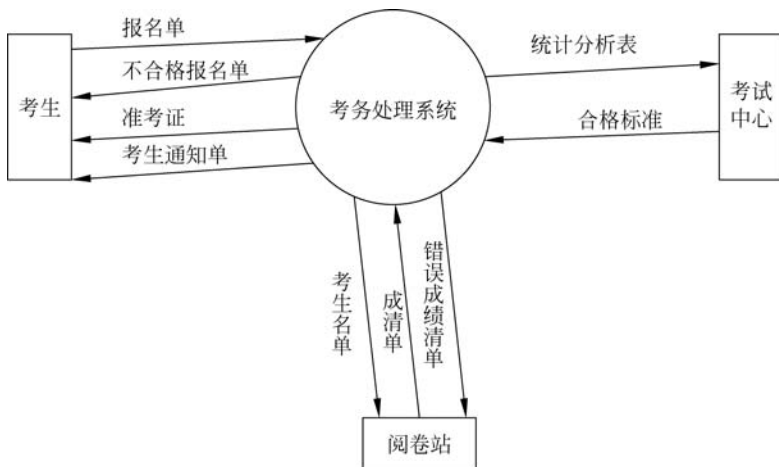


图 5-8 某考务处理系统顶层 DFD 图

3. 数据流图的分解

对顶层 DFD 图(见图 5-8)进行细化,得到 1 层 DFD 图,细化时要遵守上文所介绍的各

项原则,如图 5-9 所示。软件被细分为两个数据处理,“登记报名表”和“统计成绩”,即两个“泡泡”;同时引入了数据存储“考生名册”。

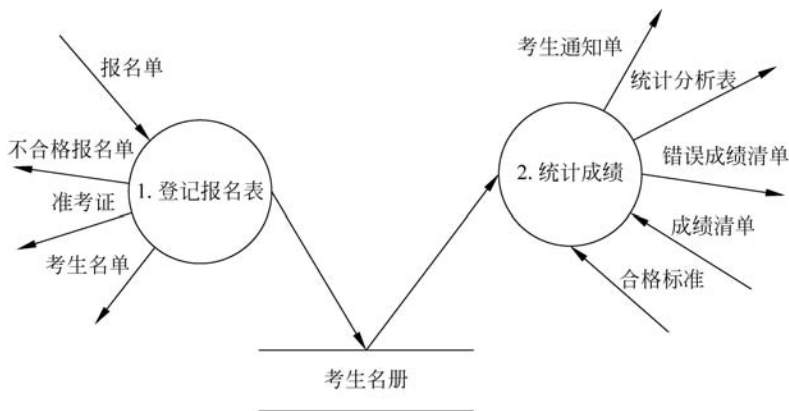


图 5-9 某考务处理系统 1 层 DFD 图

同理,可以对“登记报名表”和“统计成绩”分别细化,得到该系统两张 2 层 DFD 图,如图 5-10 和图 5-11 所示。

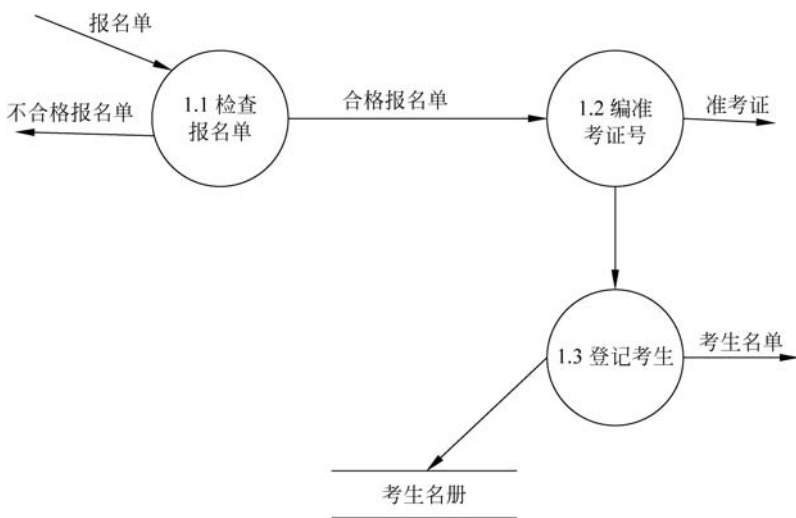


图 5-10 登记报名表 2 层 DFD 图

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。

- 自然性是指概念要合理和清晰。

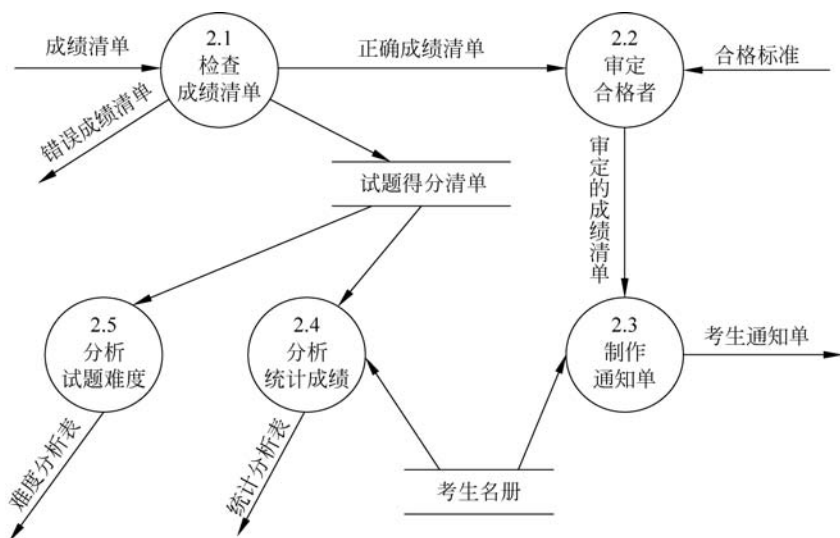


图 5-11 统计成绩 2 层 DFD 图

- 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。
- 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。

关于数据流图的绘制方法,本章的实例部分会详细介绍。

5.3.2 数据建模

数据建模的思想是在较高的抽象层次(概念层)上对数据库结构进行建模。数据模型用 E-R 图来描述。

实体-关系图(简称 E-R 图)可以明确描述待开发系统的概念结构数据模型。对于较复杂的系统,通常要先构造出各部分的 E-R 图,然后将各分 E-R 图集成总的 E-R 图,并对 E-R 图进行优化,以得到整个系统的概念结构模型。

在建模的过程中,E-R 图以实体、关系和属性 3 个基本概念概括数据的基本结构。实体就是现实世界中的事物,多用矩形框来表示,框内含有相应的实体名称。属性多用椭圆形表示,并用无向边与相应的实体联系起来,表示该属性归某实体所有。可以说,实体是由若干个属性组成的,每个属性都代表了实体的某些特征。例如,在某教务系统中“学生”实体的属性如图 5-12 所示。

关系用菱形表示,并用无向边分别与有关实体连接起来,以此描述实体之间的关系。实体之间存在着三种关系类型,分别是一对一、一对多、多对多,它们分别反映

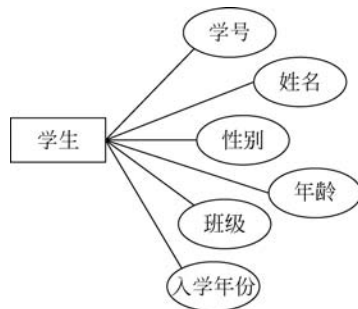


图 5-12 “学生”实体及其属性

了实体之间不同的对应关系。图 5-13 所示的“人员”与“车位”之间是一对一的关系,即一个人员只能分配一个车位,且一个车位只能属于一个人员。“订单”与“订单行”之间是一对多的关系,即一个订单包含若干个订单行,而一个订单行只属于一个订单。“学生”与“课程”之间是多对多的关系,即一个学生能登记若干门课程,且一门课程能被多个学生登记。

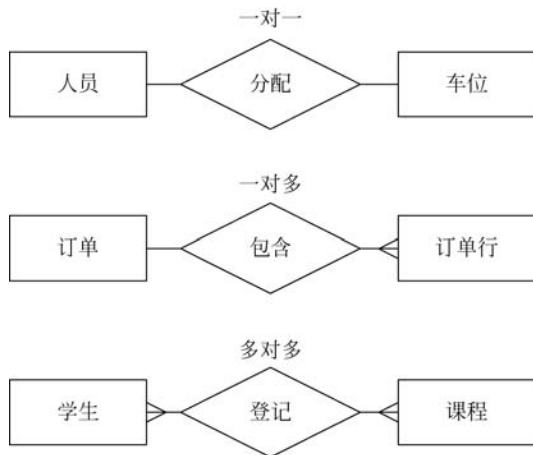


图 5-13 三种关系类型

图 5-14 是某教务系统中课程、学生和教师之间的 E-R 图。其中,方框表示实体,有学生、教师和课程三个实体;椭圆形表示实体的属性,如学生实体的属性有学号、姓名、性别和专业;菱形表示关系,学生和课程是选课关系,且是一个多对多关系,教师和课程是任教关系,且是一个一对多关系;实体与属性、实体与关系之间用实线进行连接。

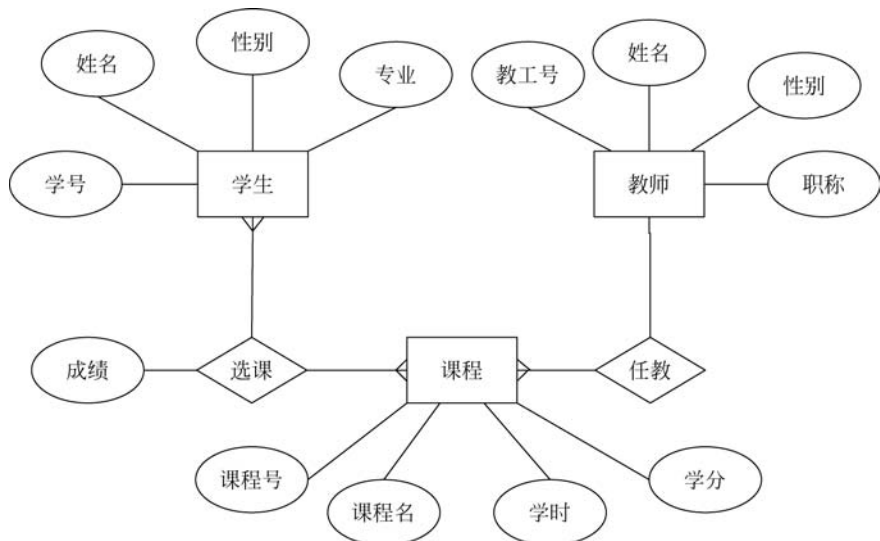


图 5-14 某教务系统 E-R 图

另外,关系本身也可能有属性,这在多对多的关系中尤其常见。如图 5-14 所示,成绩就是选课这个关系的一个属性。

运用 E-R 图,概念结构设计在调查用户需求的基础上,对现实世界中的数据及其关系

进行分析、整理和优化。需要指出的是,E-R图并不具有唯一性,也就是说,对于同一个系统,可能有多E-R图,这是由于不同的分析人员看问题的角度不同而造成的。

5.3.3 行为建模

状态转换图是一种描述系统对内部或外部事件响应的行为模型。它描述系统状态、事件和事件引发系统在状态之间的转换,而不是描述系统中数据的流动。这种模型尤其适合用来描述实时系统,因为这类系统多是由外部环境的激励而驱动的。

使用状态转换图具有以下优点。

- (1) 状态之间的关系能够直观地被捕捉到。
- (2) 状态转换图因其单纯性,能够机械地分析许多情况,可很容易地建立分析工具。
- (3) 状态转换图能够很方便地对应状态转换表等其他描述工具。

并不是所有系统都需要画状态转换图,有时系统中的某些数据对象在不同状态下会呈现不同的行为方式,此时应分析数据对象的状态,画出状态转换图,才可正确地认识数据对象的行为,并定义其行为。对这些行为规则较复杂的数据对象需要进行如下分析。

- (1) 找出数据对象的所有状态。
- (2) 分析在不同的状态下,数据对象的行为规则是否不同,若无不同则可将其合并成一种状态。
- (3) 分析从一种状态可以转换成哪几种状态,是数据对象的什么行为导致这种状态的转换。

1. 状态及状态转换

状态是任何可以被观察到的系统行为模式,一个状态代表系统的一种行为模式。状态规定了系统对事件的响应方式。系统对事件的响应,既可以是做一个(或一系列)动作,也可以是仅仅改变系统本身的状态,还可以是既改变状态又做动作。

在状态转换图中定义的状态主要有:初态(即初始状态)、终态(即最终状态)和中间状态。初态用黑圆点表示,终态用黑圆点外加一个圆表示(很像一只牛眼睛),状态图中的状态用圆角四边形表示(可以用两条水平横线把它分成上、中、下3个部分。上面部分为状态的名称,这部分是必须有的;中间部分为状态变量的名字和值,这部分是可选的;下面部分是活动表,这部分也是可选的),状态之间为状态转换,用带箭头的线表示。带箭头的线上的事件发生时,状态转换开始(有时也称之为转换被“触发”)。在一张状态转换图中只能有一个初态,而终态则可以没有,也可以有多个。状态转换图中使用的主要符号如图5-15所示。

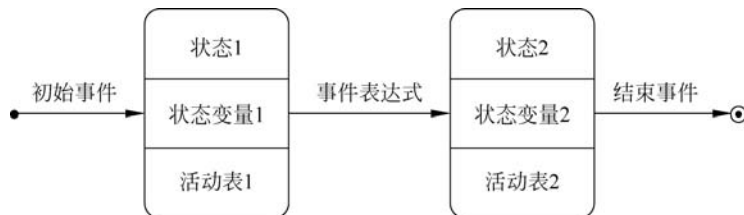


图 5-15 状态转换图中使用的主要符号

状态中的活动表的语法格式如下：

事件名(参数表)/动作表达式

其中,“事件名”可以是任何事件的名称,需要时可以为事件指定参数表;活动表中的动作表达式描述应做的具体动作。

2. 事件

事件是在某个特定时刻发生的事情,它是对引起系统做动作或(和)从一个状态转换到另一个状态的外界事件的抽象。例如,观众使用电视遥控器,用户移动鼠标、单击鼠标等都是事件。简而言之,事件就是引起系统做动作或(和)转换状态的控制信息。

状态转换通常是由事件触发的,在这种情况下应在表示状态转换的箭头线上标出触发转换的事件表达式。

如果在箭头线上未标明事件,则表示在源状态的内部活动执行完之后自动触发转换。事件表达式的语法如下:

事件说明[守卫条件]/动作表达式

事件说明的语法为:事件名(参数表)。

守卫条件是一个布尔表达式。如果同时使用事件说明和守卫条件,则当且仅当事件发生且布尔表达式为真时,状态转换才发生。如果只有守卫条件没有事件说明,则只要守卫条件为真,状态转换就发生。

动作表达式是一个过程表达式,当状态转换开始时执行该表达式。

3. 例子

为了具体说明怎样用状态转换图建立系统的行为模型,下面举一个例子。

图书馆管理系统:图书可被借阅、分类、归还、续借,图书也可能破损和遗失。根据以上情况画出图书馆管理系统中图书的状态转换图,如图 5-16 所示。

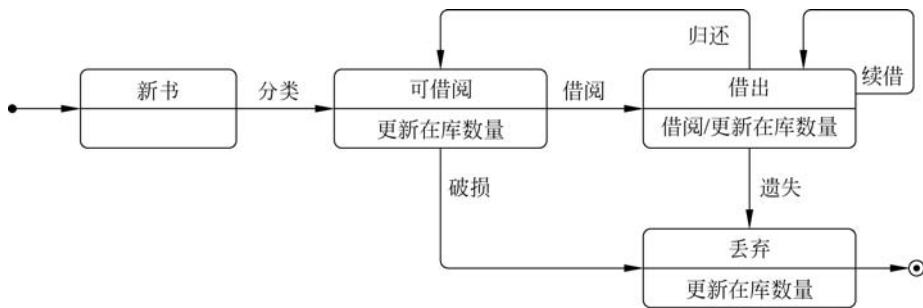


图 5-16 图书馆系统中图书的状态转换图

图书在初始时需要进行分类并更新在库数量。如果图书发生借阅,则执行借阅操作,并对在库图书数量进行更新。在借阅期间,如果图书发生续借操作,则对该图书重新执行借阅操作并更新在库数量。如果借阅的图书被归还,则需要对在库图书数量进行更新。此外,如果在库图书发生破损或者借阅图书发生遗失,则对在库图书的数量进行更新。

5.3.4 数据字典

如前所述,分析模型包括功能模型、数据模型和行为模型。数据字典以一种系统化的方式定义在分析模型中出现的数据对象及控制信息的特性,给出它们的准确定义,包括数据流、数据存储、数据项、数据加工,以及数据源点、数据汇点等。数据字典成为将分析模型中的3种模型黏合在一起的“黏合剂”,是分析模型的“核心”。

数据字典中采用的符号如表5-2所示。

表 5-2 数据字典符号

符 号	含 义	示 例
=	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
$m\{\dots\}n$ 或 $\{\dots\}_m^n$	重复	$X=2\{a\}6$ 或 $\{a\}_2^6$ 表示重复 2~6 次 a
$\{\dots\}$	重复	$X=\{a\}$ 表示 X 由 0 个或多个 a 组成
(...)	可选	$X=(a)$ 表示 a 在 X 中可能出现,也可能不出现
"..."	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	$X=1..9$ 表示 X 可取 1~9 中的任意一个值

例如,数据流“应聘者名单”由若干应聘者姓名、性别、年龄、专业和联系电话等信息组成,那么“应聘者名单”可以表示为:应聘者名单={应聘者姓名+性别+年龄+专业+联系电话}。数据项考试成绩可以表示为:考试成绩=0..100。再如,某教务系统的学生成绩库文件的数据字典描述可以表示为以下形式。

文件名:学生成绩库

记录定义:学生成绩=学号+姓名+{课程代码+成绩+[必修|选修]}

学号:由6位数字组成

姓名:2~5个汉字

课程代码:8位字符串

成绩:1~3位十进制整数

文件组织:以学号为关键字递增排列

5.3.5 加工规格说明

在对数据流图的分解中,位于最底层数据流图的数据处理,也称为基本加工或原子加工,对于每一个基本加工都需要进一步说明,这称为加工规格说明,也称为处理规格说明。在编写基本加工的规格说明时,主要目的是要表达“做什么”,而不是“怎样做”。加工规格说明一般用结构化语言、判定表和判定树来表述。

1. 结构化语言

结构化语言也称为过程设计语言(Procedure Design Language, PDL),也称为伪代码,

在某些情况下,在加工规格说明中会用到。但一般来说,最好将用 PDL 来描述加工规格说明的工作推迟到过程设计阶段进行比较好。PDL 的介绍可参见 6.9.4 节。

2. 判定表

在某些数据处理中,某个数据处理(即加工)的执行可能需要依赖于多个逻辑条件的取值,此时可用判定表。判定表能够清晰地表示复杂的条件组合与应做的动作之间的对应关系。

一张判定表由 5 部分组成,左上部列出所有条件,左下部是所有可能做的动作,右上部是表示各种条件组合的矩阵,右下部是和每种条件组合相对应的动作。判定表右半部的每一列实质上是一条规则,规定了与特定的条件组合相对应的动作。

下面以某工厂生产的奖励的算法为例说明判定表的组织方法。某工厂生产两种产品 A 和 B。凡工人每月的实际生产量超过计划指标者均有奖励。奖励政策为:

对于产品 A 的生产者,超产数 n 小于或等于 100 件时,每超产 1 件奖励 2 元; n 大于 100 件小于等于 150 件时,大于 100 件的部分每件奖励 2.5 元,其余的每件奖励金额不变; n 大于 150 件时,超过 150 件的部分每件奖励 3 元,其余按超产 150 件以内的方案处理。

对于产品 B 的生产者,超产数 n 小于或等于 50 件时,每超产 1 件奖励 3 元; n 大于 50 件小于等于 100 件时,大于 50 件的部分每件奖励 4 元,其余的每件奖励金额不变; n 大于 100 件时,超过 100 件的部分每件奖励 5 元,其余按超产 100 件以内的方案处理。

此处理功能的判定表如图 5-17 所示。

决策规则号		1	2	3	5	5	6
条件	产品A	Y	Y	Y	N	N	N
	产品B	N	N	N	Y	Y	Y
	$n \leq 50$	Y	N	N	Y	N	N
	$50 < n \leq 100$	Y	N	N	N	Y	N
	$100 < n \leq 150$	N	Y	N	N	N	Y
	$n > 150$	N	N	Y	N	N	Y
奖励政策	$2 \times n$	√					
	$2.5 \times (n - 100) + 200$		√				
	$3 \times (n - 150) + 325$			√			
	$3 \times n$				√		
	$4 \times (n - 50) + 150$					√	
	$5 \times (n - 100) + 350$						√

图 5-17 此处理功能的判定表

从上面这个例子可以看出,判定表能够简洁又无歧义地描述处理规则。当把判定表和布尔代数或卡诺图结合起来使用时,可以对判定表进行校验或简化。判定表并不适于作为一种通用的工具,没有一种简单的方法使它能同时清晰地表示顺序和重复等处理特性。

判定表也可用在结构化设计中。

3. 判定树

判定表虽然能清晰地表示复杂的条件组合与应做的动作之间的对应关系,但其含义却不是一眼就能看出来的,初次接触这种工具的人要理解它需要一个简短的学习过程。此外,当数据元素的值多于两个时,判定表的简洁程度也将下降。

判定树是判定表的变种,也能清晰地表示复杂的条件组合与应做的动作之间的对应关系。判定树也是用来表述加工规格说明的一种工具。判定树的优点在于,它的形式简单到不需任何说明,让人一眼就可以看出其含义,因此易于掌握和使用。多年来判定树一直受到人们的重视,是一种比较常用的系统分析和设计的工具。图 5-18 所示为与图 5-17 等价的判定树。从图 5-18 可以看出,虽然判定树比判定表更直观,但简洁性却不如判定表,数据元素的同一个值往往要重复写多遍,而且越接近树的叶端重复次数越多。此外还可以看出,画判定树时分枝的次序可能对最终画出的判定树的简洁程度有较大影响。显然判定表并不存在这样的问题。

判定树也可用在结构化设计中。

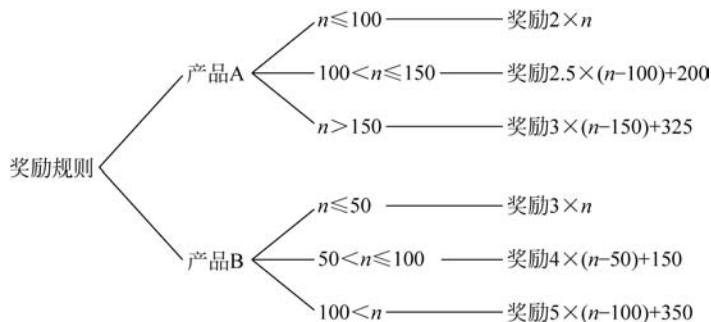


图 5-18 用判定树表示此处理功能的算法

5.4 结构化分析的图形工具

除了前述所用的数据流图、E-R 图、状态图、数据字典和加工规格说明(结构化语言、判定表和判定树)外,在结构化的分析中,有时还会用到层次方框图、Warnier 图和 IPO 图这 3 种图形工具。

5.4.1 层次方框图

层次方框图由树形结构的一系列多层次的矩形组成,用来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,它表示数据结构的整体。下面的各层矩形框表示这个数据的子集,最底层的各个框表示这个数据的不能再分割的元素。这里需要提醒的是,层次方框图不是功能模块图,方框之间的关系是组成关系,而不是调用关系。

图 5-19 为电子相册管理系统组成的层次方框图。

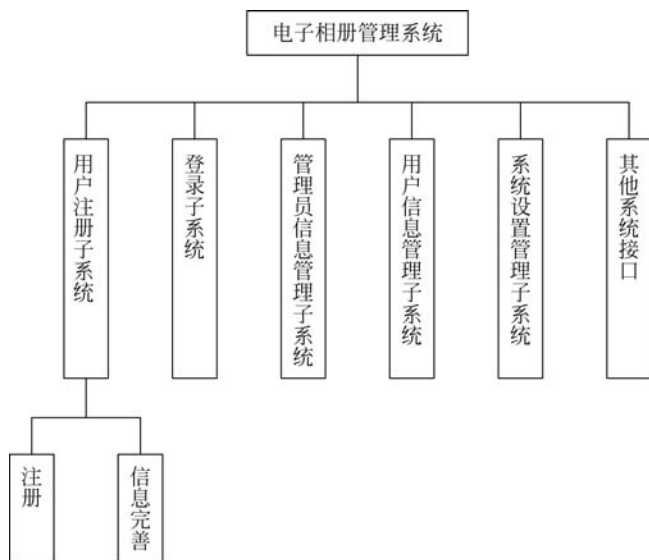


图 5-19 电子相册管理系统组成的层次方框图

5.4.2 Warnier 图

Warnier 图是表示数据层次结构的另一种图形工具,它与层次方框图相似,也用树形结构来描绘数据结构。Warnier 图提供了比层次方框图更详细的描绘手段,能指出某一类数据或某一数据元素重复出现的次数,并能指明某一特定数据在某一类数据中是否有条件地出现。

Warnier 图使用如下的几种符号。

- (1) 花括号用来区分信息的层次,在一个花括号中的所有名字都属于一类信息。
- (2) 异域符号 $\oplus$ 表明一类信息或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信息类(或元素)在这个数据结构中出现的次数。
- (3) 圆括号“()”中的数字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。

软件产品的组成就可用 Warnier 图描述,如图 5-20 所示。

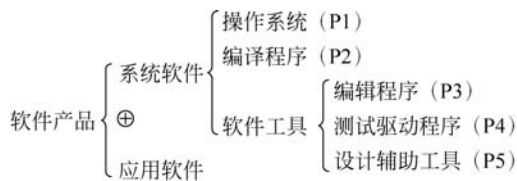


图 5-20 软件产品组成的 Warnier 图

5.4.3 IPO 图

IPO 图是输入/处理/输出图的简称,它是 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,因此这种工具很容易掌握和使用。它的基本形式是在左边的框中列出有关的输入,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。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。在 IPO 图中用空心大箭头指出数据通信的情况。图 5-21 所示为一个主文件更新的 IPO 图。



图 5-21 主文件更新的 IPO 图

一种改进的模块 IPO 图的形式如图 5-22 所示,这种图除了描述输入、处理、输出过程外,还包括了某些附加的信息,这些附加的信息非常有利于理解系统及对该模块的实现,它们包括系统名称、设计人员、设计日期、模块名称、模块编号,还包括调用本模块的模块清单、本模块调用的模块清单以及全局的、局部的数据变量等。

改进的模块IPO图的形式	
系统模块IPO图设计样式	
系统名称:	
模块名称:	模块编号:
设计人:	设计日期:
模块描述:	
被调用模块:	调用模块:
输入参数:	输入说明:
输出参数:	输出说明:
变量说明:	
处理说明:	
备用:	

图 5-22 一种改进的模块 IPO 图

IPO 图也可用在结构化设计中。

尽管使用结构化方法建模具有一定的优势,但它还是有以下的局限性。

- (1) 不提供对非功能需求的有效理解和建模。
- (2) 不提供对用户选择合适方法的指导,也没有对方法适用的特殊环境的忠告。
- (3) 往往产生大量文档,系统需求的要素被隐藏在一大堆具体细节的描述中。
- (4) 产生的模型不注意细节,用户总觉得难以理解,因而很难验证模型的真实性。

5.5 结构化分析实例

【例 5-1】 某培训机构入学管理系统有报名、缴费、就读等多项功能,并有课程表(课程号,课程名,收费标准)、学员登记表(学员号,姓名,电话)、学员选课表(学员号,课程号,班级

号)、账目表(学员号,收费金额)等诸多数据表。

下面是对其各项功能的说明。

(1) 报名: 由报名处负责,需要在学员登记表上进行报名登记,需要查询课程表让学员选报课程,学员所报课程将记录到学员选课表。

(2) 交费: 由收费处负责,需要根据学员所报课程的收费标准进行收费,然后在账目表上记账,并打印收款收据给办理交费的学员。

(3) 就读: 由培训处负责,其在验证学员收款收据后,根据学员所报课程将学员安排到合适班级就读。

请用结构化方法画出入学管理系统顶层图、1层图,并写出其数据字典。

【解析】

(1) 对于一个培训机构,外部用户主要有非学员、学员、工作人员。非学员通过报名成为学员。学员只有缴费,才可上课。工作人员需要登记学员、收费以及安排学员就读,根据以上分析得到顶层图(见图 5-23)。

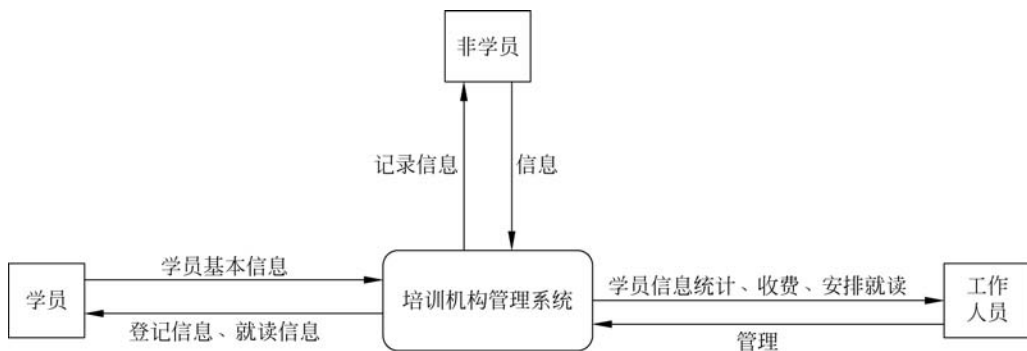


图 5-23 该系统顶层图

(2) 一个非学员通过报名成为学员,他需要将个人信息提供给报名处,报名处负责记录信息,并通过查询课程表提供学员选课信息;学员选课;并将学员选课信息记录在学员选课表。报名 1 层图如图 5-24 所示。

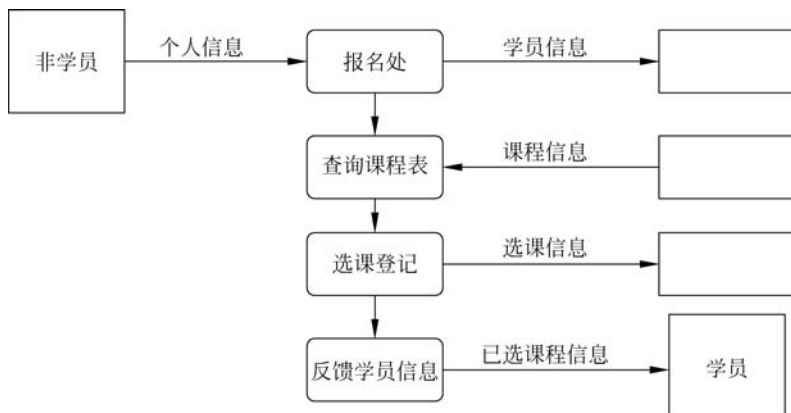


图 5-24 报名 1 层图

(3) 学员将学员号报给收费处。收费处通过查询选课表获取课程信息,通过查询课程表查询应收金额,并将信息记录在账目表,最后向学员收费并打印账目信息。收费 1 层图如图 5-25 所示。

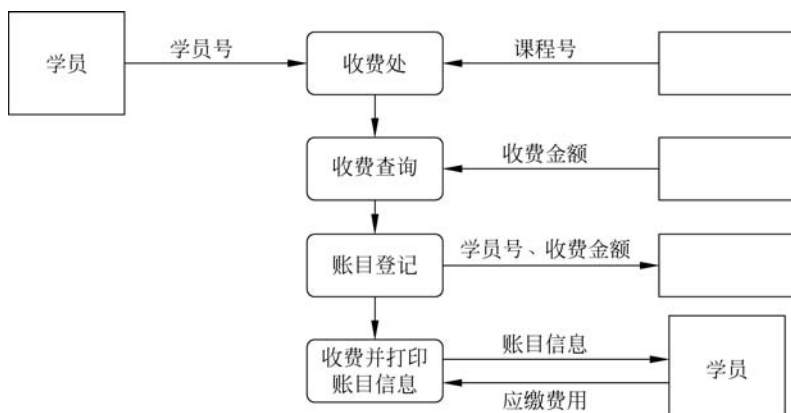


图 5-25 收费 0 层图

(4) 学员向培训处提供缴费凭证。培训处验证好学员缴费凭证后,应通过查询选课表提供学员班级号,分配其到指定班级上课。就读 1 层图如图 5-26 所示。

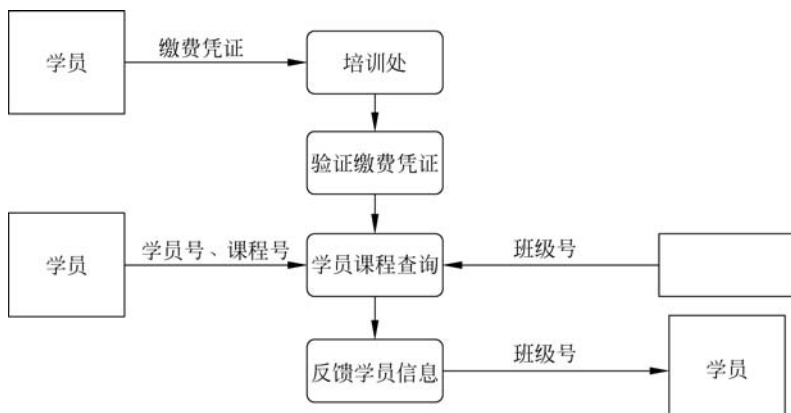


图 5-26 就读 1 层图

入学管理系统的数据字典如下。

(1) 顶层图数据字典: 非学员 = {姓名 + 电话}

学员 = {学员号 + 姓名 + 电话}

信息 = {姓名 + 电话}

学员基本信息 = {学员 + 姓名 + 电话}

工作人员 = {姓名 + 工作人员代号}

姓名: 2{汉字}4

工作人员代号: 4{数字}4

登记信息 = {学员号 + 姓名 + 电话}

就读信息 = {学员号 + 课程号 + 班级号}

- (2) 报名 1 层图数据字典：非学员 = {姓名 + 电话}
学员信息 = {学员号 + 姓名 + 电话}
课程信息 = {课程号 + 课程名 + 收费标准}
选课信息 = {学员号 + 课程号}
学员 = {学员号 + 姓名 + 电话}
学员号：6{数字}6
姓名：2{汉字}4
电话：11{数字}11
收费金额：2{数字}4
课程号：4{数字}4
- (3) 收费 1 层图数据字典：学员 = {学员号 + 姓名 + 电话}
课程号：4{数字}4
收费金额：2{数字}4
学员号：6{数字}6
账目信息 = {学员号 + 收费金额}
应缴费用：2{数字}4
- (4) 就读 1 层图数据字典：学员 = {学员号 + 姓名 + 电话}
学员号：6{数字}6
课程号：4{数字}4
班级号：3{数字}3

习题

1. 选择题

- (1) 在需求分析之前有必要进行()工作。
A. 程序设计
B. 可行性研究
C. E-R 分析
D. 行为建模
- (2) 需求分析是一个(), 它应该贯穿于系统的整个生命周期中, 而不是仅仅属于软件生命周期早期的一项工作。
A. 概念
B. 工具
C. 方法
D. 过程
- (3) 软件需求规格说明书的内容不应该包括()。
A. 对重要功能的描述
B. 对算法的详细过程描述
C. 对数据的要求
D. 软件的性能
- (4) 软件需求分析阶段的工作, 可以分为以下 5 个方面: 对问题的识别、分析、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
- (5) 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. 燃尽图
C. 状态转换图
D. 数据词典

- (6) 数据流图是进行软件需求分析的常用图形工具,其基本图形符号是()。
- A. 输入、输出、外部实体和加工 B. 变换、加工、数据流和存储
- C. 加工、数据流、数据存储和外部实体 D. 变换、数据存储、加工和数据流
- (7) 结构化分析方法的基本思想是()。
- A. 自底向上逐步分解 B. 自顶向下逐步分解
- C. 自底向上逐步抽象 D. 自顶向下逐步抽象
- (8) 在 E-R 图中,包含以下基本成分()。
- A. 数据、对象、实体 B. 控制、关系、对象
- C. 实体、关系、控制 D. 实体、属性、关系

2. 判断题

- (1) 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。 ()
- (2) 需求分析是开发方的工作,用户的参与度不大。 ()
- (3) 需求规格说明书在软件开发中具有重要的作用,它也可以作为软件可行性研究的依据。 ()
- (4) 需求分析的主要目的是解决软件开发的具体方案。 ()
- (5) 需求规格说明书描述了系统每个功能的具体实现。 ()
- (6) 非功能需求是从各个角度对系统的约束和限制,反映了应用对软件系统质量和特性的额外要求。 ()
- (7) 需求分析阶段的成果主要是需求规格说明书,但该成果与软件设计、编码、测试直至维护关系不大。 ()
- (8) 分层的 DFD 图可用于可行性研究阶段,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) 在需求分析阶段主要采用图形工具来描述的原因是图形的信息量大,便于描述规模大的软件系统。 ()
- (11) 设计数据流图时只需考虑系统必须完成的基本逻辑功能,完全不需考虑怎样具体地实现这些功能。 ()

3. 简答题

- (1) 如何理解需求分析的作用和重要性?
- (2) 常用的需求获取的方法有哪些? 对比各种方法的优缺点。
- (3) 如何理解结构化需求分析方法的基本思想?
- (4) 请简述数据流图的作用。
- (5) 请简述数据字典的作用。
- (6) 请简述 E-R 图的作用。
- (7) 请简述状态图的作用。

4. 应用题

- (1) 某图书管理系统有以下功能。

① 借书：输入读者借书证号。系统首先检查借书证是否有效，若有效，对于第一次借书的读者，在借书文件上建立档案。否则，查阅借书文件，检查该读者所借图书是否超过 10 本，若已达 10 本，拒借，未达 10 本，办理借书（检查该读者目录并将借书情况录入借书文件）。

② 还书：从借书文件中读出与读者有关的记录，查阅所借日期，如果超期（3 个月）作罚款处理；否则，修改库存目录与借书文件。

③ 查询：可通过借书文件、库存目录文件查询读者情况、图书借阅情况及库存情况，打印各种统计表。

用结构化分析方法画出系统顶层图、1 层图（数据流图），并写出数据字典。

（2）根据以下描述画出相应的状态转换图。

到 ATM 机前插入银行卡后输入密码，如果密码不正确则系统会要求再次输入密码，如三次输入不正确则退出服务；密码正确后，系统会提示选择服务类型，如选择存款则进行存款操作，存款完毕后可选择继续服务，也可以选择退出服务；如选择取款则进行取款操作，取款完毕后可选择继续服务，也可以选择退出服务。

（3）某企业集团有若干工厂，每个工厂生产多种产品，且每一种产品可以在多个工厂生产，每个工厂按照固定的计划数量生产产品，计划数量不低于 300 件；每个工厂聘用多名职工，且每名职工只能在一个工厂工作，工厂聘用职工有聘期和工资。工厂的属性有工厂编号、厂名、地址，产品的属性有产品编号、产品名、规格，职工的属性有职工号、姓名、技术等级。请画出 E-R 图。