

软件需求分析

需求分析是软件开发过程中的一个非常重要的阶段,是整个软件开发的重要依据和基础,关系软件系统开发的成败,是决定软件产品质量的关键。该阶段的主要任务是明确软件系统必须“做什么”,即明确系统的功能、性能、可靠性等方面的具体要求。本章主要介绍需求分析的概念、原则、任务、步骤、方法,需求分析文档等内容。■



教学目标

- 理解软件需求分析的概念和特点、目的和原则。
- 熟悉软件系统需求分析的具体任务及步骤。
- 掌握需求分析描述工具并编写软件需求文档。

■教学课件

第 3 章 软件
需求分析



3.1 软件需求分析概述

【案例 3-1】 全球第一台带有图形用户界面(GUI)和鼠标的个人电脑 Apple Lisa(以乔布斯女儿的名字“利萨”命名)在 1983 年上市时的售价为 9995 美元(约相当于今天的两万多美元),远远超过市场用户的承受能力,而且运行速度缓慢,性价比太低。所以,其销量远低于苹果公司的预期,企业用户更愿意选择价格更低的 IBM 电脑。很明显该产品在做需求的时候,忽略了非常重要的价格因素,导致产品研发失败。

软件需求分析阶段,主要是搞清软件应用用户的实际具体需求,包括功能需求、性能需求、数据需求、安全及可靠性要求、运行环境和将来可能的业务变化及拓展要求等,并建立系统的逻辑模型,写出“软件需求规格说明(SRS)”等文档。

■知识拓展

软件需求的重
要性



3.1.1 软件需求分析的概念及特点

1. 需求的概念

IEEE 的软件工作标准术语表(1990)将需求定义为如下。■

(1) 用户解决问题或达到目标所需的条件或能力。

(2) 系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或能力。

(3) 一种反映上面(1)或(2)所描述的条件或能力的文档说明。

通俗地讲,“需求”就是用户的需要,包括用户要解决的问题、达到的目标以及实现这些目标所需要的条件,是一个程序或系统开发工作的说明,表现形式一般为文档。

软件需求分为3个层次:业务需求、用户需求、功能需求,还包括某些类别的非功能需求。

(1) 业务需求:反映组织机构或客户对系统和产品高层次的目标要求,它们在项目视图与范围文档中予以说明。

(2) 用户需求:是从用户角度描述系统所完成的任务或者是用户期望有的产品属性。用户需求文档用于描述用户使用软件产品要完成的任务。

(3) 功能需求:主要描述系统所提供的功能或服务,即定义系统的主要功能、系统的输入输出信息、系统的约束等。

(4) 非功能需求:作为功能需求的补充,主要描述那些与系统的具体功能无关,但与系统的总体特性相关的特征,如安全性、可靠性、响应时间、可移植性、可重用性等。

2. 软件需求分析

软件需求分析(Requirement Analysis)也称为需求分析、软件需求、系统需求分析或需求分析工程等。通常是软件需求分析师经过深入细致的调研和分析,准确理解用户需求和项目的功能、性能、可靠性等具体要求,将用户非形式化的需求表述转化为完整的具体需求定义,再将需求定义写成需求规约的过程。

需求分析的任务就是借助于当前系统的逻辑模型导出目标系统的逻辑模型,解决目标系统“做什么?”的问题。在需求分析阶段最后,需要提交较为详细的数据流图、数据字典和需求分析报告。

3. 软件需求分析的特点

需求分析的特点及难点,主要体现在以下5个方面。

(1) 指标确定难。主要原因包括:一是应用领域的复杂性及业务变化难以具体确定;二是用户需求的因素多且很难准确描述,如软件的功能、性能、可靠性和接口及运行环境等。

(2) 需求动态性。软件的用户需求和实际业务与数据更新发展变化,需求不成熟、不稳定和不规范。

(3) 交流共识难。需求分析涉及的人、事、物及相关因素多,与用户、业务专家、需求分析人员和项目管理员等进行交流时,不同的背景知识、角色和角度等,交流形成共识较难。

(4) 完备一致难。不同人员对系统的要求及认识不尽相同,对问题的表述方式方法有差异,各方面的需求描述容易产生难以消除的不一致或矛盾。

(5) 深入完善难。需求理解针对客户环境和业务流程的改变、市场趋势的变化等,也

会随着分析、设计和实现而不断深入完善,可能在最后重新修订软件需求。分析人员应认识到需求变化的必然性,并采取措施减少需求变更对软件的影响。



注意: 为了克服需求分析的困难,应围绕需求分析的方法、计算机辅助开发工具及形式化需求分析等方面展开研究,同时,需要运用丰富的实际经验。

3.1.2 软件需求分析的目的和原则

1. 软件需求分析的目的及重点

软件需求分析的主要目的是获取用户及项目的具体需求,通过对实际需求的获取、分析、文档化和验证等需求分析过程,为软件的进一步设计和实现提供依据。

(1) 需求划分。将软件功能、性能、可靠性等相关需求进行分类、逐一细化。

(2) 面向用户及项目获取分析需求。软件研发其他阶段都是面向技术的,只有需求分析阶段是面向用户的,深入调研获取并分析软件的功能、性能、可靠性等,也可从系统和用户需求中推导出软件的具体需求,并检查需求定义准确性以及是否存在二义性。

(3) 检查和解决不同需求之间存在的矛盾或不一致问题,尽量达到均衡和优化。

(4) 确定软件的边界及范围,以及软件与环境的相互作用方式等。例如,业务实际应用及运行边界、范围和具体环境,以及网络运行及存储环境和数据库应用等。

(5) 对需求文档化并进行最后验证与确认。

需求分析的重点: 通过分析业务流程和数据流程等技术手段,与客户共同确定业务模型、功能模型、性能模型、接口模型等主要图表及要素。



注意: 对必要的需求变更应当经过认真申报、评审、跟踪和比较分析后才能实施。

2. 软件需求分析的原则

为了促进软件研发工作的规范化、科学化,一些专家和专业人士提出了许多软件需求分析方法,如结构化方法、原型化法、面向对象方法等。在实际需求分析工作中,各种需求分析方法都有其独特的思路和表示方法,基本都适用以下的需求分析基本原则。

(1) 功能分解,逐层细化。将需求问题按具体功能、性能等方面逐步分解细化,建立问题层次结构并逐一分析。

(2) 表达理解问题的数据域和功能域。数据域包括数据流、数据内容和数据结构,功能域则反映这3方面的控制信息。

(3) 建立模型。模型包括各种图表是对研究对象特征的一种重要表达形式,通过逻辑视图可以帮助分析人员更好地理解软件的信息、功能和行为。



讨论思考

(1) 什么是需求分析? 需求分析主要确定新系统的哪些工作?

(2) 需求分析的特点、目的和重点是什么?

- (3) 需求分析的原则有哪些?
- (4) 针对教务管理系统等,分析系统中存在的问题。

3.2 软件需求分析的任务及过程

需求分析的任务是明确目标系统的确切要求,建立需求文档并进行最后的验证与确认。需求分析是发现、逐步求精、建立模型、规格说明和复审的过程。

3.2.1 软件需求分析的任务

软件需求分析的基本任务是通过软件开发人员与用户的交流和讨论,准确地分析理解原系统,定义新系统的基本功能、性能、开发时间、投资情况人员安排等,并获得需求规格说明书。需求分析的具体任务包括如下几个方面。

1. 确定目标系统的具体要求

在可行性研究的基础上,双方通过交流,确定对问题的综合需求。这些需求包括功能需求、性能需求、环境需求和用户界面需求;另外还有系统的可靠性、安全性、可移植性和可维护性等方面的需求。双方在讨论这些需求的内容时一般通过双方交流、调查研究来获取,并达到共同的理解。

(1) 确定功能需求。确定系统功能需求,画出功能结构图,完成新系统的功能点列表,即功能模型。有时将性能模型、界面模型和接口模型的内容都合并其中,功能模型可用 Use Case 矩阵/图表示。

【案例 3-2】 某高校毕业设计管理系统的功能列表(功能模型),如表 3-1 所示。

表 3-1 某高校毕业设计管理系统的功能

功能类别/标识符	使用角色	目标描述
选题管理(CTM)	教师/学生/教务人员	完成教师立题、学生选题的双向选择过程。最终达到每人一题
过程管理(PPM)	教师/学生/教务人员	完成教师与学生交流、中期检查、教师与学生互评过程
答辩管理(DM)	教师	完成答辩准备工作,提交答辩结果
后期处理(BPM)	教务人员	完成收集、上报材料,统计成绩,评优过程
登录管理(LM)	教师/学生/教务人员	提供用户登录验证及用户权限查询的功能
系统维护(SM)	管理员	系统维护包括身份管理、流程管理和数据维护 3 个子功能块

(2) 获取性能需求。性能需求是为保证软件功能的实现和正确运行,对软件所规定

的效率、可靠性、安全性等,包括软件的效率(运行处理速度)、可靠性、安全性、适用性、可移植性、可维护性和可扩充性等方面的需求,还应考虑业务发展的扩展及更新维护等。

【案例 3-3】 某考务系统的部分性能点列表(性能模型),如表 3-2 所示。

表 3-2 某考务系统的性能点列表

编号	性能名称	使用部门	使用角色	性能描述	输入	系统响应	输出
1	网上报名响应时间	考生部门	考生	300 并发用户情况下报名时间<1.5s	考生身份证、姓名、考试科目等信息	检查报名信息,如果正确存入数据库	显示报名信息
2	成绩查询响应时间	考生部门	考生	300 并发用户情况下查询时间<1.5s	考生身份证、考号、姓名	按照输入的组 合条件,进行 查询	显示考生身份 证、姓名、科 目、成绩
3	成绩统计响应时间	考务部门	管理员	后台统计时间<2s	科目/学校	按照输入条 件,统计考试 各分数段情况	显示各分数段 人数及百分比

(3) 明确处理关系,列出接口列表。应用软件可能还与机构内部的其他应用软件集成,所以,需要明确与外部应用软件数据交换的内容、格式与接口,以实现数据及功能的有机结合。

【案例 3-4】 某系统与财务收费系统的部分接口列表,如表 3-3 所示。

表 3-3 目标系统的接口列表(接口模型)

编号	接口名称	接口规范	接口标准	入口参数	出口参数	传输速率
1	与财务系统的接口	财务系统规定的接口规范	记账凭证与分录的具体格式	① 凭证记录参数:凭证编号、日期、单据张数、借方合计、贷方合计。 ② 分录记录参数:凭证编号、日期、借方、贷方、数量、单价、摘要	① 凭证记录格式:编号、状态、会计期间、凭证字号、日期、单据张数、审核、过账、制单、过账状态、借方合计、贷方合计。 ② 分录记录格式:分录编号、凭证编号、摘要、科目代码、结算号、结算日期、结算方式、借方、贷方、数量、单价	一张凭证一次处理传送

(4) 确定系统的运行环境及界面。环境需求包括:软件运行时所需要的硬件的机型、外设;软件的操作系统、开发与维护工具和数据库管理系统等要求。运行环境包括服务器及核心计算机与网络资源(系统软件、硬件和初始化数据)的配置计划、采购计划、安

装调试进度、人员培训计划等内容。界面需求包括界面的风格、用户与软件的交互方式、数据的输入输出格式等。

2. 建立目标系统的逻辑模型

软件系统的逻辑模型分为数据模型、功能模型和行为模型。可用层次的方式对逻辑模型进行细化,并采用相应的图形以及数据字典进行描述。其中数据模型采用 E-R 图描述;功能模型常用数据流图来描述;行为模型常用状态转换图来描绘系统的各种行为模式(状态)和不同状态间的转换;数据字典用来描述软件使用或产生的所有数据对象。具体描述工具及用法将在 3.5 节进行介绍。

3. 编写需求文档,验证确认需求

(1) 编写“需求规格说明书(Software Requirement Specification, SRS)”。软件需求规格说明书描述系统的数据、功能、行为、性能需求、设计约束、验收标准,以及其他与系统需求相关的信息。

(2) 编写初步用户使用手册。使用手册反映系统的功能界面和用户使用的具体要求,用户手册能强制分析人员从用户使用的观点考虑软件。

(3) 编写确认测试计划,作为今后确认和验收的依据。

(4) 完善开发计划。在需求分析阶段对开发的系统有了更进一步的了解,因此对原计划要进行适当修正并加以完善。

对各项具体需求在逐一验证后由用户确认签字。需求分析文档的具体编写步骤和格式等将在 3.6 节介绍。



注意: 上述任务要具体分析,灵活运用。如果需求分析之后,对将要实现的新系统仍然感到不够明确时,不应签字确认,还需进行进一步深入分析。

3.2.2 软件需求分析的过程

软件需求分析的过程也称为需求开发,可分为需求获取、需求描述、需求有效性验证及编写需求文档等步骤,如图 3-1 所示。

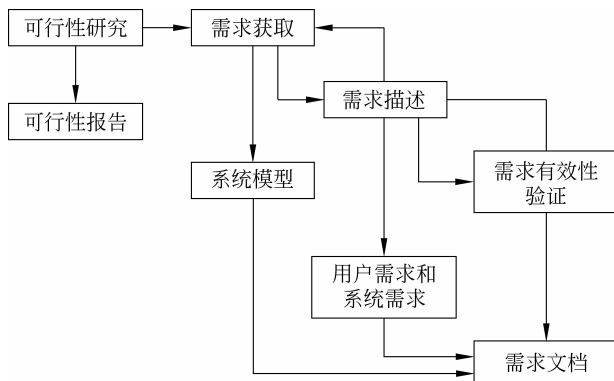


图 3-1 软件需求分析的过程

1. 需求获取

需求获取(Requirements Elicitation)也称为**需求调查**,是需求开发的核心,其活动包括收集、发现、提炼和定义需求。**获取的目的**是为了发现业务需求、用户需求、功能需求和非功能性需求。需求获取是软件开发过程中最关键、最易出错且需要密集沟通的阶段。在具体**获取需求过程中采用的技术**主要包括如下。

(1) 调研访谈。调研和访谈是最常见的方式。在访谈时应提前做好问题或某种假设模型,以使用户能从中找到思考问题的出发点。

(2) 实地观察。有些需求用户很难描述清楚,需要分析师到现场去体验和观察用户的实际工作流程,了解用户如何利用软件和其他人协作完成某项任务。

(3) 场景描述。需求分析师为每个用户任务设计一个场景,以提问的方式提取需求。场景通常以用例图来表示。

(4) 构造原型。在需求分析过程中,开发人员根据对软件的理解,先建立一个系统原型,然后让用户对其评估,并提出修改意见,最终实现全面、准确地确定系统的具体需求。

(5) 问卷调查。获得需求的其他常见方法是向客户机构的相关人员发放调查表,一般经仔细考虑的书面回答可能比面谈对问题的口头回答更清晰准确,问卷调查的关键是问题设计。

2. 需求描述

从系统角度理解软件,确定对所开发软件的综合要求,并提出需求的实现条件及应该达到的标准。其需求具体包括功能需求(做什么)、性能需求(要达到什么指标)、可靠性需求(不发生故障的概率)、环境需求(如机型、操作系统等)、安全保密需求、用户界面需求、资源使用需求(网络传输带宽、软件运行所需的内存、CPU等),以及软件成本消耗与开发进度需求,预测以后系统可能达到的目标等。同时,以各种图表和方法进行详细描述。

3. 编写需求文档

在上述需求分析的基础上,建立软件的需求说明文档;编写初步用户使用手册;编写确认测试计划;修改完善项目开发计划。

4. 需求验证

需求验证也称为需求检验或评审,是对初步确定的需求进行验证或评审,明确正确且可行的具体需求。需求验证从软件的一致性、完整性、现实性和有效性等方面进行审核和验证,以确定正确和可行的需求,排除不可行的。验证方法包括自查法、用户审查法、专家审查法和原型法等。

知识拓展
需求分析的
过程



讨论思考

- (1) 需求分析的具体任务有哪些?
- (2) 需求分析的常规步骤是什么?

3.3 软件需求分析方法

开发不同的软件项目,所采用的需求分析方法各具特点、各有不同,目前常用的分析方法有结构化分析法和面向对象的分析方法。在需求分析过程中不存在哪一种方法最好,而应当通过组合利用不同的方法实现最好的软件需求模型。

3.3.1 软件需求分析方法类型

目前,软件需求的分析与设计方法较多,目前常用的需求分析方法分为4种: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。将一个系统看作由多个功能模块组成的集合,各功能模块又可分解为若干子功能和接口,子功能还可继续分解。即:

功能分解=功能+子(模块)功能+功能接口

(2) 结构化分析方法。结构化分析方法是一种从业务应用问题空间到某种处理表示的映射方法,是结构化方法中重要且被普遍接受的表示方法,由DFD和数据词典构成并表示。此分析法又称为数据流法。其基本策略是跟踪数据流,即研究问题域中数据的流动方式及在各个环节上所进行的处理,从而发现数据流和加工(Bubble)。

结构化分析=数据流+数据处理(加工)+数据存储+端点+处理说明+数据字典

(3) 信息建模方法。主要是从数据角度对现实世界建立模型。大型软件较复杂,很难直接对其进行分析与设计,常借助模型。模型是开发中常用的工具,系统包括数据处理、事务管理和决策支持。实质上,也可看成由一系列有序模型构成,其有序模型通常为功能模型、信息模型、数据模型、控制模型和决策模型,有序是指这些模型是分别在系统的不同开发阶段及开发层次上建立的。建立系统常用的基本工具是E-R图,经过改进后称为信息建模方法,后来又发展为语义数据建模方法,并引入了许多面向对象的特点。

信息建模=实体(对象)+属性+关系+父类型/子类型+关联对象

此方法的核心概念是实体和关系,基本工具是E-R图,其基本要素是实体、属性和联系。方法的基本策略是从现实中找出实体,再用属性描述。

(4) 面向对象的分析方法。面向对象的分析方法(OOA)的关键是识别问题域内的对象,分析其之间的关系,并建立3类模型:对象模型、动态模型和功能模型。OOA还可表示为:

面向对象=对象/类+结构与连接+继承+封装+消息通信

OOA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(属性)和行为特征(服务)。面向对象的分析与设计方法将在第5章具体介绍。

3.3.2 软件需求分析方法对比

面向功能分析、面向对象分析、面向数据分析都是面向流程分析的,功能、对象和数据都是在流程中产生且为流程服务的,各适用不同的新系统。

最早的面向功能分析将软件需求看作倒置的“功能树”，树根是总功能，每个节点都是一个具体功能，以“自顶向下、逐步求精”的思想，适用于结构化软件工程。面向数据分析是指面向元数据和中间数据的分析，只有将这两类数据及其关系分析透彻才可达到目的。

3 种需求分析方法优缺点和适用范围对比，如表 3-4 所示。

表 3-4 3 种需求分析方法对比

分析方法名称	目的	优缺点	适用范围
面向功能分析	获取功能模型	简单明了	系统软件和应用软件
面向对象分析	获取对象模型	复杂抽象	系统软件和应用软件
面向数据分析	获取数据模型	抓住本质	关系数据库信息系统

3.3.3 软件需求分析技巧

在实际应用中，进行需求分析需要一些技巧，主要包括如下。

- (1) 需求分析是分析师与用户双方进行配合的项目，需要密切交流合作。
- (2) 在微观上/宏观上都应以流程为主。
- (3) 注重事实，坚持客观调研及主见，不应偏听偏信。
- (4) 构建需求金字塔。决策层提出宏观上的统计、查询、决策需求，管理层提出业务管理和作业控制需求，操作层提出录入、修改、提交、处理、打印、界面、传输、通信、时间与速度等方面的操作需求。
- (5) 注重主动征求各层的意见和建议，一般需求分析过程需要集中汇报征求意见 2 或 3 次。

 讨论思考

- (1) 从系统的角度出发可以将需求分析方法分为哪几种？
- (2) 需求分析的技巧有哪些？

3.4 结构化分析方法

结构化开发方法(Structured Developing Method)是软件开发方法中最成熟、应用最广泛的方法，主要特点是快速、自然和便捷。结构化开发方法由结构化分析方法(SA 法)、结构化设计方法(SD 法)及结构化程序设计方法(SP 法)构成。

3.4.1 结构化分析方法的基本思想

结构化分析(Structured Analysis,SA)方法是面向数据流的需求分析方法，是 20 世纪 70 年代末由 Yourdon、Constaintine 及 DeMarco 等人提出并发展的。SA 方法根据软件内部的数据传递、变换关系，自顶向下、逐层分解，绘制出满足功能要求的模型。

结构化方法总的指导思想是“自顶向下、逐步求精”(如图 3-2 所示)，其基本原则是抽象与分解。

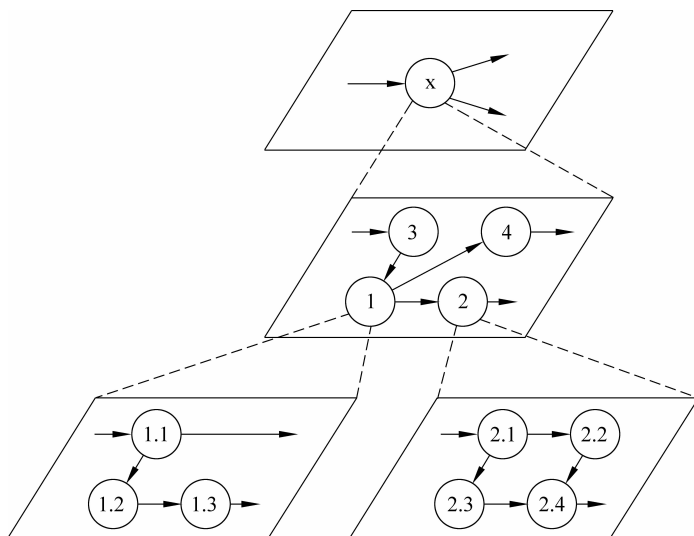


图 3-2 自顶向下逐层分解

(1) 抽象。分析问题时先考虑问题本质的属性,暂把细节略去,以后再逐层添加细节,直至涉及最详细的内容。

(2) 分解。对复杂的问题,常分解为几个相对易于解决的小问题,然后再分别解决。分解的方法可分层进行,原理是忽略细节,先考虑问题最本质的方面,形成问题的高层概念;然后再逐层添加细节。

3.4.2 结构化分析的步骤

1. 结构化分析的具体步骤

对软件进行结构化分析的具体步骤如下。

(1) 构建原系统物理模型。对原系统进行详细调研并收集资料,通过认识原系统的工作过程,将看到、听到和收集到的实际情况用图表或文字进行描述。用模型表示对原系统的理解,如业务流程图等。

(2) 抽象出原系统的逻辑模型。逻辑模型反映了原系统“做什么”的功能,需要去除物理模型中非本质的物理因素等,抽象并提取其本质的因素。

(3) 建立新系统逻辑模型。在原系统的逻辑模型基础上,可将新系统与原系统逻辑进行比较分析,查看决定变化的范围,找出要改变的部分,并抽象为一个子模块,以确定子模块的外部环境及输入输出。

(4) 进一步补充和优化。新系统的逻辑模型只是一个主体,为了完整地进行描述,还应进行补充。补充的内容包括:应用环境及与外界环境的相互联系,说明目标系统的人机界面,尚未详细考虑的环节,如出错处理、输入输出格式、存储容量和响应时间等性能要求与限制。

结构化分析导出的分析模型包括数据模型、功能模型和行为模型,这些模型以“数据字典”为核心,数据字典描述了软件使用的所有数据对象,如图 3-3 所示。其中,实体-联系图(E-R 图)作为数据建模的基础,描述数据对象实体及其关系;数据流图(DFD)作为

功能建模的基础,描述数据怎样转换以及转换功能;状态转换图(State Transition Diagram,STD)作为行为建模的基础,表示系统的各种行为状态以及状态间的转换方式。

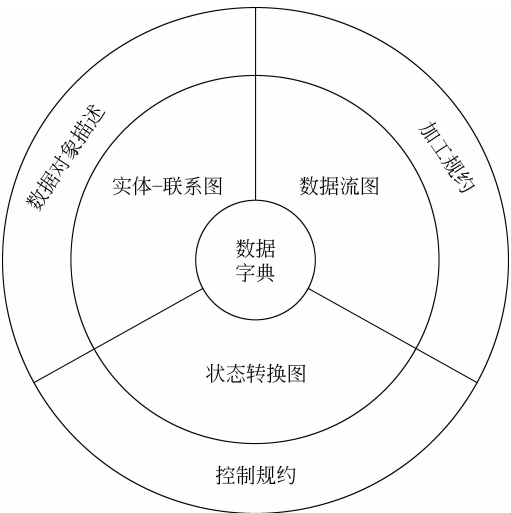


图 3-3 结构化分析模型

2. 结构化分析的流程

结构化方法发展历史悠久且应用广泛,主要优点为:简单、实用、成熟;适合于瀑布模型,易为开发者掌握;成功率较高,据对美国 1000 家公司统计,该方法的成功率高达 90.2%,仅次于面向对象的方法;该方法特别适合于数据处理领域中的应用,对其他领域的应用也基本适用。其缺点是:不太适应规模大的复杂项目,难以解决软件重用问题,较难适应需求变化,难以彻底解决维护问题。整个结构化开发方法流程如图 3-4 所示。

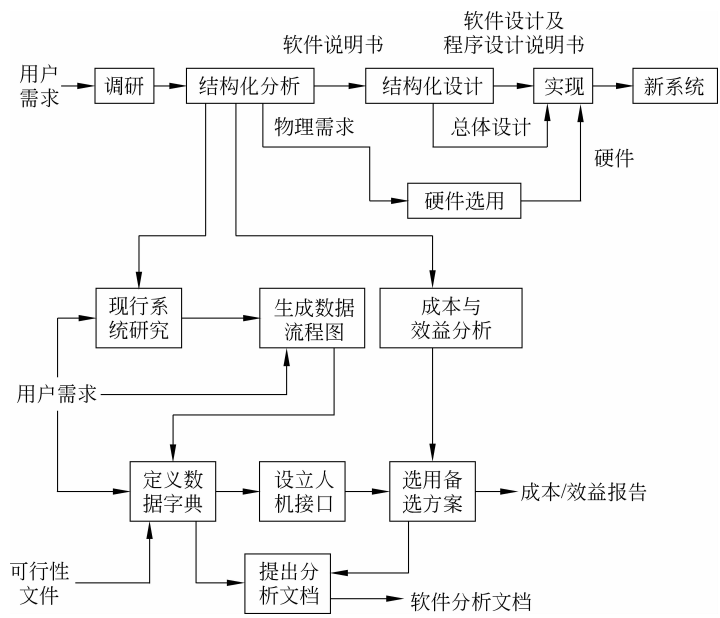


图 3-4 结构化开发方法流程

3.5 结构化分析建模工具

需求描述工具的选择通常与具体需求分析方法和阶段有关,面向过程和面向数据的分析方法,常用的描述工具包括实体-联系图、数据流图、状态转换图和数据字典。面向对象的分析方法主要采用 UML 语言和用例图、类图、活动图、时序图等,具体将在第 5 章介绍。

3.5.1 实体-联系图

实体-联系图(Entity Relationship Diagram, E-R 图)是目前最常用的数据建模方法。主要用于在需求分析阶段清晰地表达目标系统中数据之间的联系及其组织方式,建立系统的实体-联系模型(E-R 模型)。实体-联系模型可以在软件实现时转换成各种不同数据库管理系统所支持的数据物理模型。实体-联系模型由实体、联系和属性 3 个基本成分组成。

(1) 实体:指客观世界存在的,且可以相互区分的事物。实体可以是人,也可以是物,还可以是抽象概念,如学生、课程、产品都是实体。

(2) 属性:也称为性质,指实体某一方面的特征。一个实体通常由多个属性值组成,如学生实体具有学号、姓名、出生年月、入学日期、专业等属性。

(3) 联系:指实体之间的相互联系。实体之间的联系可主要划分为 3 类。

一对一(1:1),例如,一个学院有一名院长。

一对多(1:n),例如,每个出版社出版多本书,但是每本书只能出自一个出版社。

多对多($m:n$),例如,学生与课程之间的联系是多对多的关系,一个学生可以学多门课程,每门课程有多个学生。

E-R 图由矩形框、菱形框、椭圆或圆角矩形及连线组成。其中,矩形框表示实体,菱形框表示关系,椭圆或圆角矩形表示实体(或关系)的属性。

【案例 3-5】 成绩管理系统的 E-R 图如图 3-5 所示。

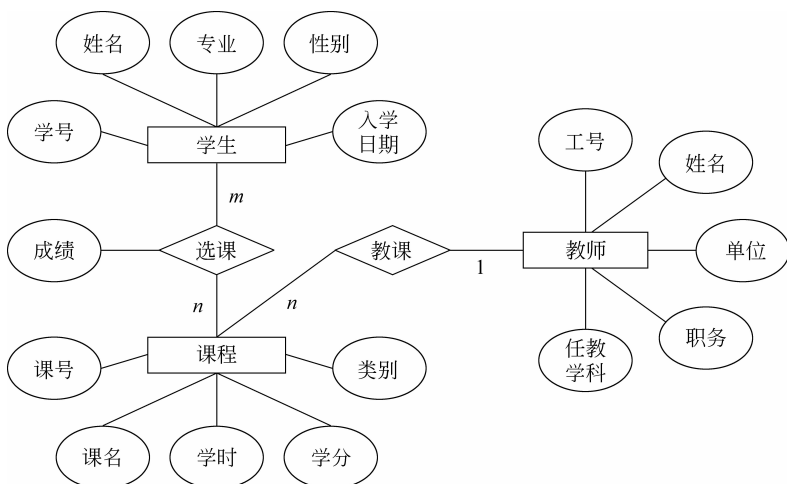


图 3-5 成绩管理系统的 E-R 图

3.5.2 数据流图

1. 数据流图的概念和作用

数据流图(Data Flow Diagram,DFD)是一种图形化的系统模型,在一张图中将新软件系统建模为输入、处理、输出和数据存储。运用图形方式描述系统内部的数据流程,形象、准确地表达了系统的各处理环节以及各环节之间的数据联系,是结构化系统分析方法的主要表达工具。DFD 运用加工、文件、数据流线等图形来反映系统的逻辑功能及其内部的数据联系。它也是一种功能处理模型,且简单易理解、应用广泛。

DFD 是结构化分析的最基本工具。由一系列表示系统中元素的图形符号组成,这些符号表达了系统中各元素之间的数据的具体流动和处理过程,如输入和输出、人工处理、数据处理、数据库、文件和设备等。

知识拓展

TFD 与 DFD
的区别与联系



2. 基本图形符号

DFD 的描述符号主要有 4 种: 起点(或终点)、数据流连线、数据加工/处理、输入输出文件,如表 3-5 所示。

表 3-5 DFD 描述符号

名 称	图 例	说 明
起点(或终点)		数据流的起点或终点,表示数据源和数据库
数据加工/处理		表示对流到此处的数据进行加工或处理,即对数据的算法分析与科学计算
输入输出文件		表示输入输出文件,说明加工/处理前的输入文件,记录加工/处理后的输出文件,也可为单线
数据流连线		表示数据流的流动方向

【案例 3-6】 某礼品销售 App 的主要功能是: 系统接收用户的订单,并对订单验证,验证过程主要根据礼品目录检查订单的正确性,并由顾客档案确定新老顾客及信誉情况。验证正确的订单,暂存在待处理的订单文件中。集中后对订单进行成批处理,对于有货的订单将通过短信给用户发送送货信息,对缺货的订单发送供应商。店铺经理可以统计销售情况。该系统的 DFD 图如图 3-6 所示。

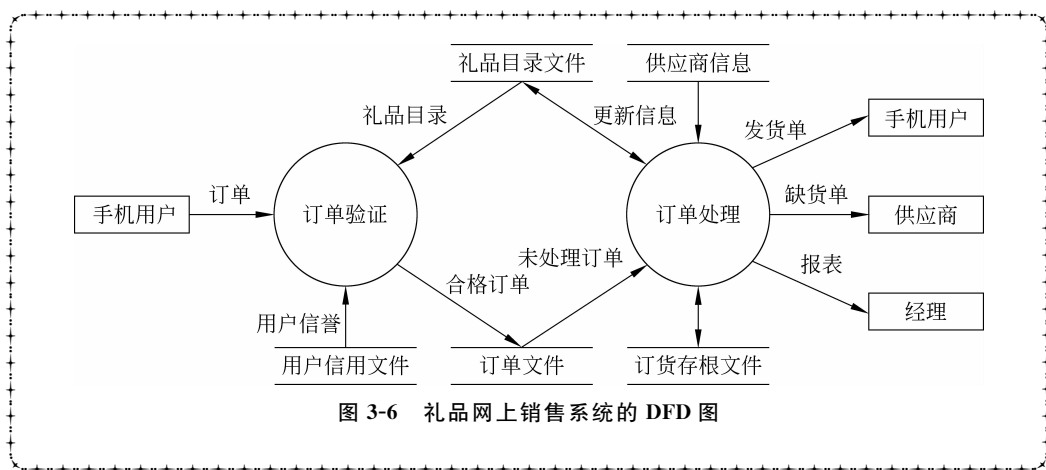


图 3-6 礼品网上销售系统的 DFD 图

有时 DFD 符号表示不尽相同。另一种较为常见的 DFD 有 4 种基本符号元素：外部实体、数据处理、数据流和数据存储。其基本符号如图 3-7 所示。

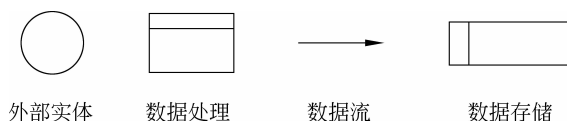


图 3-7 DFD 的另一种基本符号

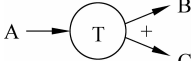
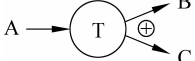
3. 数据流图的附加符号

在数据流图中,有时有两个以上的数据流进入或流出同一个加工,这样,多个数据流之间往往存在一定的关系。为了表示这些数据流之间的关系,需要在数据流图中给这些数据流对应的加工加上一定的标记符号。常见的几种图例的表示方法如表 3-6 所示。

表 3-6 DFD 附加符号

图 例	说 明
	“*”表示数据流之间是“与”关系(同时存在),即数据 A 和 B 同时输入才能变换成 C
	“+”表示数据流之间是“或”关系,即数据 A 或 B 输入,或 A 和 B 同时输入才能变换成 C
	“⊕”表示只能从几个数据流中选一个(互斥关系),即只有数据 A 或只有数据 B(但不能 A、B 同时)输入时变换为 C
	数据 A 变换成数据 B 和 C

续表

图 例	说 明
	数据 A 变换成数据 B 或 C,或同时变为 B 和 C
	数据 A 变换成数据 B 或 C,但不能同时变为 B 和 C

4. DFD 绘制方法

实际上,可用 DFD 表示软件中多个层面的数据处理过程。对于复杂问题的数据处理过程,可用多个 DFD 表示,按照问题的层次结构进行逐步分解,并自顶向下分层绘制其结构关系。**DFD 的制作工具**包括 Microsoft Visio 或 Power Designer 等绘图工具。

(1) 绘制顶层 DFD(系统的输入输出)。将系统视为一个整体,查看整体与外界的联系。分析通过外界获取的数据,即系统输入;向外界提供服务的数据,即系统输出。例如,考试系统的顶层数据流图如图 3-8 所示。

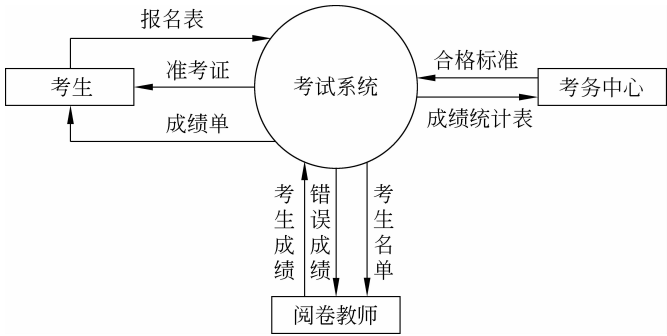


图 3-8 考试系统的顶层数据流图

(2) 绘制系统内部 DFD。DFD 主要是用于描述系统内部的处理过程,即绘制下层 DFD。考试系统的一层数据流图如图 3-9 所示。

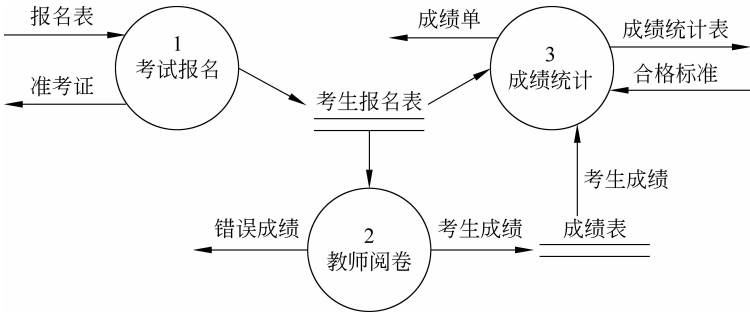


图 3-9 考试系统的一层数据流图

考试系统的二层数据流图如图 3-10 所示。

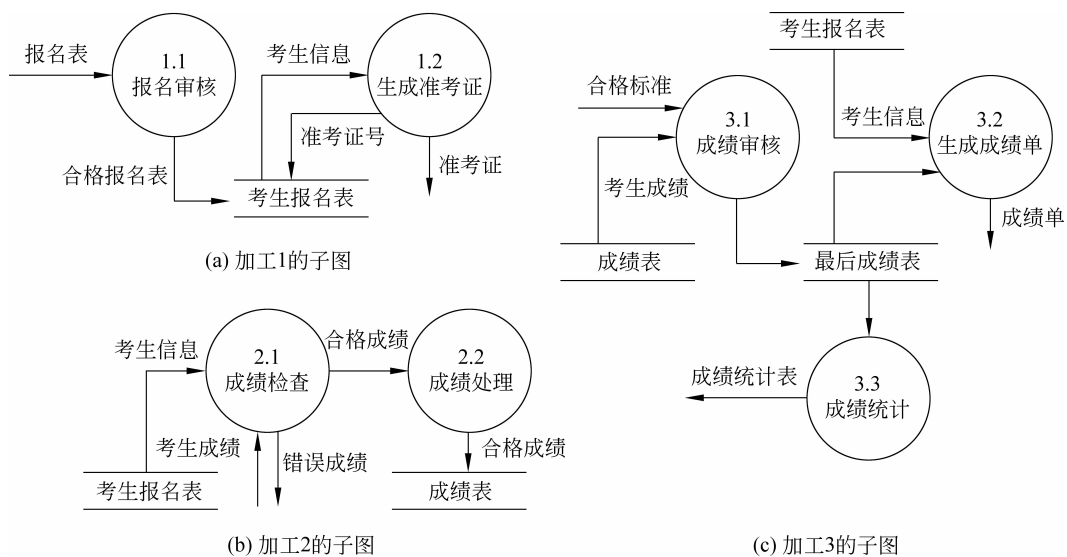


图 3-10 考试系统的二层数据流图

(3) DFD 绘制规则。

- ① 自顶向下、由外向内绘制。注意绘制层次和顺序,用父子图描述不同的层次。
- ② 命名编号。从 0 开始编层号,对数据流命名(含义明显时可略),如图 3-11 所示。

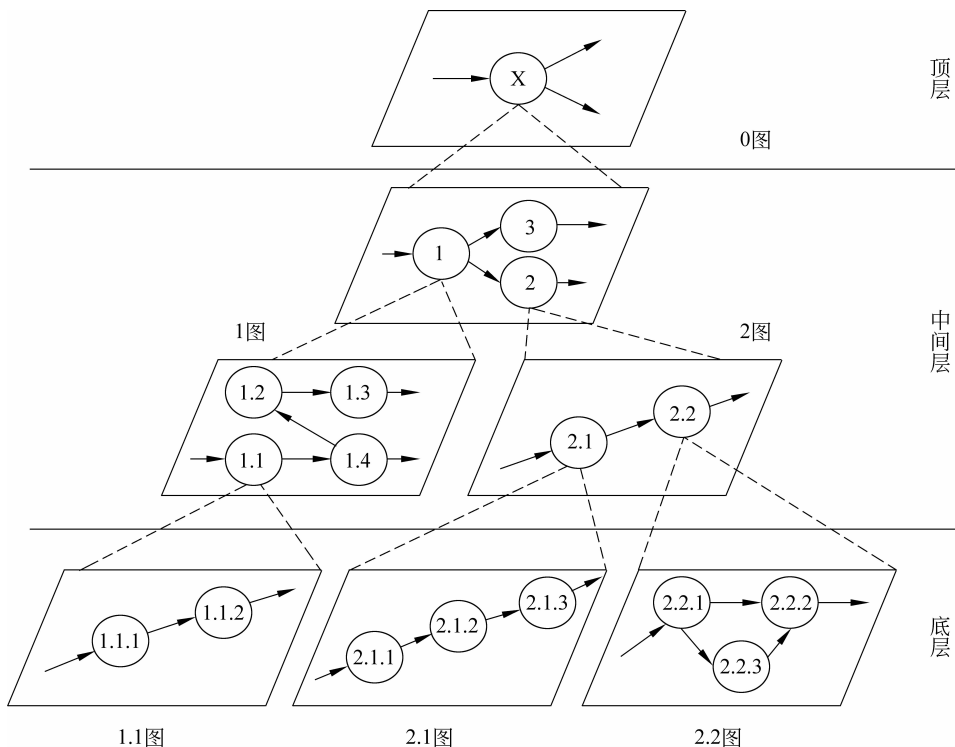


图 3-11 自顶向下分层描述

③ 调整位置,尽量避免数据流的交叉。

④ 对需要在两个设备上进行的处理,应避免直接相连。可在其之间加一个数据存储。

⑤ 如果一个外部实体提供给某一处理的数据流过多,可将其合并成一个综合数据流。

⑥ 下层图中的数据流应与上层图中的数据流等价。

⑦ 对于大而复杂的系统,其图中的各元素应加以编号。通常在编号前冠以字母,表示不同的元素,用 P 表示处理、D 表示数据流、F 表示数据存储、S 表示外部实体。

5. 注意事项

在进行需求分析前,需要制定一个 DFD 的规范标准。绘制 DFD 时要注意以下几点。

(1) 统一编号及命名。统一编号,并对数据流、数据存储或加工的命名应容易理解。如顶层的加工编号为 $1, 2, 3, \dots, n$, 一层的加工编号为 $1.1, 1.2, \dots, 1.n, 2.1, 2.2, \dots, 2.n$, 以此类推。

(2) 不用标出控制关系。在绘制 DFD 时注意不绘制控制流。

(3) 输入输出流。各加工至少有一个输入流和一个输出流,表明数据的来源与去向。

(4) 数据流应当守恒。一个数据流子图应该对应于其父图中的一个加工,两者的输入数据流和输出数据流必须一致。

6. DFD 实际应用示例

【案例 3-7】 某高校“毕业生就业服务系统”的主要用户分为 3 类:就业管理人员、应届毕业生和招聘公司。分为四个子系统。

验证注册子系统。使用角色:应届毕业生、招聘公司、就业管理人员。主要功能:登录验证及公司、学生的注册。

招聘管理子系统。使用角色:招聘公司。主要功能是发布招聘信息、查询学生应聘信息、安排面试通知。

应聘管理子系统。使用角色:应届毕业生。主要功能是利用移动终端实现查询招聘信息、发送应聘信息、查询面试通知。

系统管理子系统。使用角色:就业管理人员。主要功能是负责招聘企业的审核、招聘信息的统计。

该系统的 DFD 采用了“自顶向下,由外向内”的绘制原则,其 3 层部分 DFD 如图 3-12~图 3-14 所示。

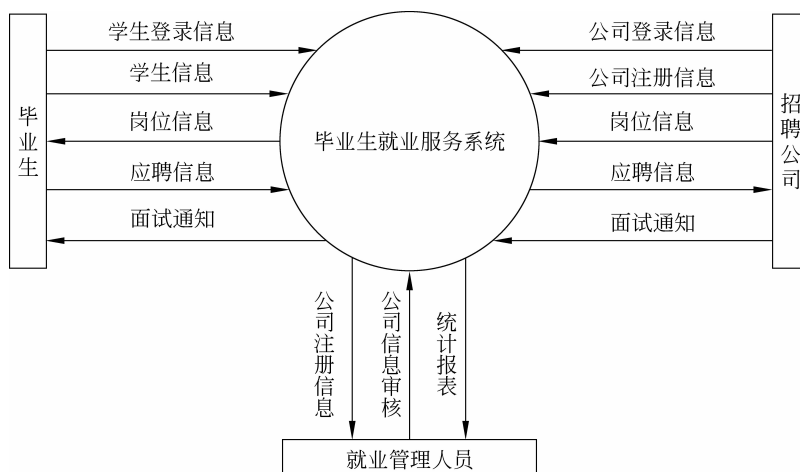


图 3-12 毕业生就业服务系统顶层数据流

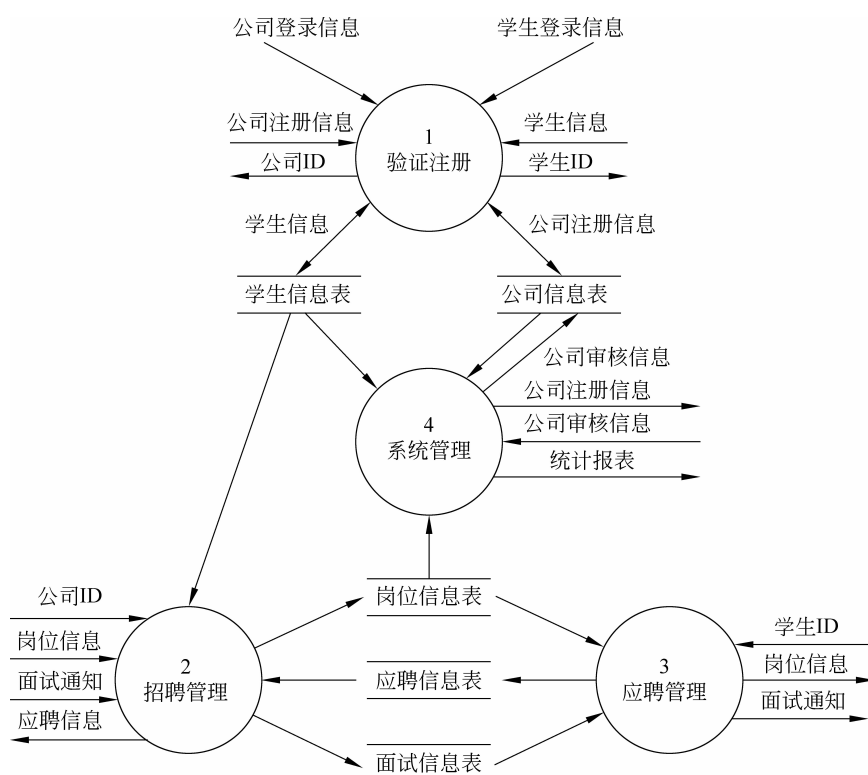


图 3-13 毕业生就业服务系统第一层数据流

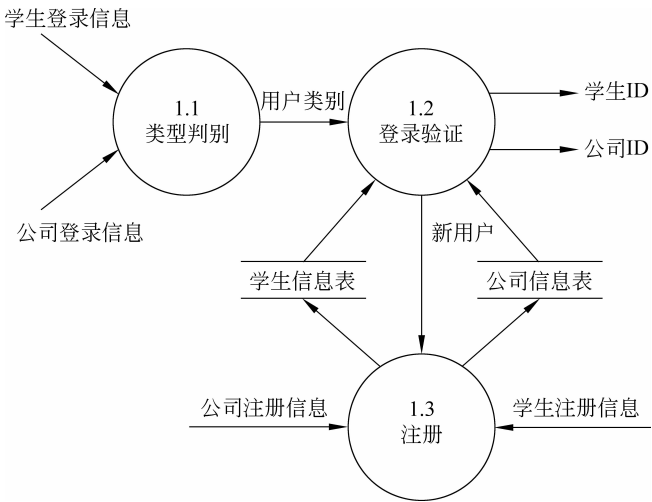


图 3-14 加工 1 的第二层数据流图

3.5.3 数据字典

数据字典(Data Dictionary,DD)是指存储数据源定义和属性(描述说明)的文档,是数据描述的重要组成部分。为了完整地描述系统,只用表达系统“分解”的分层 DFD 还不够,还需借助数据字典和“概要说明”对图中的数据和加工给出解释。对 DFD 中包含的所有元素的定义的集合构成了数据字典。有 4 类条目:数据流、数据项、文件及基本加工。在定义数据流或文件时,可使用表 3-7 给出的符号。将这些条目按照一定的规则组织,构成数据字典。

表 3-7 在数据字典定义中使用的符号

符 号	定 义	举例及说明
=	被定义为	
+	与	$x=a+b$,表示 x 由 a 和 b 组成
$[\dots \dots]$	或	$x=[a b]$,表示 x 由 a 或 b 组成
$\{\dots\}$	重复	$x=\{a\}$,表示 x 由 0 个或多个 a 组成
$m\{\dots\}n$ 或 $\{\dots\}_m^n$	重复	$x=2\{a\}5$ 或 $x=\{a\}_2^5$,表示 x 中最少出现 2 次 a ,最多出现 5 次 a ,5、2 为重复次数的上下限
$(\dots)$	可选	$x=(a)$,表示 a 可在 x 中出现,也可不出现
“...”	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 中任意一个值

DD 以一种准确无二义性的说明方式为软件分析、设计及维护提供了有关数据元素一致的定義和详细描述。数据字典要求完整性、一致性和可用性。

(1) 数据流条目。给出了DFD中数据流的定义,通常对数据流的简单描述为列出该数据流的各组成数据项,主要包括数据流名称、别名及简述、数据流来源和去处、数据流组成、流通量。

【案例 3-8】 “毕业生就业服务系统”中的数据流“招聘信息”条目。

数据流名称: 招聘信息。

别名: 无。

简述: 公司发布招聘信息。

来源: 招聘公司。

去向: 加工 2“招聘管理”。

数据流量: 100 份/每月。

组成: 公司代码+公司名称+招聘岗位+招聘人数+招聘条件+工资待遇。

(2) 文件条目。给出某个文件的定义,文件的定义通常是列出文件记录的组成数据流,还可指出文件的组织方式。

【案例 3-9】 “毕业生就业服务系统”中公司注册信息文件。

数据文件名: 公司信息表。

别名: Company_information。

简述: 存储招聘公司基本信息。

组成: 公司 ID+公司名称+公司性质+注册资金+人员数量+企业效益+公司地址+法人代表+经营范围。

存储方式: 顺序。

组织方式: 以“公司 ID”为关键字。

存取频率: 1000 次/天。

(3) 数据项条目。给出某个数据单项的定义,通常是该数据项的值类型、允许值等。

【案例 3-10】 公司性质数据项。

数据项名称: 公司性质。

别名: Company_category。

简述: 略。

类型: 字符串。

长度: 10。

取值范围: [国有|有限责任|中外合资|外商独资|集体|个体]。