

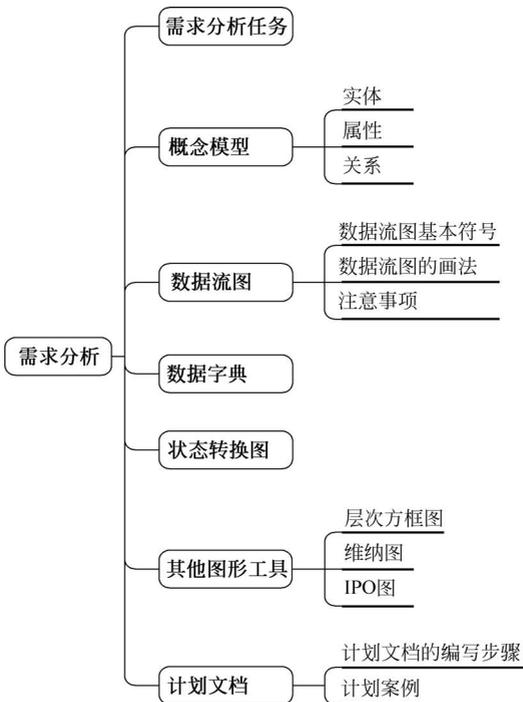


软件需求分析是软件开发过程中的重要阶段,是软件生命周期非常重要的一步,是关系到软件开发成败的关键步骤。它在问题定义和可行性研究阶段之后进行。它的基本任务是准确地回答“系统必须做什么?”这个问题。虽然在可行性研究阶段粗略了解了用户的需求,甚至还提出了一些可行的方案,但是可行性研究的基本目的是用最小的代价在尽可能短的时间内确定问题是否存在可行解,因此许多细节易被忽略,一个微小的疏漏都可能导致误解或铸成系统的严重错误,在纠正时将付出巨大的代价。因而可行性研究并不能代替需求分析,它实际上并没有准确地回答“系统必须做什么”这个关键问题。

需求分析并不是确定系统应该怎样完成它的工作,而是确定系统必须完成的工作有哪些。在需求分析结束之前,系统分析员应准确地描述软件需求,写出软件需求规格说明书。以书面形式准确地描述软件需求。在此过程中,分析员和用户都起着关键的、必不可少的作用。



本章思维导图



3.1 需求分析任务

3.1.1 确定具体要求

需求分析阶段要确定系统的具体要求,下面说明需求分析阶段的具体任务。

1. 确定系统运行环境

确定系统运行环境指的是系统部署和运行所需要的硬件、软件、网络等基础设施和环境条件。

(1) 确定系统所需的硬件环境:包括确定系统运行所需的服务器配置、客户端设备要求等。

(2) 确定系统所需的软件环境:包括确定系统所需的操作系统、数据库管理系统、开发工具等。

(3) 确定系统的网络环境:包括确定系统需要支持的网络协议、网络带宽要求等。

(4) 确定系统的安全环境:包括确定系统所需的安全控制措施、数据加密要求等。

(5) 确定系统的可扩展性和可维护性:包括确定系统需要支持的用户数量、数据量等,并考虑系统未来的扩展和维护需求。

2. 确定系统的具体功能

确定系统必须具备的所有功能,通过需求分析划分出系统必须完成的所有功能。

(1) 收集用户需求:通过与用户、利益相关者和领域专家的沟通,收集用户的需求和期望。

(2) 分析和整理需求:对收集到的需求进行整理和分析,将其分类为功能性需求(系统需要具备的具体功能)和非功能性需求(如性能、安全性、可靠性等要求)。

(3) 确定系统功能:基于用户需求和分析结果,确定系统需要具备的具体功能。这些功能应当清晰、可测量,并且能够满足用户需求。

3. 确定系统的可靠性与可用性

(1) 可靠性:可靠性指的是系统在特定条件下提供准确的和一致的结果的能力。一个可靠的系统应该能在各种情况下正常运行,不会发生意外的崩溃或错误。可靠性可以通过系统的稳定性、容错性、恢复能力等方面来衡量。

(2) 可用性:可用性指的是系统需要时可用的时间比例。一个具有高可用性的系统应能够保证在用户需要时可以正常运行,不会因为不可预见的故障或维护而导致系统不可用。

4. 确定接口需求

接口需求主要用于定义系统与外部环境或其他系统之间的交互方式和规范。常见的接口需求有用户接口需求,软件接口需求,硬件接口需求,通信接口需求等。

5. 其他可能被提出的要求

应该明确列出不在当前系统开发范畴内,但是未来可能出现的要求。这样在被需要时更容易增加需求和修改需求。



3.1.2 建立逻辑模型

需求分析实际上就是建立系统逻辑模型的活动。

模型是为了完全了解事务而对事务做出的一种抽象活动,模型以一种简洁、准确、结构清晰的方式系统地描述了软件需求,可以帮助软件工程师更好地理解系统的结构和行为,从而进行系统设计、分析和测试。使得需求分析任务更容易实现,结果更系统化,同时易于发现用户描述中的模糊性和不一致性。模型分为三种,即数据模型、功能模型和行为模型。

数据模型:描述了数据的结构和关系,建立数据模型是为了理解和表示问题的信息域。

功能模型:描述系统的功能和用例,建立功能模型是为了定义软件有何功能。

行为模型:描述系统中各个组成部分的交互和行为,建立行为模型是为了表示软件的行为。

为了实现上述目标用三种不同的图像,通过数据字典进行描述。数据字典用来描述软件使用或者软件产生的所有实体。数据模型用实体-关系图来描述实体与实体之间的关系;功能模型用数据流图来进行描述;行为模型可以用状态转换图来描述系统的各种行为模式(状态)和不同状态之间的转换。以下是建立模型的基本目标。

(1) 描述用户的需求:通过与用户进行交流,了解用户所需要的功能。

(2) 为软件设计奠定基础:有效的模型可以为软件开发提供正确的方向,进一步保证开发的顺利进行。

(3) 定义需求以便于验收软件产品:对需求的定义可以在验收时用于验证软件是否达成预定功能,从而便于验收软件产品。

3.1.3 需求规格说明

需求分析阶段除了建立模型以外,还必须写出需求规格说明书。软件需求规格说明书附有可执行原型和初步的用户手册。

软件需求规格说明书是分析任务的最终产物,通过建立完整的信息描述、详细的功能和行为描述、性能需求和设计约束的说明、合适的验收标准,给出对目标软件的各种需求。

其中,最知名的标准是 IEEE/ANSI 830—1993 标准,表 3.1 是软件需求规格说明书的框架。

表 3.1 软件需求规格说明书的框架

I. 引言	A. 系统参考文献
	B. 整体描述
	C. 软件项目描述
II. 信息描述	A. 信息内容表示
	B. 数据流表示: 1. 数据流; 2. 控制流
III. 功能描述	A. 功能划分
	B. 功能描述: 1. 处理说明; 2. 限制/局限; 3. 性能需求; 4. 设计约束; 5. 支撑图
	C. 控制描述: 1. 控制规格说明; 2. 设计约束
IV. 行为描述	A. 系统状态
	B. 事件和响应

续表

V. 检验标准	A. 性能范围
	B. 测试种类
	C. 预期的软件响应
	D. 特殊考虑
VI. 参考书目	
VII. 附录	

3.1.4 修正开发计划

修正开发计划是软件工程中 对原始开发计划进行调整和修改的过程。这可能是由于发现了新的需求、技术限制、资源限制或其他外部因素导致的变化。修正开发计划通常需要对项目范围、时间表、预算和资源分配等方面进行调整,以确保项目能够按时交付并满足客户需求。修正开发计划是软件工程中常见的实践,可以帮助项目团队应对变化和挑战,以确保项目成功完成。

3.1.5 制订测试计划

为了验证所开发系统能否满足用户的需求,必须对系统的功能进行测试。在系统开发初期就制订相应的测试计划有利于明确设计目标,保证设计的正确性。

测试计划包括测试的范围、测试的目标、测试的策略和方法、测试资源的分配、测试进度安排、风险评估以及测试的标准和评估方式等内容。制订测试计划是为了确保软件项目能够按照既定的质量标准进行测试,以验证软件是否符合用户需求和设计规范,同时也帮助项目团队在测试过程中更好地管理和控制测试活动。

软件测试计划描述了测试的方法,测试活动的范围、资源和进度。它规定了被测试的项目、特性、应该完成的任务等。

项目在确认交付时,根据之前制定的软件测试计划进行验收,所以测试计划应该得到用户的同意。

3.1.6 编写用户手册

在系统的需求分析阶段可以编写初步的用户手册,在以后的各个软件开发阶段逐步对其进行改进和完善。

1. 引言

- (1) 编写目的: 阐明编写用户手册的目的,指明读者对象。
- (2) 项目背景: 说明项目来源、委托单位开发单位及主管部门。
- (3) 定义: 列出用户手册中使用的专门术语的定义和缩写词的原意。

(4) 参考资料: 列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,可包括 a. 项目的计划任务书、合同或批文; b. 项目开发计划; c. 需求规格说明书; d. 概要设计说明书; e. 详细设计说明书; f. 测试计划; g. 手册中引用的其他资料、采用的软件工程标准或软件工程规范。

2. 软件概述

- (1) 目标: 介绍软件要实现的目标。

(2) 功能：介绍软件详细的功能。

(3) 性能：介绍软件的性能，包括 a. 数据精确度。如输入、输出及处理数据的精度；b. 时间特性。如响应时间、处理时间、数据传输时间等；c. 灵活性。在操作方式、运行环境需做某些变更时软件的适应能力。

3. 运行环境

(1) 硬件：列出软件系统运行时所需的硬件最小配置，包括 a. 计算机型号、主存容量；b. 外存储器、媒体、记录格式、设备型号及数量；c. 输入、输出设备；d. 数据传输设备及数据转换设备的型号及数量。

(2) 支持软件：包括 a. 操作系统名称及版本号；b. 语言编译系统或汇编系统的名称及版本号；c. 数据库管理系统的名称及版本号；d. 其他必要的支持软件。

4. 使用说明

1) 安装和初始化

给出程序的存储形式操作命令、反馈信息及其含意、表明安装完成的测试实例以及安装所需的软件工具等。

2) 输入

给出输入数据或参数的要求。

3) 数据的逻辑描述

可以将数据分为静态数据和动态数据。静态数据是指系统运行过程中主要作为参考的数据，它们一般不随系统的运行而改变，在很长的一段时间内不会变化。动态数据包括所有在系统运行时要发生变化的数据和系统运行时要输入输出的数据。

(1) 静态数据。

(2) 动态输出数据。

(3) 动态输入数据。

(4) 内部生成数据。

(5) 数据约定：对数据的限定，包括但不限于容量、数据的最大值和最小值。对于在设计和开发中的限制更要明确提出。

4) 数据的收集

(1) 要求和范围：输入数据的来源；输出设备的形式；数据的范围，当数据为非数字量时要给每一种合法值的形式和含义；更新处理的频率。

(2) 输入的承担者。

(3) 预处理。

(4) 影响：要说明数据要求对于设备，软件，用户和开发单位可能产生的影响等。

【例 3.1】 为开发一款名为“学生信息管理系统(SIMS)”的软件所编写的一个初步的用户手册。

解析如下。

用户手册主要包括如下内容。

1. 引言

编写目的：本用户手册旨在指导使用 SIMS 的教职员工和学生，以确保他们正确理解和使用系统功能。

项目背景：SIMS 由教育部委托开发单位“软件工程有限公司”开发，主管部门为各个教育机构。

SIMS：学生信息管理系统；

教职员工：指学校教师、工作人员等；

学生：在校学生。

参考资料：项目的计划任务书、项目开发计划、需求规格说明书、概要设计说明书、详细设计说明书、测试计划、其他相关的软件工程标准或规范。

2. 软件概述

目标：SIMS 旨在实现学校、教职员工和学生之间信息交流、记录和管理。

功能：SIMS 包括学生档案管理、成绩录入、考勤管理、课程安排、通知发布等功能。

性能：数据精确度。保证学生成绩等数据的准确性。

时间特性：系统响应时间应在 1 秒内。

灵活性：能够在不同运行环境下正常运行。

3. 运行环境

1) 硬件

计算机配置：至少 4GB 内存，双核处理器。

外存储器：至少 100GB 可用空间。

输入设备：键盘、鼠标。

输出设备：显示器、打印机。

2) 支持软件

操作系统：Windows 10 或更高版本。

数据库管理系统：MySQL 8.0。

4. 使用说明

1) 安装和初始化

(1) 将 SIMS 安装包解压至本地目录。

(2) 运行安装程序，按照提示完成安装。

(3) 输入数据库连接信息，进行初始化设置。

2) 输入

(1) 添加学生信息：姓名、学号、班级等。

(2) 录入成绩：选择课程、输入成绩等。

5. 数据的逻辑描述

静态数据：学生档案信息、课程信息、教职工信息。

动态输出数据：考试成绩单、课程表、动态输入数据、学生成绩录入、请假信息录入。

内部生成数据：学期课程统计数据、考试分数排名数据。

数据约定：学号为 8 位数字；成绩为 0~100 之间的数字。

6. 数据的收集

1) 要求和范围

输入数据的来源：学校教务处、教师。

输出设备的形式：电子文档、打印格式。

数据范围：成绩、考勤、学生信息。

更新处理的频率：每学期末更新一次。

2) 输入的承担者

教师：录入成绩、考勤。

学生：个人信息更新。

3) 预处理

数据输入前进行格式验证和逻辑检查。

4) 影响

数据处理可能影响系统性能和用户体验。

以上是初步的用户手册范例，在软件开发各阶段可逐步改进和完善用户手册内容以满足实际需求。

3.1.7 需求复审

需求复审是由系统分析员和用户一起对需求分析结果进行严格的审查，以确保软件需求的一致性、完整性和正确性，并且对可能存在的问题进行修正和改进。通过需求复审，可以有效地减少需求变更和开发中的错误，提高软件开发的效率和质量。

系统分析员得到的实体-关系图、详细的数据流图、数据字典、状态转换图和一些简明的算法描述准确吗？完整吗？有没有遗漏必要的处理或数据元素？数据元素从何而来？如何处理？正确吗？……这一切都必须有确切的答案，而这些答案只能来自于系统用户。因而，必须请用户对需求分析仔细复查。

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

需求分析阶段结束时应提供的文档有修正后的项目开发计划、软件需求规格说明书、实体-关系图、详细的数据流图、数据字典、状态转换图和一些简明的算法描述、数据要求说明书、初步的测试计划、初步的用户手册等。

3.2 概念模型



为了理解和表示问题域的信息，需要建立概念模型。概念模型是现实世界中事物内部及事物之间的联系。在信息世界中反映为同一实体集中各个实体内部的联系和不同实体集的各个实体之间的联系。概念模型可以用实体-关系图来描述。实体-关系图简称 E-R 图，是一种用于描述数据之间关系的图形化工具，主要用于分析和设计系统中的数据结构和数据之间的关系。软件工程中的实体关系包括以下要素。

实体。现实世界中客观存在的事物。如汽车、面包。

实体集。具有相同性质的同类实体的集合为实体集。如所有的汽车是实体集。

属性是实体的特性,每个实体会会有多个属性。如学生实体有姓名、学号、性别、年龄、所在班级等属性。

实体标识符(也称键)。能唯一表示该实体的属性或者属性集称为实体标识符。如学生实体中的学号就是实体标识符。

实体联系。实体集间的对应关系。

关键词。用来区分不同实体的关键属性。如学号是学生的关键属性。

3.2.1 数据对象(实体)

数据对象是对软件必须理解的复合信息的抽象。所谓复合信息是指具有一系列不同性质或属性的事物,仅有单个值的事物(例如宽度)不是数据对象。

数据对象可以是外部实体(例如产生或使用信息的任何事物)、事物(例如报表)、行为(例如打电话)、事件(例如响警报)、角色(例如教师、学生)、单位(例如会计科)、地点(例如仓库)或结构(例如文件)等。总之,可以由一组属性来定义的实体都可以被认为是数据对象。

3.2.2 数据对象的属性

属性定义了数据对象的性质。必须把一个或多个属性定义为“标识符”,也就是说,当人们希望找到数据对象的一个实例时,用标识符属性作为“关键字”(通常简称为“键”)。应该根据对所要解决的问题来确定特定数据对象的一组合适的属性。

例如,为了开发机动车管理系统,描述汽车的属性应该是生产厂家、品牌、型号、发动机号码、车体类型、颜色、车主姓名、住址、驾驶证号码、生产日期及购买日期等。

3.2.3 数据对象间的关系

数据对象彼此之间相互连接的方式称为联系,也称为关系。实体间的联系分为两类,一类是同一实体集中各个实体间的联系;一类是不同实体集中各个实体间的联系。而不同实体集中各个实体间的联系又分为以下三类。

1. 一对一联系(1:1)

如果有实体集 A 中的每一个值,实体集 B 中至多有一个值与之联系,反之亦然,则称实体 A 与实体 B 之间是一一对一的联系。

如:班级是一个实体集,正班长也是一个实体集。每个班级有且只能有一个正班长,而一个正班长只能在一个班级任职,这样班级和正班长是一一对应关系。

2. 一对多联系(1:n)

如果有实体集 A 中的每一个值,实体 B 中有 n 个值($n \geq 1$)与之联系;反之,若实体集 B 中的每一个值,实体 A 中至多有一个值与之联系,则称实体 A 与实体 B 之间是一对多的联系。

如:一个班级有若干名学生,一个学生只属于一个班级。

3. 多对多联系($m:n$)

如果有实体集 A 中的每一个值,实体集 B 中有 n 个值($n \geq 1$)与之联系;反之,若实体集 B 中的每一个值,实体 A 中可以有 m 个值($m \geq 1$)与之联系,则称实体 A 与实体 B 之间是多对多的联系。

如：学生与课程之间就是多对多的联系。
两个不同实体集间的三种联系如图 3.1 所示。

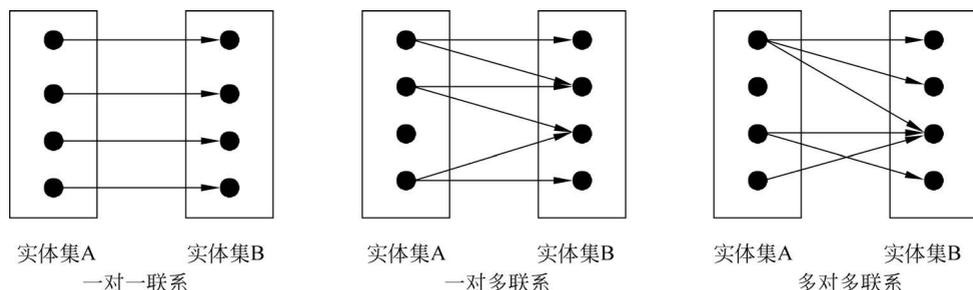


图 3.1 两个不同实体集间的三种联系

3.2.4 实体-关系图

实体-关系(Entity-Relationship, E-R)图,用来描述实体本身的特征及实体间的联系,是一种描述现实世界的概念模型,是一个非常重要的模型。

1. E-R 图的四个基本构成

实体：矩形框表示实体,实体名一般为名词。

属性：椭圆形表示属性,属性名一般为名词。

联系：菱形框表示实体间的联系,联系名一般为动词。

连线：用于连接实体型与连接类型,也可用于表示实体与属性的联系,可注明种类;对关键码的属性,在属性名下划一道横线。

例如,图 3.2 表示学生与课程间的联系(“学”)是多对多的,即一个学生可以学多门课程,而每门课程可以有多个学生来学。

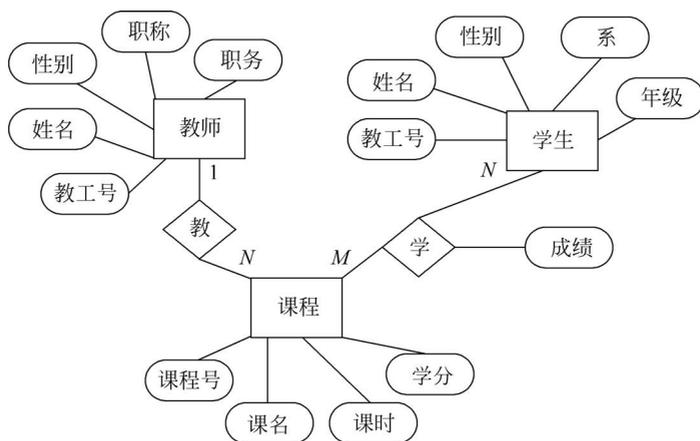


图 3.2 某校教学管理 E-R 图

联系也可能有属性。例如,学生“学”某门课程所取得的成绩,既不是学生的属性也不是课程的属性。由于“成绩”既依赖于某名特定的学生又依赖于某门特定的课程,所以它是学生与课程之间的联系“学”的属性(图 3.2)。

2. 建立 E-R 图的过程

首先确定实体及实体的属性,然后确定联系,再将实体和联系连接起来形成 E-R 图,最后确定实体类型的关键词,在属性名下划一道横线。

例如,学生与院系、课程、教师、班级、专业之间的 E-R 图如图 3.3 所示(属性略);学生实体图如图 3.4 所示。

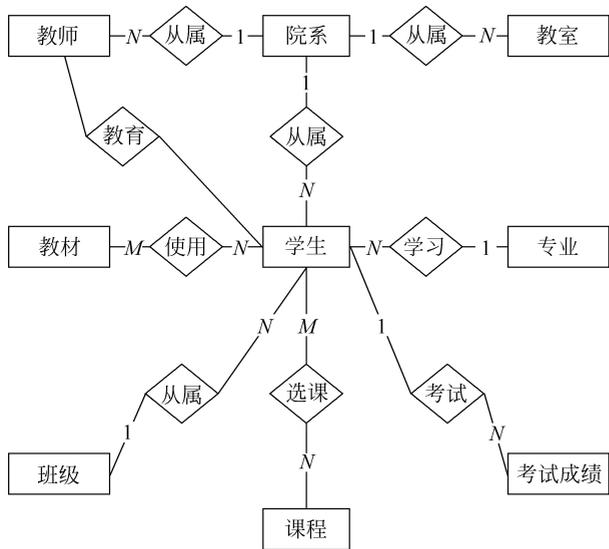


图 3.3 E-R 图

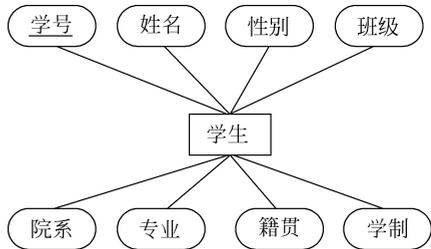


图 3.4 学生实体图

注：由于 E-R 图中实体属性多,因此分为 2 个图举例呈现。



3.3 数据流图

3.3 数据流图

数据流图(Data Flow Diagram,DFD)是用于表示系统逻辑模型的一种工具。它以直观的图形清晰地描述了系统数据的流动和处理过程,图中没有任何具体的物理元素,主要强调的是数据流和处理过程,即使不是计算机专业技术人员也很容易理解。数据流图是软件开发人员和用户之间很好的通信工具。设计数据流图时只需考虑软件系统必须完成的基本逻辑功能,不需要考虑如何具体实现这些功能,它是软件开发的出发点。

3.3.1 数据流图基本符号

数据流图有 4 种基本符号,如图 3.5 所示。长方形表示数据的源点或终点;圆形代表

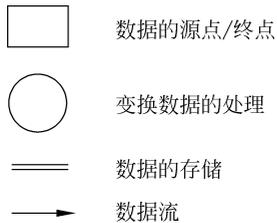


图 3.5 数据流图基本符号

变换数据的处理;两条平行横线代表数据存储;箭头表示数据流,即特定数据的流动方向。注意,数据流与程序流程图中用箭头表示的控制流有本质不同,千万不要混淆。熟悉程序流程图的初学者在画数据流图时,往往试图在数据流图中表现分支条件或循环,殊不知这样做将造成混乱,画不出正确的数据流图。在数据流图中应该描绘所有可能的数据流向,而不应该描绘出现某个数据流的条件。

除上述 4 种基本符号之外,有时也使用几种附加符号,“*”表示数据之间的关系(同时存在);“+”表示“或”关系,“⊕”表示只能从中选一个(互斥的关系)。

3.3.2 数据流图的画法

数据流图是为了让用户明确系统之中数据流动和处理的情况,即系统的基本逻辑功能。对于一个系统来说,数据流图的表示方法并不唯一,比较好的方法是分层次的描述系统。顶层数据流图描述系统的总体,表明了系统的关键功能,然后分别对每一个关键功能进行适当的详细描述。分层次的描述便于用户了解系统。

1. 画数据流图的基本原则

- (1) 数据流图中所有的符号必须是前面所述的 4 种基本符号和附加符号。
- (2) 数据流图的主图(顶层)必须含有前面所述的 4 种符号,缺一不可。
- (3) 数据流图主图上数据流必须封闭在外部实体之间(外部实体可以是一个,也可以是多个)。
- (4) 加工(变换数据处理)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) 任何一个数据流子图必须与它父图上的一个加工相对应,父图中有几个加工,就可能有几张子图,两者的输入数据流和输出数据流必须一致,即所谓“平衡”。
- (6) 数据流图中的每个元素都必须有名字(流向数据存储或从数据存储流出的数据流除外)。

2. 画数据流图的步骤

先画数据流图的主图,大致可分为以下几步。

第一步,先找外部实体(可以是人、物或其他软件系统),找到了外部实体,则系统与外部世界的界面就得以确定,系统的源点和终点也就找到了;

第二步,找出外部实体的输入和输出数据流;

第三步,在图的边上画出系统的外部实体;

第四步,从外部实体的输出流(源点)出发,按照系统的逻辑需要,逐步画出一系列变换数据的处理,直到找到外部实体处所需的输入流(终点),形成数据流的封闭;

第五步,按照上述原则进行检查和修改,最后按照上述步骤画出所有子图。

【例 3.2】 飞机票预订系统的数据流图。

飞机票预订系统的数据流图见图 3.6。

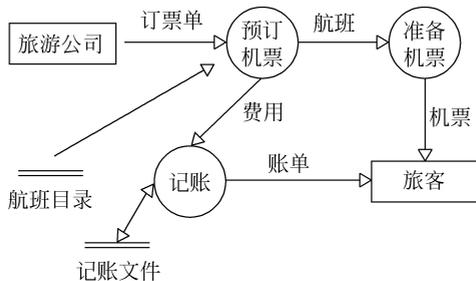


图 3.6 飞机票预订系统的数据流图

说明：“→”箭头，表示数据流；“○”圆或椭圆，表示变换数据的处理；“□”矩形框，表示数据的源点或终点；“=”双杠或单杠，表示数据存储(文件)。

3.3.3 注意事项

(1) 变换数据的处理不一定是一个程序。一个处理框可以代表一个程序或者一个模块，也可以代表一个处理过程。

(2) 一个数据存储不一定是一个文件。它可以表示一个文件或者一个数据项，数据可以存储在任何介质上。

(3) 数据存储和数据流都是数据，只是所处的状态不同。数据存储是静止的数据，数据流是运动的数据。

(4) 分层次地画数据流图。据研究表明，若一张数据流图中包含的处理多于9个时，人们很难领会它的含义，所以此时数据流图应该分层绘制。将复杂的功能分解为子功能来细化数据流图有利于人们理解其含义，所以数据流图可分为高层总体的数据流图和多张细化的数据流子图。

(5) 数据流图的细化原则。数据流图分层细化之时必须保持信息的连续性，即细化前后对应功能的输入输出数据必须相同。当一个功能细化为子功能需要写出程序代码时，就不应该进行细化。

(6) 画数据流图时，只考虑数据流的静态关系，不考虑其动态关系(如启动、停止等与时间有关的问题)，也不考虑出错处理问题。

(7) 画数据流图时，只考虑常规状态，不考虑异常状态，这两点一般留在设计阶段解决。

(8) 画数据流图不是画程序流程图，二者有本质的区别。数据流图只描述“做什么”，不描述“怎么做”和做的顺序，而程序流程图表示对数据进行加工的控制和细节。

(9) 不能期望数据流图一次画成，而是要经过各项反复才能完成。

(10) 描绘复杂系统的数据流图通常很大，对于画在几张纸上的图很难阅读和理解。一个比较好的方法是分层描绘这个系统。在分层细画时，必须保持信息的连续性，父图和子图要平衡，每次只细画一个加工。



3.4 数据字典

3.4 数据字典

数据字典是对数据流图中的数据元素、数据流、文件、处理的定义的集合。数据字典是在软件分析和设计的过程中提供数据描述，通常包含了系统中所有数据元素的定义、描述、属性和关系等信息，它是数据流图中必不可少的辅助资料。数据流图和数据流图中每个元素的确切定义，它们共同才能构成完整的系统规格说明。

数据字典由以下五类条目组成。

1. 数据元素

数据元素是数据的最小组成单位，包含以下内容。

(1) 数据元素的名称：数据元素的名称或标识符，用于唯一地标识该数据元素。

(2) 数据元素的类型：描述数据元素的类型，例如整数、浮点数、字符串、日期等。

(3) 数据元素的长度：描述数据元素的长度或容量限制，特别是对于字符串或数组等

数据类型。

(4) 数据元素的取值范围：描述数据元素允许的取值范围,包括最小值和最大值。

(5) 数据元素的定义：对数据元素的功能、含义和用途进行详细的描述。

2. 数据流

数据流条目用于描述系统中的数据流,包括数据流的名称、描述、类型、来源、去向、处理等信息。

3. 数据存储

数据存储条目用于描述系统中的数据存储,包括数据存储的名称、描述、结构、内容、访问方式等信息。

4. 数据处理

数据处理条目用于描述系统中的数据处理,包括数据处理的名称、描述、输入、输出、处理逻辑等信息。

5. 外部项

数据源等外部实体,表示了系统数据的来源以及去处。外部项越少越好,若过多则系统独立性差,确定的人机界面不合适。

【例 3.3】 学生实体的数据字典。

实体名称：学生(Student)

数据项：

(1) 学生 ID(Student ID)。数据类型：整数；描述：学生的唯一标识符。

(2) 姓名(Name)。数据类型：字符串；描述：学生的姓名。

(3) 年龄(Age)。数据类型：整数；描述：学生的年龄。

(4) 性别(Gender)。数据类型：字符串；描述：学生的性别。

(5) 班级(Class)。数据类型：字符串；描述：学生所在的班级。

(6) 地址(Address)。数据类型：字符串；描述：学生的家庭地址。

3.5 状态转换图

状态通常指的是系统或对象在特定时间点或时间段内所处的特定条件或情况。状态是描述系统、组件或对象的属性和特征的一种方式,它反映了系统或对象在某一时刻的行为、性质或特征。一个状态表示系统的一种行为模式。有时对象在不同的状态下呈现不同的行为方式,所以应该分析对象的状态,才能正确地认识实体的行为并且定义它的操作。

状态转换图是一种用于描述系统状态以及状态之间转换关系的图形化工具。通过状态转换图,可以清晰地描述系统的各种状态以及状态之间的转换关系,帮助开发人员和利益相关者更好地理解系统的行为并进行系统的设计、分析和测试。

并不是所有实体都要画状态转换图,有些实体有意义明确的状态,当其行为在不同的状态下有一定的改变时,才需要画出状态转换图。

1. 状态转换图的画法

画状态转换图的步骤有以下几步。

(1) 确定系统的状态。首先,需要明确系统的所有可能状态。这些状态是系统在某一时刻

间点的特定情况或条件。例如,一个订单处理系统可能有“待处理”“已处理”“已取消”等状态。

(2) 确定状态之间的转换关系。确定系统中状态之间的转换关系,即在某些条件下系统从一个状态转换到另一个状态。这些条件可以是外部事件、用户输入或系统内部条件的改变。需要明确每个状态之间可能的转换条件和触发事件。

(3) 绘制状态转换图。使用图形工具(如 UML 工具、绘图软件等),根据上述确定的状态和状态之间的转换关系,绘制状态转换图。在图中,用节点表示每个状态,用有向边表示状态之间的转换关系,箭头上标注触发状态转换的事件或条件。

(4) 标注状态转换的条件和动作。在状态转换图中,需要标注每个状态转换的条件和可能执行的动作。这些条件和动作描述了系统在状态转换时的行为。

(5) 审查和验证。绘制完成后,需要对状态转换图进行审查和验证,确保图中描述的状态和转换关系符合系统的实际行为,并且没有遗漏或错误。

2. 状态转换图的符号

状态转换图中常用的符号如下。

(1) 椭圆:表示实体的一种状态,椭圆内部填写状态名称。

(2) 箭头:表示从箭头出发的状态可以转换为箭头指向的状态。

(3) 事件:箭头线上可标注引起状态转换的事件名。事件后可以加方括号,方括号内可写状态转换的条件。

(4) 实心圆:指出该实体被创建之后所处的初始状态。

(5) 内部实心的同心圆:表示最终的状态。

【例 3.4】 状态图中使用的主要符号。

状态图中使用的主要符号见图 3.7。

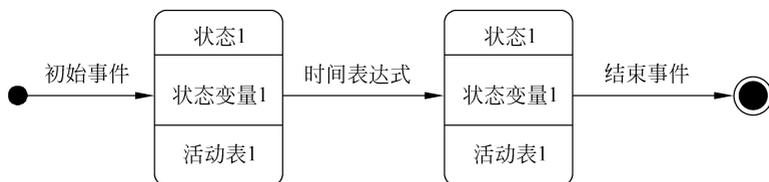


图 3.7 状态图中使用的主要符号



3.6 其他图形工具

3.6 其他图形工具

需求分析阶段,除了使用之前讲过的几种方法之外,还经常使用一些其他的图形工具来描述需求关系和逻辑处理功能,3.6 节将介绍层次方框图、维纳图和 IPO 图。

3.6.1 层次方框图

层次方框图由一系列多层次的树形结构的矩形框组成,用来描述数据的层次结构。层次方框图的顶层是一个单独的矩形框,它代表数据结构的整体,下面各层的矩形框代表这个数据结构的子集,最底层的各个矩形框代表组成这个数据的不能再分割的基本元素。随着结构描述向下层的细化,层次方框图对数据结构的描述也越来越详细,系统分析员从顶层数据开始分类,沿着图中每条路径不断细化,直到确定数据结构的全部细节时为止,这种处理

模式很适合需求分析阶段的需要。但在使用中需要注意,方框之间的联系表示组成关系,不是调用关系,因为每个方框不是模块。

【例 3.5】 一家计算机公司全部产品的数据结构可用层次方框图描述。
全部产品层次方框图见图 3.8。

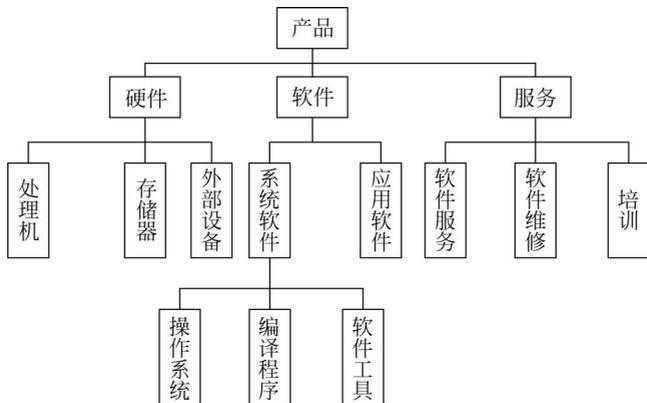


图 3.8 一家计算机公司全部产品层次方框图

3.6.2 维纳图

维纳图是表示信息层次体系的一种图形工具,由法国计算机科学家 J. D. Warnier 提出。维纳图又称 Warnier 图,同层次方框图类似,也可以用来描述树形结构的信息,可以指出一类信息或一个信息是重复出现,也可指明信息是有条件出现的。在维纳图中使用以下几种符号。

- (1) 大括号“{}”用来区分信息的层次;
- (2) 异或符号“⊕”指出一个信息类或一个数据元素在一定条件下出现,符号上、下方的名字代表的信息只能出现一个;
- (3) 圆括号“()”指出这类数据重复出现的次数。

【例 3.6】 描绘一类产品的维纳图。
一类产品的维纳图见图 3.9。

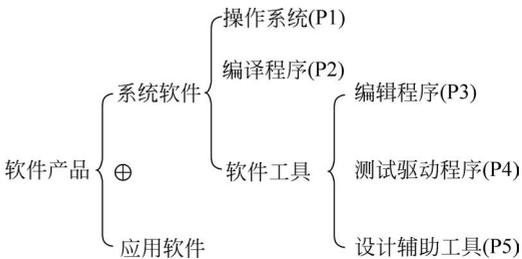


图 3.9 一类产品的维纳图

3.6.3 IPO 图

IPO 图是输入—处理—输出图 (Input-Process-Output) 的简称,是美国 IBM 公司发展完善起来的一种图形工具。IPO 图使用的基本符号少而简单,因此易学易懂。它的基本形式是画 3 个方框,在左边的框中列出有关输入数据,在中间框内列出主要处理,在右边的框内列出产生的输出数据。处理框中列出的处理次序是按执行顺序书写的。但是这些符号还不能精确地描述执行处理的详细情况,在 IPO 图中,还用类似向量符号的空心箭头清楚地指出数据通信的情况。

【例 3.7】 考试成绩管理系统的 IPO 图,通过这个例子可以了解 IPO 图的用法。
考试成绩管理系统的 IPO 图见图 3.10。

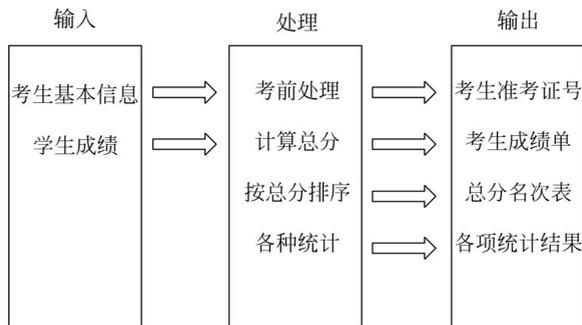


图 3.10 考试成绩管理系统的 IPO 图

3.7 计划文档

3.7.1 计划文档的编写步骤

计划文档的编写步骤如下。

(1) 确定需求：首先需要明确软件工程计划文档的编写目的和范围，包括需要开发的软件功能、性能要求、用户界面设计等方面的需求。

(2) 制订计划：根据需求，制定软件工程计划的时间表和里程碑，确定开发团队的组成和分工，以及开发过程中所需的资源和预算。

(3) 分析风险：对可能出现的风险进行分析和评估，包括技术风险、市场风险和人员风险等，制定相应的风险应对措施。

(4) 制定开发流程：确定软件开发的具体流程和方法，包括需求分析、设计、编码、测试、部署和维护等阶段的工作内容和时间安排。

(5) 编写文档：根据制订的开发计划和流程，编写软件工程计划文档，包括项目背景、范围、目标，开发团队的组织结构和分工，开发过程的具体安排等内容。

(6) 审核和修改：对编写的软件工程计划文档进行内部审核和修改，确保文档的准确性和完整性。

(7) 批准和发布：经过审核和修改后，软件工程计划文档由项目负责人或管理者批准并发布，以便开发团队按照计划进行工作。

3.7.2 计划案例

【案例】 制订演唱会观众管理和信息分析系统的软件开发计划。

某演唱公司要开发一个演唱会观众管理和信息分析系统，若立即开始考虑实现演唱会观众信息管理系统的详细方案并且开始编写程序，这不符合软件工程的开发思想。系统分析员首先要考虑开发这样一个系统能否可行，是否能够产生经济效益，而不是立刻实现它。还要进一步考虑，用户面临的问题有什么。根据上述计划文档编写步骤进行编写。

1. 确定需求

开发一个演唱会观众管理和信息分析系统，包括观众信息录入、座位管理、票务销售、数据分析等功能。

要求系统能够支持大型演唱会的观众信息管理和数据分析,具备安全性和高可靠性。

2. 制订计划

开发周期为 5 个月,包括需求分析、设计、编码、测试和部署等阶段。

开发团队包括 5 名开发工程师、2 名测试工程师和 1 名项目经理。

预算为 60 万元,包括硬件设备、软件工具和人员成本。

3. 分析风险

技术风险:座位管理和大数据分析可能带来开发复杂度。

人员风险:团队成员离职或项目调岗可能影响开发进度。

市场风险:需求变更或竞争对手的影响可能影响项目规划。

4. 制定开发流程

需求分析阶段:1 个月,确定功能需求、用户界面设计和数据库设计。

设计阶段:1 个月,完成系统架构、数据库设计和接口定义。

编码阶段:1.5 个月,进行模块编码和集成测试。

测试阶段:1 个月,进行功能、性能和安全等各项测试。

部署阶段:0.5 个月,上线前准备和用户培训。

5. 编写文档

项目背景:介绍演唱会观众管理和信息分析系统的开发背景和目标。

开发团队:列出各成员的职责和工作安排。

开发流程:详细描述各阶段的工作内容和时间安排。

6. 审核和修改

内部审核人员对文档进行审核,并对文档进行必要的修改和完善。

7. 批准和发布

项目负责人批准并发布软件工程计划文档。

本章总结

可行性研究阶段对用户进行详细的研究调查之后,确定软件开发的系统功能,目标,规模以及所开发软件与其他软件之间的关系。需要从技术方面,经济方面等方面写出可行性研究报告。

需求分析是建立系统逻辑模型的活动,研究用户的需求以便得到系统或者软件需求的过程。功能模型使用状态转移图来进行描述。需求分析阶段需要建立各种模型,除了建立模型之外还应该写出软件需求规格说明书、修改软件开发计划、系统测试计划。

习 题

一、单选题

1. 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
2. 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图 D. 数据字典
3. 软件需求分析一般应确定的是用户对软件的()。
- A. 功能需求 B. 非功能需求
C. 性能需求 D. 功能需求和非功能需求
4. 描述类中某个对象的行为,反映了状态与事件关系的是()。
- A. 对象图 B. 状态图 C. 流程图 D. 结构图
5. 在软件需求分析中,开发人员需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. 要求软件工作效率怎样 D. 要让软件具有何种结构
6. 需求分析阶段最重要的技术文档之一是()。
- A. 项目开发计划 B. 设计说明书
C. 需求规格说明书 D. 可行性分析报告

二、填空题

1. 软件需求分析阶段的目的是(),并把双方共同的理解明确地表达成一份书面文档()。
2. 软件需求分类,分为()需求和()需求。
3. 需求开发的步骤包括()、()、编写规格说明、需求验证。

三、简答题

1. 需求分析阶段的基本任务是什么?
2. 需求分析阶段需要执行哪些活动?



第3章实践活动



实践活动

软件需求分析

——以大学生党员信息管理系统为例

一、背景知识

需求分析是软件开发过程中至关重要的一环,它确保了软件项目的成功和用户满意度。通过仔细分析和理解用户的需求,开发团队能够核实正在构建的软件产品能否满足用户的期望和需求。需求分析有助于识别和定义问题的范围,明确项目的目标,并为整个开发过程奠定坚实的基础。本实验以大学生党员信息管理系统的需求分析报告为样例,介绍需求分析的基本书写流程。

二、实验目的

1. 掌握需求分析基本方法;
2. 掌握需求分析报告书写流程。

三、实验内容与步骤

以下为以大学生党员信息管理系统为主题的需求分析报告样例。

1. 确定对系统的综合要求

1.1 功能要求

功能是刻画系统行为,特别是系统与环境关系的重要概念。功能需求定义了必须实现的软件功能,使得用户通过这些功能完成他们的任务,从而实现业务需要。

角色分析: 学生党员信息管理系统共有三个角色,即普通用户、党支部、管理员,见表 3.2。

表 3.2 系统角色类型

角 色	职责或功能
普通用户	党费管理,党员管理,信息浏览,培训指导
党支部	组织关系管理,党员管理,信息浏览,培训指导,党费管理
管理员	用户管理,权限管理,数据库管理,系统维护,系统登录

业务功能: 从业务功能出发,给出了系统的总体用例图,包括党员管理、信息浏览、培训指导、组织关系管理、党费管理等用例,如图 3.11 所示。

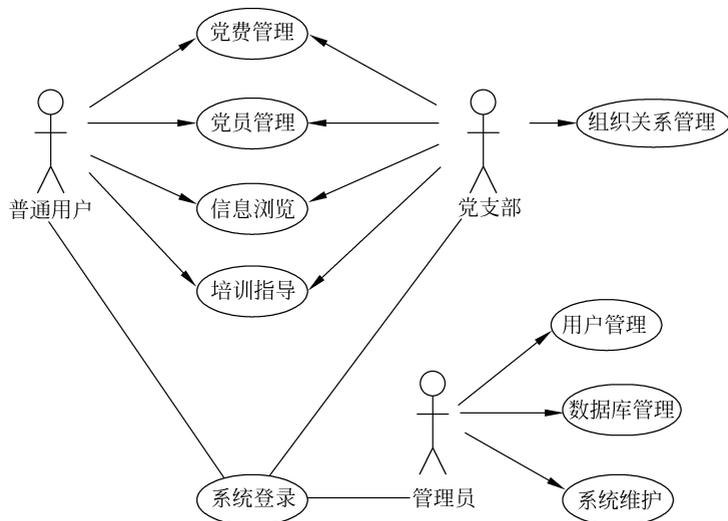


图 3.11 系统总体用例图

1) 党员管理

党员管理主要包括团员管理和发展党员管理。团员管理是指那些想要入党但没有入党的共青团团员的信息管理。发展党员管理主要包括一个申请入党人员到发展成为正式党员的整个流程,包括确定为积极分子,确定为发展对象,发展对象政治审查,确定为预备党员,预备党员转正等流程。每个不同的阶段,人员分类有所变动。

2) 信息浏览

信息浏览包括党支部发布的新闻资讯、培训会议、活动通知三方面。不同学院的信息会分配到各个学院学生登录页面上。其中,培训会议会提前通知时间地点,会议后发布内容及注意事项,或者近期需要提交的材料,还有此次会议的考勤情况公布。

3) 党费管理

党费管理用于对党员缴纳党费情况进行登记记录管理,包括党费基数设置、党费收缴登记、党费缴纳情况浏览三部分。

4) 培训指导

培训指导包括入党过程中党支部开展的培训会议,入党流程的详细资料指导。其中,党支部开展的培训会议,针对各个阶段的不同学院同学,会有详细的通知及会议的大概内容公布,具体请看信息浏览页面。此外,对于入党流程的详细资料指导,会发布各个流程的模板,将详细指导同学填写相关信息及要上交的有关材料。

5) 组织关系管理

组织关系管理主要是将支部间调转、人员调入、人员调出、党员退党人员还原到党员库。其中,支部间调转是指属于同一节点下的不同党支部或联合党支部之间的人员调转。

6) 用户管理

用户管理主要是管理进入系统的用户,包括用户的增加、修改、删除以及权限分配等。

系统管理员负责创建和维护用户组,对用户组进行权限分配,并对学校下属各个学院进行用户管理以及特殊权限管理。

系统管理员可以对用户组进行整体授权,也可以对个别用户进行特殊授权。

系统管理员能够根据用户的变化随时更换对用户的授权。

系统管理员能根据新增用户的职责或身份,将用户分配到指定的用户组,系统能自动根据用户组的权限默认给新用户授权。

系统支持对用户进行口令初始化,用户转移,删除或查询等操作。

权限范围包括范围权限、模块权限、指标权限、时间权限。

权限级别分为浏览、修改、增加、删除或无。

7) 系统维护

系统维护用于设置个人信息,设置系统公告信息,并对系统的各类参数进行设定。

8) 数据库管理

数据库管理的功能主要包括数据备份,数据库维护。数据备份主要是对数据库中的相关数据进行备份。数据库维护主要是对数据库日志等进行维护。

1.2 性能要求

(1) 易使用性:系统界面直观、简洁、美观,用户使用方便。

(2) 可靠性:系统提供的功能完善,准确反映用户需求,不存在较大的错误。

(3) 可维护性:系统管理人员能够对系统实时进行维护,包括服务器、数据库、硬件等。数据库各表格之间交叉的部分可以互相查询,但修改只能在本表格中进行。系统应具有良好的可维护性,在发生故障时,能够以最快的速度恢复运行。

(4) 可扩展性:对于管理系统的设计,应充分考虑到日后的可扩展性。系统的设计要满足未来发展所需,预留各种必要的标准接口,以便可以根据需要随时添加必要的功能模块或者更换不需要的系统模块,可以配置新的硬件设备,扩大系统整体功能。要满足系统的可扩展性,前提是系统要具有开放性,即要将系统设计为一个开放性的系统,符合一定的规范,能提供足够的手段进行功能的调整和扩展。通过软件的升级和硬件的更新,完成系统的升级和更新换代。

1.3 运行要求

服务器配置如表 3.3 所示。

表 3.3 服务器配置

系统所用数据库	SQL Server
操作系统	Windows 2003
Web 服务器	IIS7.5
数据库	MySQL Server

客户端配置如表 3.4 所示。

表 3.4 客户端配置

操作系统	Windows XP 或以上版本操作系统
浏览器	IE6.0 及其以上版本 (IE7.0、IE8.0)
分辨率	

1.4 其他要求

1) 安全性与完整性需求

安全性是指管理员能够方便地对信息进行浏览、添加、修改、删除、查询、统计等操作。制定安全策略确保系统硬件设备的安全,人员对数据操作过程中的安全、网络的安全和数据库的安全。完整性是指能够防止合法用户使用数据库时向数据库中添加不合语义的数据。通过各个表之间的联系来实现数据完整性约束。

2) 图形界面需求

系统具备友好的人机交互界面,整个系统界面风格统一,系统功能分类明确,具备界面提示功能。

3) 可扩展性和兼容性需求

该系统应尽可能满足硬件和软件平台,但同时也考虑到预留接口,在未来的功能扩展、程序的扩展和修改时,不破坏原有的架构,增强系统的适应性,让系统跨平台也能够正常使用。

4) 容错性需求

系统如存在较小错误和漏洞时,确保系统不受影响能正常运行。

2. 逐层画出系统的数据流图

0 层数据流图如图 3.12 所示。

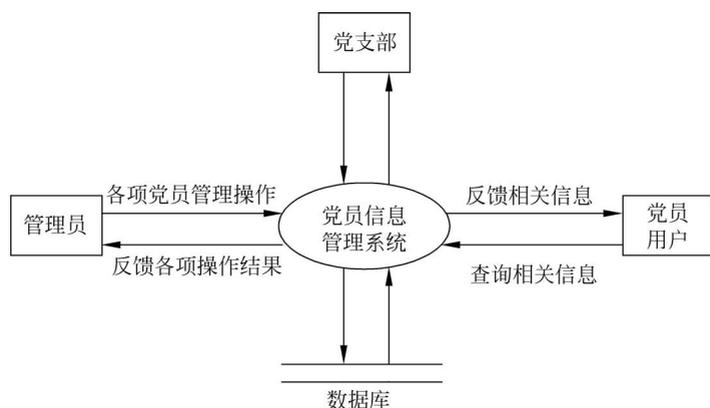


图 3.12 0 层数据流图

1层数据流图如图 3.13 所示。

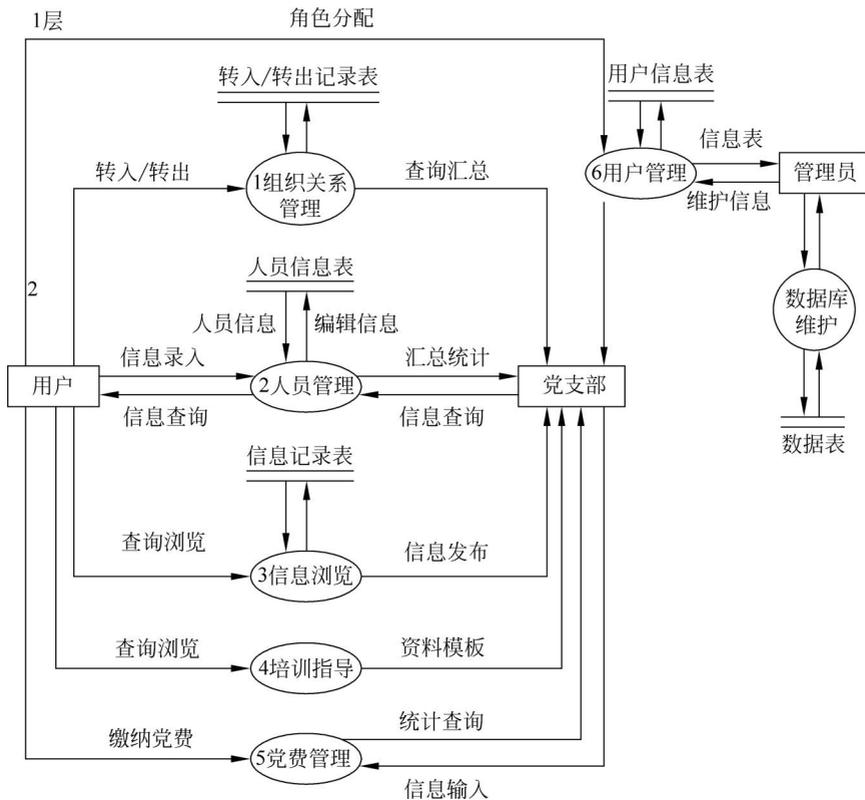


图 3.13 1层数据流图

2层数据流图如图 3.14~图 3.19 所示。

(1) 组织关系管理数据流图,见图 3.14。

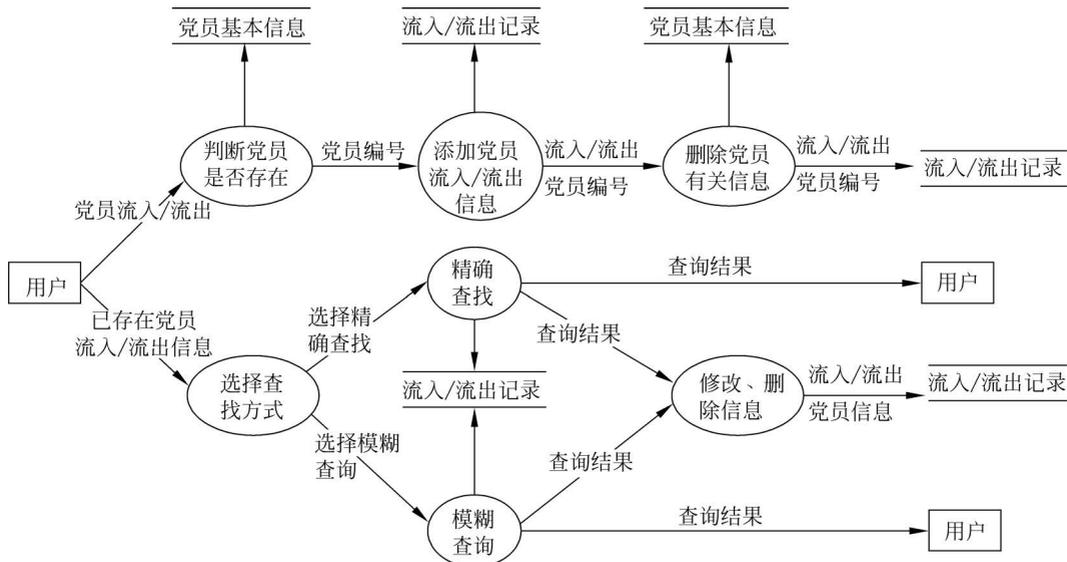


图 3.14 组织关系管理数据流图

(2) 党员管理数据流图, 见图 3.15。

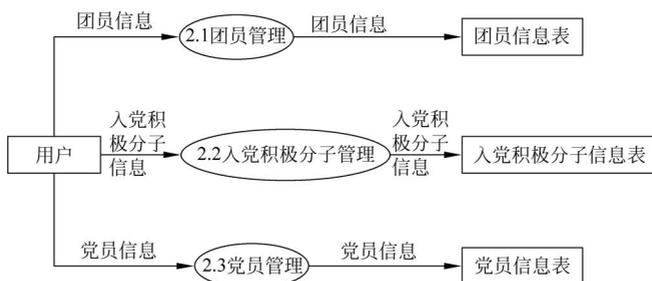


图 3.15 党员管理数据流图

(3) 信息浏览数据流图, 见图 3.16。

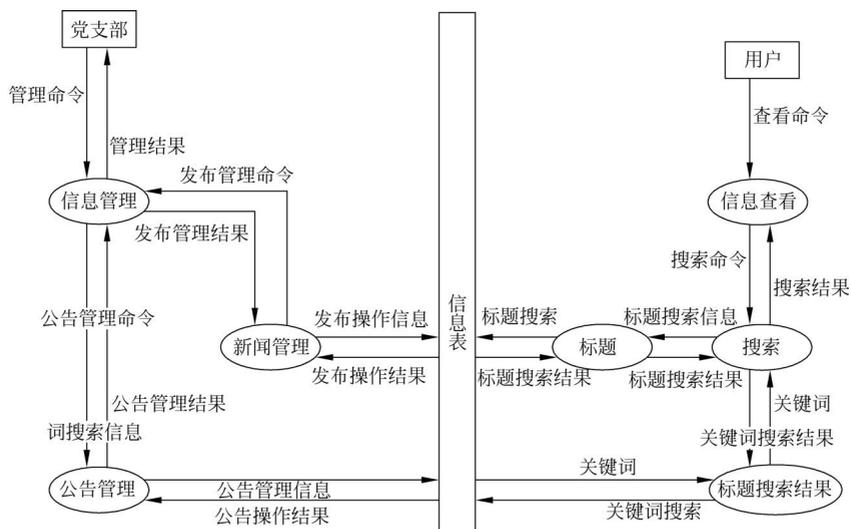


图 3.16 信息浏览数据流图

(4) 培训指导数据流图, 见图 3.17。

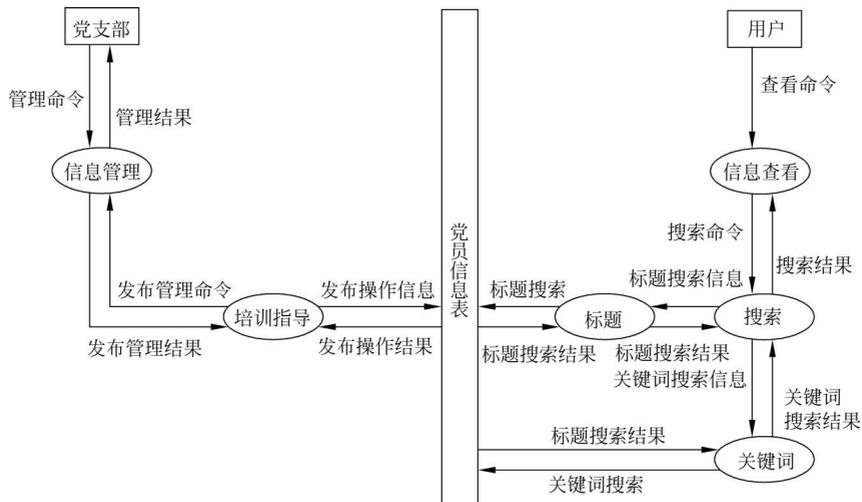


图 3.17 培训指导数据流图

(5) 党费管理数据流图, 见图 3.18。

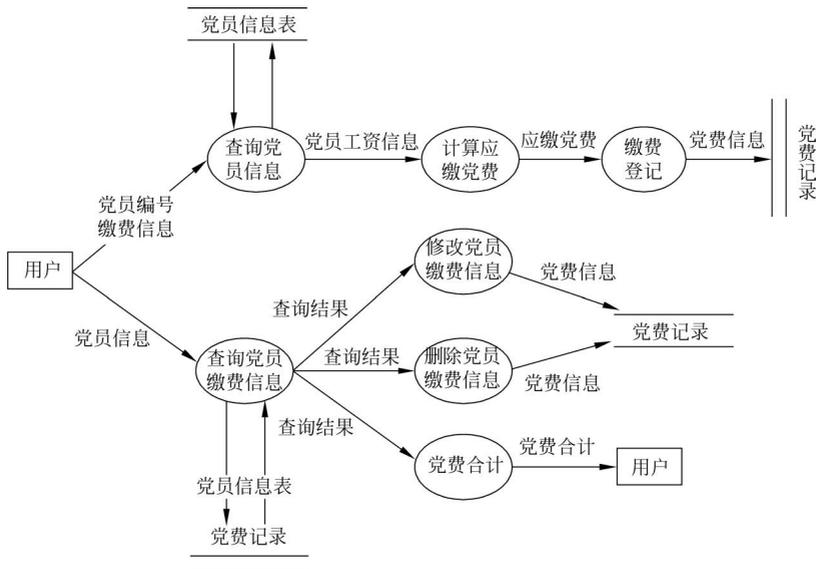


图 3.18 党费管理数据流图

(6) 用户管理数据流图, 见图 3.19。

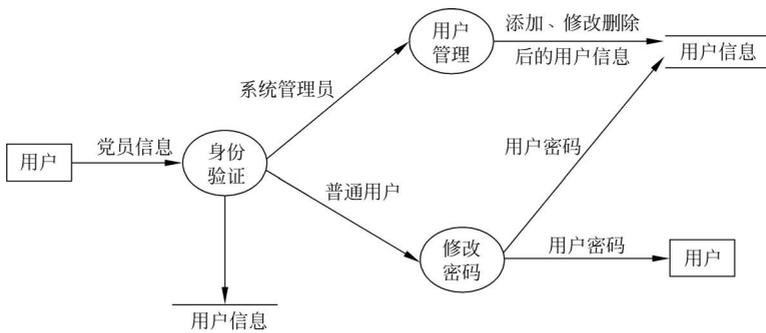


图 3.19 用户管理数据流图

3. 层数据流图

(1) 入党积极分子数据流图, 见图 3.20。

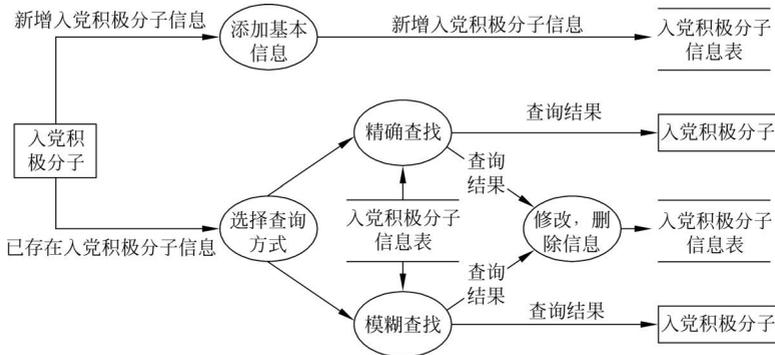


图 3.20 入党积极分子数据流图

(2) 党员管理数据流图, 见图 3.21。

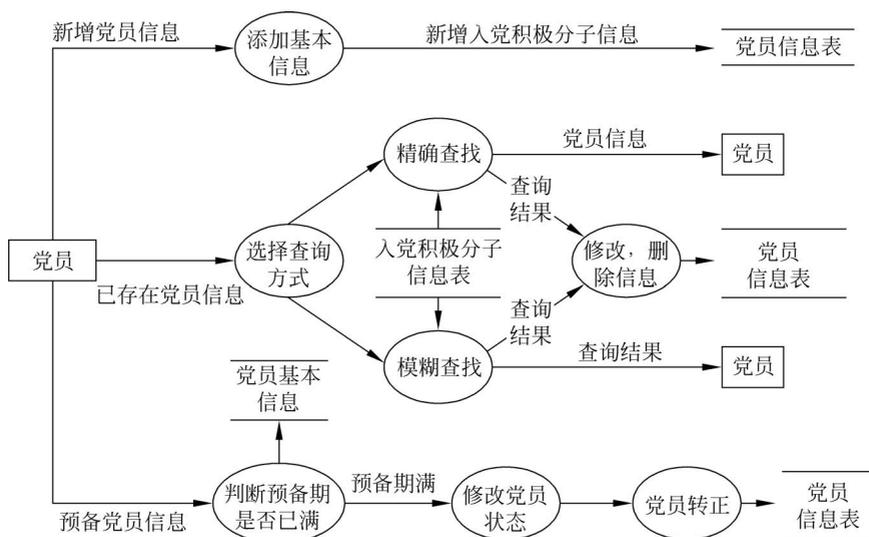


图 3.21 党员管理数据流图

4. 数据字典

4.1 数据流条目

(1) 流动党员信息表, 见表 3.5。

表 3.5 流动党员信息表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
姓名	varchar	50	姓名	否	否	是
班级	varchar	50	班级	否	否	否
性别	char	10	性别	否	否	否
民族	varchar	50	民族	否	否	否
籍贯	varchar	50	籍贯	否	否	否
出生日期	varchar	50	出生日期	否	否	否
文化程度	varchar	50	文化程度	否	否	否
正式或预备党员	varchar	50	正式或预备党员	否	否	否
入党时间	varchar	50	入党时间	否	否	否
身份证号码	varchar	100	身份证号码	否	否	否
家庭住址	varchar	120	家庭住址	否	否	否
原党支部联系人	varchar	50	原党支部联系人	否	否	否
原党支部联系人电话	varchar	50	原党支部联系人电话	否	否	否
流入地(单位)党支部名称	varchar	50	流入地(单位)党支部名称	否	否	否
流入时间	varchar	50	流入时间	否	否	否
流入党支部联系人	varchar	50	流入党支部联系人	否	否	否
流入党支部联系人电话	varchar	50	流入党支部联系人电话	否	否	否

(2) 党员关系转入转出信息表,见表 3.6。

表 3.6 党员关系转入转出信息表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
姓名	varchar	50	姓名	否	否	是
转出单位	varchar	100	转出单位	否	否	否
转入单位	varchar	100	转入单位	否	否	否
转出日期	varchar	50	转出日期	否	否	否
党员原所在基层党委通信地址	varchar	150	党员原所在基层党委通信地址	否	否	否
党员原所在基层党委联系电话	varchar	50	党员原所在基层党委联系电话	否	否	否
党员原所在基层党委邮编	varchar	50	党员原所在基层党委邮编	否	否	否
转入单位经办人	varchar	50	转入单位经办人	否	否	否
转入单位经办人联系电话	varchar	100	转入单位经办人联系电话	否	否	否
接收时间	varchar	50	接收时间	否	否	否

4.2 数据项条目

党员学号

数据项名称: 党员学号

数据项别名: party member_no

说明: 学生党员的唯一标识

类型: 字符串

长度: 10

取值范围及含义: 1~2 位为校区号; 3~4 位为学院编号; 5~8 位为专业编号; 9~10 位为人员编号

4.3 数据文件条目

(1) 入党积极分子信息表,见表 3.7。

表 3.7 入党积极分子信息表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
学号	char	10	学号	否	否	是
姓名	varchar	50	姓名	否	否	否
班级	varchar	50	班级	否	否	否
性别	char	10	性别	否	否	否
民族	varchar	50	民族	否	否	否
籍贯	varchar	50	籍贯	否	否	否
出生日期	varchar	50	出生日期	否	否	否
职务	varchar	50	职务	否	否	否
申请时间	varchar	50	申请时间	否	否	否
联系人员	varchar	50	联系人员	否	否	否
培养教育	varchar	50	培养教育	否	否	否

续表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
培养人	varchar	50	培养人	否	否	否
考察意见	varchar	50	考察意见	否	否	否
培训时间	varchar	50	培训时间	否	否	否
积极分子时间	varchar	50	积极分子时间	否	否	否
党课成绩	varchar	50	党课成绩	否	否	否
电话	varchar	50	电话	否	否	否
地址	varchar	120	地址	否	否	否
Email	varchar	50	Email	否	否	否
备注	varchar	250	备注	否	否	否

(2) 预备党员信息表,见表 3.8。

表 3.8 预备党员信息表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
学号	char	10	学号	否	否	是
姓名	varchar	50	姓名	否	否	否
班级	varchar	50	班级	否	否	否
性别	char	10	性别	否	否	否
民族	varchar	50	民族	否	否	否
籍贯	varchar	50	籍贯	否	否	否
出生日期	varchar	50	出生日期	否	否	否
职务	varchar	50	职务	否	否	否
申请时间	varchar	50	申请时间	否	否	否
培养教育	varchar	50	培养教育	否	否	否
考察意见	varchar	50	考察意见	否	否	否
党费收缴	varchar	50	党费收缴	否	否	否
培训时间	varchar	50	培训时间	否	否	否
积极分子时间	varchar	50	积极分子时间	否	否	否
预备党员时间	varchar	50	预备党员时间	否	否	否
党课成绩	varchar	50	党课成绩	否	否	否
电话	varchar	50	电话	否	否	否
地址	varchar	120	地址	否	否	否
Email	varchar	50	Email	否	否	否
备注	varchar	250	备注	否	否	否

(3) 正式党员信息表,见表 3.9。

表 3.9 正式党员信息表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
学号	char	10	学号	否	否	是

续表

字段名称	数据类型	字段属性				
		字段大小	描述	是否为空	是否外键	是否主键
姓名	varchar	50	姓名	否	否	否
班级	varchar	50	班级	否	否	否
性别	char	10	性别	否	否	否
民族	varchar	50	民族	否	否	否
籍贯	varchar	50	籍贯	否	否	否
出生日期	varchar	50	出生日期	否	否	否
职务	varchar	50	职务	否	否	否
申请时间	varchar	50	申请时间	否	否	否
培训时间	varchar	50	培训时间	否	否	否
积极分子时间	varchar	50	积极分子时间	否	否	否
预备党员时间	varchar	50	预备党员时间	否	否	否
转正时间	varchar	50	转正时间	否	否	否
党课成绩	varchar	50	党课成绩	否	否	否
电话	varchar	50	电话	否	否	否
地址	varchar	120	地址	否	否	否
Email	varchar	50	Email	否	否	否
备注	varchar	250	备注	否	否	否

4.4 数据加工条目

审核表

数据加工名称：审核预备党员是否期满

加工编号：2.3

说明：根据预备党员申请转正的时间和正式成为预备党员的时间相减后审核是否满一年

输入数据流：申请转正的时间和正式成为预备党员的时间

输出数据流：预备党员是否期满,能否转正

加工逻辑：

预备党员提交转正申请

在预备党员信息表中找出该预备党员正式成为预备党员的时间

If(申请转正时间-正式成为预备党员的时间>1)

{输出预备期满,同意转正}

else{输出预备期未滿一年}

四、实验思考和总结

请大家根据上述实验步骤,独立完成下列题目,并完成实验报告。校园网系统通常被设计为满足学校或大学校园网络需求的综合性解决方案。在第2章可行性分析的基础上,请按照样例需求分析报告完成该系统需求分析报告并进行总结,分析内容主要包括数据流图、数据字典等内容。

