

项目 2 软件需求分析

【教学目标】

1. 了解软件定义过程。
2. 掌握需求获取的过程和主要方法。
3. 了解结构化分析方法。
4. 了解数据流图、数据字典的基本元素和应用。
5. 了解面向对象需求分析方法。
6. 掌握需求分析说明及评审。

【项目简介】

某软件开发公司为某高校开发学生心理动态跟踪评价系统,在软件需求分析阶段产生的软件需求规格说明书的目录清单如图 2-1 所示,具体内容详见文中的软件需求规格说明书部分。

目录	
1	引言
1.1	编写目的
1.2	背景
1.3	定义
1.4	参考资料
2	任务概述
2.1	目标
2.2	用户的特点
2.3	假定与约束
3	需求规定
3.1	对功能的规定
3.2	对性能的规定
3.3	故障处理要求
3.4	其他专门要求
4	运行环境规定
4.1	设备
4.2	软件环境
4.3	接口
4.4	控制

图 2-1 软件需求说明书清单

任务 1 认识软件定义过程

【任务分析】

通过完成认识软件定义过程这个任务,我们可以了解软件定义过程所包括的三个不同阶段的任务,为实际软件项目的定义打下基础。

【知识点分析】

认识软件定义过程这个任务所涉及知识点,包括问题的定义阶段的任务、可行性研究阶段的任务、需求分析阶段的任务等。

2.1.1 问题的定义

软件定义过程一般包括 3 个阶段,即为问题的定义、可行性研究、需求分析。

开发一个软件首先要进行问题定义,其主要工作是弄清楚用户需要计算机解决的根本问题是什么,确定系统的规模和范围,以及项目所需的资源和经费;初步确定软件的总体要求以及所需要的硬件支撑软件,确定待开发软件与外界的接口、根据用户的情况确定软件对操作的要求,以及待开发软件总体上的约束和限制。系统分析员将问题定义阶段获得的结果以明确的文字整理成文档,并不断地与用户进行协商对话。该文档可作为下一步工作——可行性研究的依据。

2.1.2 可行性研究

可行性研究的任务是从技术上、经济上、使用上、法律上分析应解决的问题是否有可行的解,从而确定该软件是否值得去开发。其目的是用极少的代价在最短的时间内确定被开发的软件是否能开发成功。

(1) 技术可行性

技术上的可行性主要是根据系统分析得到的对需开发的软件、支撑软件、硬件环境和操作人员的要求,以及有关的约束和限制,来分析利用现有的技术是否能实现待开发的软件。它包括可得到的硬件和支撑软件在功能和性能上是否满足系统的要求,是否有满足系统性能要求的算法,开发人员的技术水平能否胜任系统的开发等。

(2) 经济可行性

经济可行性首先要进行待开发软件的成本估算和成本/效益分析,以确定待开发软件是否有开发的价值。这里的效益包括新开发的系统所带来的收入增加,以及新开发的系统比原有系统在使用费用上的减少。对于开发成本低、经济效益高的软件应积极开发,而对于开发成本高,经济效益低的软件或开发成本与经济效益差不多的软件通常不应开发。

(3) 使用可行性

使用可行性是指使用方式(如操作方式)用户是否容易接受。对于一个运行方式难以让人习惯的软件,用户是不会满意的。

(4) 法律可行性

法律可行性是指待开发的软件是否存在知识产权等法律有关的问题,如有这类问题,该软件即使开发成功,也难以作为产品销售。

在进行可行性分析时,通常要先研究目前正在使用的系统,然后根据待开发系统的要求导出新系统的高层逻辑模型。有时可提出几个选择的方案,并对每个方案从技术上、经济上、使用上、法律上进行可行性分析,在对各方案进行比较后,选择其中的一个作为推荐方案(有时可能要在几个方案中进行折衷),最后对推荐方案给出一个明确的结论,如“可行”、“不可行”,或“等某某条件成熟后可行”。

2.1.3 需求分析

需求分析是软件生命周期中的一个相当重要的阶段,它是软件定义时期的最后一个阶段,它准确回答了“系统必须做什么”的问题。可行性研究阶段产生的文档是需求分析阶段的出发点,在需求分析阶段系统分析员必须将可行性研究阶段确定的系统功能进一步具体化,通过开发人员与用户之间的广泛交流,最终形成一个完整、清晰、一致的需求说明。需求分析的好坏直接影响到所开发软件的成败。

需求分析的任务是借助当前系统的逻辑模型导出目标系统的逻辑模型,具体任务包括以下几个方面:

(1) 确定对系统的综合需求。

软件系统的综合需求包括:系统的界面需求(即描述系统从外部输入哪些数据,系统向外部输出哪些数据)、系统的功能要求、系统的性能要求(如响应时间、吞吐量、处理时间、对内存和外存的限制等)、安全性、保密性、可靠性方面的要求、系统的运行要求(如对硬件、支撑软件、数据通信接口的要求等)、异常处理要求等。

(2) 分析系统的数据需求

分析系统的数据需求是由系统的信息流归纳抽象出数据元素组成、数据的逻辑关系、数据字典格式和数据模型,并以输入/处理/输出(IPO)的结构方式表示。因此,必须分析系统的数据需求,这是软件需求分析的一个重要任务。

(3) 导出系统的逻辑模型

就是在理解当前系统“怎样做”的基础上,抽取其“做什么”的本质。

(4) 修正系统开发计划

明确了用户的真正需求后,可以更准确地估算软件的成本和进度,从而修正系统开发计划。

(5) 开发原型系统

对一些需求不够明确的软件,可以先开发一个原型系统,以验证用户的需求。

任务 2 需求获取的过程与方法

【任务分析】

通过完成需求获取的过程与方法这个任务,我们可以了解软件系统需求获取的过程和

软件需求获取的主要方法,为实际软件项目开发过程中的需求获取打下基础。

【知识点分析】

需求获取的过程与方法这个任务所涉及知识点包括需求获取的过程、需求获取的主要方法等。

2.2.1 需求获取的过程

(1) 问题识别

首先,系统分析员要确定用户对目标系统的综合要求,即对软件的需求,并提出这些需求实现的条件,以及需求应达到的标准。这些需求包括对功能、性能、环境、可靠性、安全保密、用户界面、资源使用、软件成本消耗与开发进度的需求,系统分析人员还应预先估计终级系统可能达到的目标。此外,还应注意其他非功能性的需求。

(2) 分析与综合

分析员必须从信息流和信息结构出发,逐步细化软件的所有功能,找出系统各元素之间的联系、接口特性和对设计的限制,判断是否存在因片面性或短期行为而导致的不合理需求,判断是否有用户尚未提出的确实有价值的潜在要求,从而剔除其中的不合理部分,增加真正需要的部分。

在分析与综合时应注意下述两条原则:

第一,在分层细化时必须保持信息连续性,也就是说细化前后对应功能的输入/输出数据必须相同;

第二,当进一步细化时将涉及如何具体地实现一个功能,并将考虑为了完成这些子功能而写出其程序代码时,就不应该再分解了。

(3) 编制需求分析阶段的文档

已经确定的需求应当得到清晰、准确的描述,通常把描述需求的文档称为软件需求说明书。同时,为了确切地表达用户对软件的输入、输出数据的要求,还需要制定数据要求说明书并编写初步的用户手册。

(4) 需求分析评审

为了对需求分析阶段工作进行复查,应该对软件功能的正确性、软件需求说明书的一致性、完备性、准确性和清晰性以及其他的的需求给予评价。评审由专门指定的人员负责,并按规程严格进行。一般说来,软件需求分析评审的主要内容应该从下述四个方面进行验证:

- ①一致性。所有需求必须是一致的,任何一条需求不能和其他需求互相矛盾。
- ②完整性。需求必须是完整的,规格说明书应该包括用户需要的每一个功能或性能。
- ③现实性。指定的需求应该是使用现有的硬件技术和软件技术基本上可以实现的。
- ④有效性。必须证明需求是正确有效的,确实能解决用户所需的问题。

2.2.2 需求获取的方法

获取用户需求的主要方法是调查研究。

(1) 了解系统的需求

软件开发常常是系统开发的一部分,仔细分析研究系统的需求规格说明,对软件的需求

获取是很有必要的。

(2) 市场调查

了解市场对等开发软件有什么要求,了解市场上有无与待开发软件类似的系统。如果有相似的产品、在功能上、性能上、价格上情况如何等。

(3) 访问用户和用户领域的专家。

把从用户那得到的信息作为重要的原始资料进行分析,访问用户领域的专家所得到的信息将有助于对用户需求的理解。

(4) 考察现场

了解用户实际的操作环境、操作过程和操作要求,对照用户提交的问题陈述,对用户需求可以更全面、更细致地认识。

在做调查研究时,可以采取如下的调查方式:

- ① 制定调查提纲,向不同层次的用户发调查表。
- ② 按用户的不同层次,分别召开调查会,了解用户对等开发系统的想法和建议。
- ③ 向用户领域的专家或在关键岗位上工作的人进行个别咨询。
- ④ 实地考察、跟踪现场业务流程。
- ⑤ 查阅与待开发系统有关的资料。
- ⑥ 使用各种调查工具,如数据流图、任务分解图、网络图等。

为了能够有效地获取和理清用户需求,应当打破用户和开发者的界限,组成一个联合分析小组,发挥各自的优势,协同工作。

在学生心理动态跟踪评价系统的需求获取过程中除向不同层次的用户下发用户调查表、客户访谈及实地考察外,还建立了一个由用户、系统分析员和领域专家参加的联合分析小组,极大地方便系统开发和用户沟通。

任务3 软件需求分析技术

【任务分析】

通过完成软件需求分析技术这个任务,我们可以了解在系统的开发过程中可采取的主要软件需求分析技术,为实际软件项目的分析做准备。

【知识点分析】

软件需求分析技术这个任务所涉及知识点包括结构化分析方法、数据流图、数据字典、面向对象需求分析方法等。

2.3.1 结构化分析方法

软件需求分析的目标是在系统模型分析的基础上,建立软件需求规格说明书,因此软件需求分析的关键在于对系统模型的分析。对于比较简单的问题,可以不必给出抽象系统模型,这样的模型存在于系统分析员的头脑中,对于比较复杂的问题,由于问题域各方面的关

联比较多,仅有这种脑子中想象的模型是不足够的,因而必须建立比较形式的抽象模型,才能准确地反映问题领域的各种复杂问题。系统模型是软件系统的抽象表示,不同的系统模型抽象导致不同风格的软件需求规格说明。

目前有两种主要的建立系统模型的方法,一种是基于数据流的系统模型,将系统分解为若干功能处理块,各块之间以数据流进行通讯,另一种是基于对象的系统模型,系统分解为一组互相关联的对象,每一对象由对象属性和在对象上的操作组成。

结构化分析(Structured Analysis)方法简称 SA 方法,是一种面向数据流的需求分析方法,适用于分析大型数据处理系统,是一种现已得到广泛使用的简单、实用的方法。

结构化分析方法的基本思想是自顶向下逐层分解。分解和抽象是人们控制问题复杂性的两种基本手段。对于一个复杂的问题,人们很难一下子考虑问题的所有方面和全部细节,通常可以把一个大问题分解成若干个小问题,每个小问题再分解成若干个更小的问题,经过多次逐层分解,每个最底层的问题都是足够简单、容易解决的,于是复杂的问题就迎刃而解了,这个过程就是“分解”。“分解”并不是等分,不能像厨师们分蛋糕那样每人等分一块,它要根据系统的逻辑特性和系统内部各成分之间的逻辑关系进行分解。在分解中要充分体现“抽象”的原则,逐层分解中的上一层就是下一层的抽象。系统的抽象模型应该按照一定的层次关系组织而成。

对于任何的复杂的系统,结构化分析工作都可以按照逐层分解的方式有计划、有步骤、有条不紊地进行,对大小规模不同的系统只是分解的层次不同而已。结构化分析方法的分析结果由以下几个部分组成,一套分层的数据流图、一本数据字典、一组加工逻辑说明(也叫小说明)、补充材料。

2.3.2 数据流图

数据流图(Data Flow Diagram, DFD)是一种描述数据流和加工的图形表示,当数据输入到系统后,经过一系统的变换(或加工),最后输出新的数据。DFD的基本形式如图 2-2 所示。

(1) DFD 的基本成分

DFD 的基本成分及其图形表示方式如图 2-3 所示。

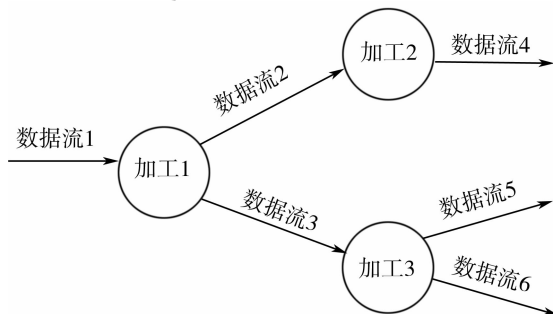


图 2-2 DFD 的基本形式

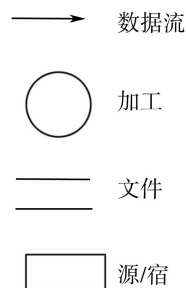


图 2-3 DFD 的基本成分

①数据流(Data Flow)

数据流由一组固定成分的数据组成,表示数据的流向。它可以从一个加工流向另一个加工,从加工向文件,从文件流向加工,从源流向加工,从加工流向宿。除了流向文件或从文

件流出的数据流不必命名(因为这种数据流的组成成分就是相应文件的组成成分)外,每个数据流都必须有一个合适的名字,以反映该数据流的含义。

②加工(Process)

加工,又称为处理,它描述了输入数据流到输出数据流之间的变换,也就是输入数据流做了什么处理后变成了输出数据流。每一个加工有一个名字和编号。编号能反映出该加工位于分层 DFD 中的哪个层次和哪张图中,也能看出它是哪个加工分解出来的子加工。

在数据流图中,如果有两个以上的数据流指向一个加工,或是从一个加工中引出两个以上的数据流,这些数据流之间往往存在一定的关系。为表示这些关系,用“*”表示相邻的一对数据流同时出现;用“ \oplus ”表示相邻的两数据流只取其一。图 2-4 所示为表明多个数据流同加工之间关系的符号。

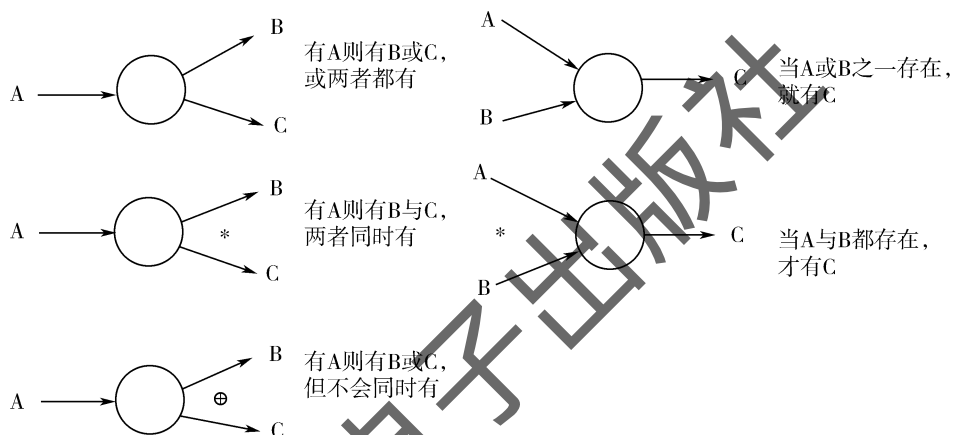


图 2-4 表明多个数据流同加工之间关系的符号

③文件(File)

文件用来表示暂时存储的数据,每个文件都有一个名字。流向文件的数据流表示写文件,流出文件的数据流表示读文件,双向箭头表示对文件既读又写。

④源/宿(Source/Sink)

源和宿通常是指存在于软件系统之外的人员或组织。它指出系统所需数据的发源地和系统所产生的数据的归宿地。在一个软件系统中,有些源和宿可以是同一个人或组织,这时源和宿使用同一个图形符号。

(2)分层数据流图的画法

①画系统的输入和输出

把整个软件系统看作一个大的加工,然后根据系统从外界的哪些源接受哪些数据流,以及系统的哪些数据流送到外界的哪些宿,就可画出系统的输入和输出图,这张图称为顶层图。

②画系统的内部

将顶层图中的加工分解成若干个加工,并用数据流将这些加工连接起来,使得顶层图中的输入数据流经一连串的加工处理后转换成顶层的输出数据流。这张图称为 0 层图。从一个加工画出一张数据流图的过程实际上就是对这个加工的分解过程。

可用下述的方法确定加工:在数据流的组成或值发生变化的地方画一个加工,这个加工

的功能就是实现这一变化;也可根据系统的功能确定加工。

确定数据流的方法可以是:当用户把若干数据看作一个单位来处理(这些数据一起到达,一起加工)时,可把这些数据看成一个数据流。通常可以把实际工作中的单据(如报名表)作为一个数据流。

对于一些以后某个时间要使用的数据可以组织成一个文件存储。

③对图和加工进行编号

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 DFD 子图,我们应该对每张图和每个加工进行编号,以利于管理。

④注意要点

◆命名

恰当地为数据流、加工、文件、源/宿命名,名字应能够反映各成分的实际含义,避免使用空洞的名字(如使用数据、信息、计算、处理等名字不好,而报名表、计算工资等较好);一个加工的输出流不应与输入数据流同名,即使它们的组成成分相同。

◆数据流与控制流

值得注意的是 DFD 中描述的是数据流,而不是控制流,它们的区别在于数据流中有数据,而控制流中没有数据。

◆加工

允许一个加工有多条数据流流向另一个加工,也允许一个加工有两个相同的输出数据流流向两个不同的加工;每个加工必须既有输入数据流,又有输出数据流。

◆文件

在自顶向下的分解过程中,若一个文件首次出现时只与一个加工有关,那么这个文件应作为这个加工的内部文件而不必画出。

◆保持父图与子图的平衡

父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同。如果父图的一个输入(或输出)数据流对应于子图中的几个输入(或输出)数据流,而子图中组成这些数据流的数据项的全体正好是父图中的一个数据流,那么它们仍然算是平衡的。

◆保持数据守恒

一个加工所有输出数据流的数据必须能从该加工的输入数据流中直接获得,或者是通过该加工能产生数据。

2.3.3 数据词典

数据词典(Data Dictionary, DD)是结构化分析的另一有力的工具。数据流图描述了系统的分解,但没有对图中各成分进行说明。数据词典就是为数据流图中的每个数据流、文件、加工,以及组成数据流或文件的数据项做出说明。其中对加工的描述称为“加工逻辑说明”,又称为“小说明”。

(1) 词典条目

① 数据流条目

一个数据流条目可以有以下内容:数据流名、编号(用于词典管理,如排序、查找)、简述(简述该数据流的含义)、组成(描述数据流由哪些数据项组成)、来源(描述该数据流来自哪个加工源)、去向(描述该数据流向哪个加工或宿)、数据量(描述在软件系统中该数据流的数量,也可以是某段时间里的数据量,如每天多少)、峰值(描述在某一时间范围内要处理的数据流的最大数量,它是设计时考虑处理速度的依据之一)等。

在这些内容中数据流名和组成是必不可少的,数据量和峰值一般也是需要的,其他内容则是可有可无的。

词典条目的定义式可使用以下符号:

$x = a + b$	表示 x 由 a 与 b 构成
$x = [a b]$ 或 $x = [a, b]$	表示 x 由 a 或 b 构成
$x = \{a\}$	表示 x 由零次或多次重复的 a 构成
$x = (a)$	表示 a 是任选的,即 a 可在 x 中出现 0 次或 1 次
$x = a..b$	表示 x 可取 a 至 b 中的任一值
$x = m\{a\}n$	表示 x 由 m 至 n 个 a 构成,至少 m 个 a ,至多 n 个 a
$x = "a"$	表示 a 是一个基本元素, x 的取值为 a

② 文件条目

一个文件条目可以有如下的内容:文件名、编号、简述(简述该文件的含义)、组成(描述该文件由哪些数据项组成)、文件组织(描述该文件的类型、按什么关键字排序等)、读文件(描述哪些加工需读该文件)、写文件(描述哪些加工需写该文件)、数据量(描述该文件的记录个数)、注解等。

其中文件名和组成是必须的,文件组织和数据量通常也是需要的,其他内容则是可有可无的。文件的组成可以使用与数据流组成相同的符号。

③ 数据项条目

一个数据项条目可以有如下内容:数据项名、编号、简述(简述该数据项的含义)、单位(该数据项的计量单位)、类型(描述该数据项的数据类型)、值域(描述该数据项的取值范围)、编辑方式(描述该数据项在输出时的编辑要求,如每三位数字前加一个逗号、与其他数据项的关系,有利于数据的合法性检查)、注解。

其中数据项名、类型都是必须的,单位、值域、编辑方式、与其他数据项的关系通常也是需要的,其他内容则是可有可无的。

④ 加工条目(加工逻辑说明)

在数据流图中,有些加工最终分解成许多足够简单的加工,我们把这些不再分解的加工称之为基本加工。为了理解基本加工,我们可以利用加工条目(加工逻辑说明,也可称为小说明)来进一步描述这个加工在什么条件下做什么事。数据词典中只列出基本加工的加工条目,而其他加工则没有加工条目,这是因为任何一个加工最后终能分解成一些基本加工,

只要有了基本加工的加工条目就可以理解其他加工。

一个基本加工的加工条目包括以下内容:加工名、编号、简述(对该加工做一简要说明)、输入数据流(描述该加工有哪些输入数据流)、输出数据流(描述该加工有哪些输出数据流)、输入文件(描述该加工有哪些输入数据流)、输出文件(描述该加工需要写哪些文件)、加工逻辑(描述该加工在什么条件下做什么事)、异常处理(描述该加工可能遇到哪些异常情况,并说明在这些异常情况出现时该加工应做出怎样的反应)、加工激发条件(描述该加工什么情况下被激发)、执行频率、注解等。

其中加工逻辑是最基本的部分,它描述了输入数据流、输入文件与输出数据流、输出文件之间的逻辑关系。

(2) 词典管理

词典管理主要把词典条目按某种格式组织后储存在词典中,并提供排序、查找、统计等功能。在数据词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之间的通信,可以消除许多可能的误解。如果要求所有开发人员都根据公共的数据词典描述数据和设计模块,则能避免许多麻烦的接口问题。

(3) 加工逻辑的描述方法

常用的加工逻辑的描述方法有三种:结构化语言、判定表、判定树。

① 结构化语言

用自然语言来描述加工逻辑是最方便的,但是自然语言往往不够精确,它可能存在二义性,而且难以用计算机处理。而形式化语言能够精确地描述事物,且易于计算机处理,但不易被用户理解。因此可以采用一种结构化语言(如结构化英语),它是介于自然语言与形式语言之间的半形式化语言,是自然语言的一个受限子集。

结构化语言的语法分为内外两层,外层语法描述操作的控制结构,如顺序、选择、循环等,这些控制结构将加工中的各个操作连接起来,内层的语法一般没有什么限制。

② 判定表

判定表用来描述一些不易用语言表达清楚或需要很大篇幅才能用语言表达清楚的加工逻辑。判定表由四个部分组成,见表 2-1。

表 2-1

判定表

条件所指对象	各种条件的组合
所有的操作	在对应的条件组合下,某个操作是否要执行

③ 判定树

判定树与判定表无本质的区别,它是另一种表示形式。

2.3.4 典型案例

(1) 案例 1

某银行的计算机储蓄系统功能是:将储户填写的存款单输入系统,如果是存款,系统记录存款人姓名、住址、存款类型、存款日期、利率等信息,并打印出存款单给储户;如果是

取款,系统计算清单给储户。请用 DFD 描绘该功能的需求,并建立相应数据字典。

①建立储蓄系统的 DFD

图 2-5 所示为储蓄系统的 DFD。

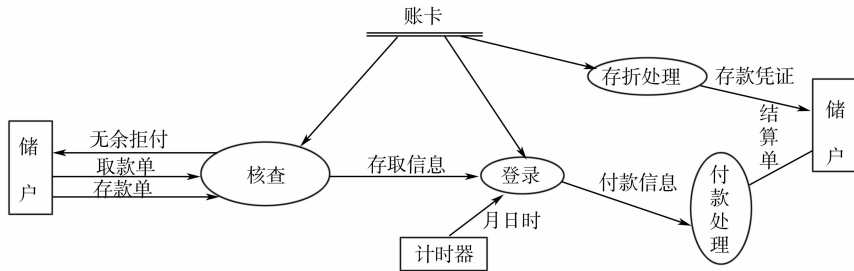


图 2-5 储蓄系统的 DFD

②数据字典

数据流名称:取款单、存款单

别名:无

简述:顾客存/取款时填写的单据

来源:顾客

去向:核查

数据流量:200 份/天

组成:日期+[取|存款金额]+姓名+地址+[存/取款类型]+密码+{账号}

数据流名称:存款凭证

别名:无

简述:顾客存款后得到的凭据

来源:银行存款系统

去向:顾客

数据流量:200 份/天

组成:日期+存款金额+储户姓名+储户地址+存款类型+账号+利率+操作员

数据流名称:结算单

别名:无

简述:顾客取款时得到的利率清单

来源:银行储蓄系统

去向:顾客

数据流量:200 份/天

组成:取款日期+取款金额+储户姓名+储户地址+存款类型+利息+总金额+操作员

数据流名称:存/取款信息

别名:无

简述:系统检查确认后得到的存/取信息

来源:银行储蓄系统

去向:登录

数据流量:200 份/天

组成:日期+[存|取款金额]+储户姓名+储户地址+存款类型+账号+利率+密码+操作员

数据流名称:付款信息

别名:无

简述:存储系统处理的取款信息

来源:银行储蓄系统

去向:付款处理

数据流量:200 份/天

组成:日期+取款金额+储户姓名+储户地址+存款类型+账号+利息+总金额+操作员

数据存储名称:账卡

别名:无

简述:存放用户信息及存取款情况

组成:储户姓名+储户地址+账号+|日期|+[存/取款金额]+存款类型+利率+密码+操作员

组织方式:数据文件,以储户姓名或账号为关键字进行索引

查询要求:要求能立即查询并修改

数据项名称:存款类型

别名:无

简述:国家规定的几类存款方式

类型:字符串

长度:2 位

取值范围及含义:00:定期;01:零存整取;10:活期;11:定/活两便

加工名:核查

编号:

激发条件:接受存款单或取款单

输入:存款单或取款单

输出:认定的合格的存/取款信息

加工逻辑:根据账卡及用户存/取款单据

if 单据 = 存款单 then

存款单据是否填写规范,是否与现金一致

else

审核是否为有效取款单,取款金额不得超出存款金额,密码是否正确

(2) 案例 2

某厂对部分职工重新分配工作的政策是:年龄在 20 岁以下者,初中文化程度脱产学习,高中文化程度当电工。20 岁至 40 岁之间,中学文化程度,男性当钳工,女性当车工,大学文化程度都当技术员。年龄在 40 岁以上者,中学文化程度当材料员,大学文化程度当技术员。请用结构化语言、判定表及判定树描述上述问题的加工逻辑。

①结构化语言表示

```

CASE 年龄 < 20
    CASE 文化程度 初中
        分配工种 脱产学习
    CASE 文化程度 高中
        分配工种 电工
ENDCASE
CASE 年龄 <= 40 . AND . >= 20
CASE 文化程度 初中 . OR . 高中
    CASE 性别 男
        分配工种 钳工
    CASE 性别 女
        分配工种 车工
ENDCASE
ENDCASE
CASE 文化程度 大学
    分配工种 技术员
ENDCASE
ENDCASE
CASE 年龄 > 40
    CASE 文化程度 初中 . OR . 高中
        分配工种 材料员
    CASE 文化程度 大学
        分配工种 技术员
ENDCASE
ENDCASE
    
```

②判定表

表 2-2 所示为判定表。

表 2-2 判定表

条件名	取值	符号	取值数
年龄	年龄 < 20	C	M1 = 3
	20 ≤ 年龄 ≤ 40	Y	
	年龄 > 40	L	
性别	男	M	M2 = 2
	女	F	
文化程度	初中	J	M3 = 3
	高中	S	
	大学	V	

③判定树

图 2-6 所示为判定树。

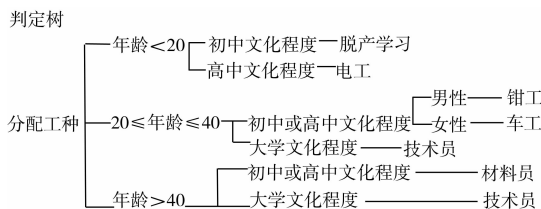


图 2-6 判定树

2.3.5 面向对象的需求分析方法

Coad 和 Yourdon 的面向对象分析和设计方法(OOA&OOD)是一种比较清晰、简明的方法。分析的过程是提取系统需求的过程,主要包括理解、表达和验证。在面向对象分析中,主要由对象模型、动态模型和功能模型组成。

OOA 有两个任务。一是形式地说明所面对的应用问题,最终成为软件系统基本构成的对象,还有系统所必须遵从的、由应用环境所决定的规则和约束。二是明确地规定构成系统的对象如何协同合作,完成指定的功能。

(1) 分析模型

面向对象建模得到的模型包含对象的三个要素,即静态结构(对象模型),交互次序(动态模型)和数据变换(功能模型)。

在面向对象的分析中,有 5 种活动,对应着以下 5 个层次:

- ① 识别对象
- ② 识别类的结构
- ③ 确定主题
- ④ 定义属性
- ⑤ 定义服务

(2) OOA 主要表示法

① 类和对象

对象是问题域中事物的抽象或者是问题域中事物的实现的抽象,它是属性值及其相应服务的一种封装,对象也称为实例;类是一个或多个对象的描述;“类和对象”指的是一个类和属于该类的对象。

在图形中,类和对象的符号表示如图 2-7 所示,粗方框表示类,类分 3 个区域,对象用围绕粗框的细框来表示。

单纯的类的符号如图 2-8 所示。

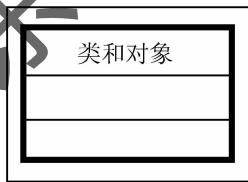


图 2-7 类和对象的符号表示



图 2-8 类的符号表示

② 结构

结构是与系统任务有关的问题域复杂性的一种表示,它是类层次结构的统称,即描述了一般/具体结构(即分类结构),也描述整体/部分结构(即组装结构)。

分类结构是一种“is a”结构。在图形中,分类结构的符号如图 2-9 所示,一般类放在顶端,具体类放在下端,它们之间用连线连起来,其中半圆弧说明这是一个一般/具体结构。

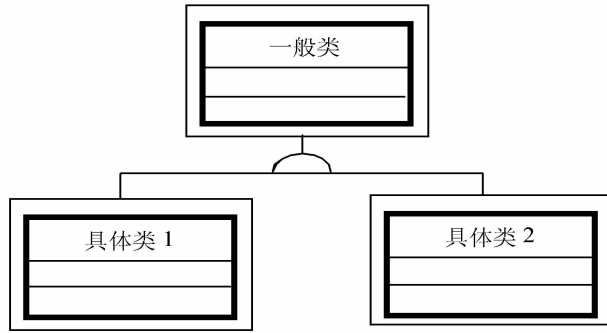


图 2-9 分类结构的符号表示

组装结构是“has a”结构。组装结构的符号如图 2-10 所示,整体类在顶端,成员类在底端,用线把它们连起来,三角形表示这是一个整体/部分结构。

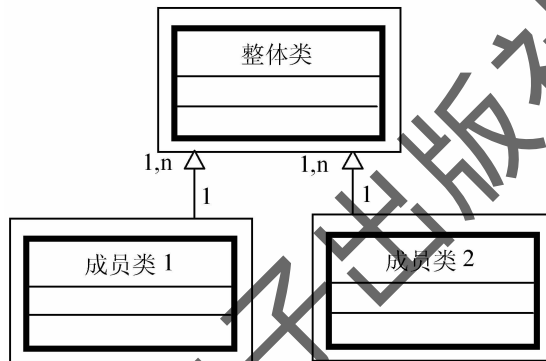
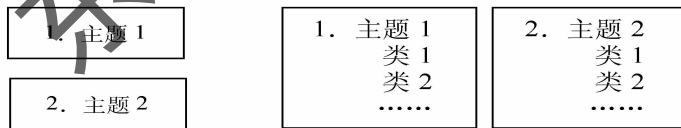


图 2-10 组装结构的符号表示

③主题

主题是一种指导读者或用户研究大型复杂模型的机制。在初步面向对象分析的基础上,主题有助于分解大型项目以便成立工作小组承担不同的主题。

在图形上,主题有两种表示形式:简单表示和扩展表示,如图 2-11 所示。



(a)主题简单表示

(b)主题扩展表示

图 2-11 主题的符号表示

④属性及实例连接

属性是类的状态信息。

实例连接是一个实例集合到另一个实例集合的映射,表示对象间的依赖关系。

在图形中,属性在类和对象的第二个区域中表示。实例连接表示为一条直线,在线上标记各对象的重数(Multiplicity),如图 2-12 所示。



图 2-12 实例连接的符号表示

⑤ 服务及消息连接

服务表示类的动态特征。

消息连接用来表示对象间的通信。

在图形中,服务在类和对象的第三个区域中表示。消息连接表示为一条带箭头的直线,箭头由发送对象指向接收对象,如图 2-13 所示。

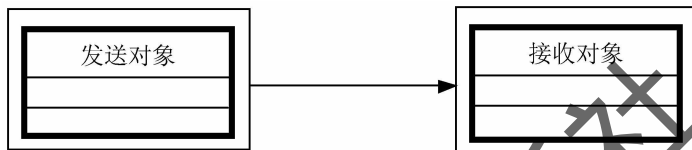


图 2-13 消息连接的符号表示

(3) OOA 的主要原则

OOA 是面向对象的软件开发过程中直接接触问题域的阶段,尽可能全面地运用抽象、封装、继承、分类、聚合、关联、消息通信、粒度控制、行为分析等这些原则完成高质量、高效率的分析。

① 抽象

系统中的对象是对现实世界中事物的抽象;

类是对象的抽象;

一般类是对特殊类的进一步抽象;

属性是事物静态特征的抽象;

服务是事物动态特征的抽象。

② 分类

分类就是把具有相同属性和服务的对象划分为一类,用类作为这些对象的抽象描述。

在 OOA 中所有的对象都是通过类来描述的。对属于同一个类的多个对象并不进行重复的描述,而是以类为核心来描述它所代表的全部对象。运用分类原则也意味着通过不同程度的抽象而形成一般/特殊结构(又称分类结构),一般类比特殊类的抽象程度更高。

③ 聚合

聚合的原则是把一个复杂的事物看成若干比较简单的事物的组装体,从而简化对复杂事物的描述。

在 OOA 中运用聚合原则就是要区分事物的整体和它的组成部分,分别用整体对象和部分对象来进行描述,形成一个整体/部分结构,以清晰地表达它们之间的组成关系。

④ 消息通信

这一原则要求对象之间只能通过消息进行通信,而不允许在对象之外直接地存取对象内部的属性。通过消息进行通信是由于封装原则而引起的。在 OOA 中要求用消息连接表示出对象之间的动态联系。

⑤ 粒度控制

人们在研究一个问题域时既需要微观的思考,也需要宏观的思考。在 OOA 中运用粒度控制原则就是引入主题的概念,把 OOA 模型中的类按一定的规则进行组合,形成一些主题,如果主题数量仍较多,则进一步组合为更大的主题。这样使 OOA 模型具有大小不同的粒度层次,从而有利于分析员和读者对复杂性的控制。

⑥ 行为分析

现实世界中事物的行为是复杂的。由大量的事物所构成的问题域中各种行为往往相互依赖、相互交织。控制行为复杂性的原则有以下几点:

- ◆ 确定行为的归属和作用范围;
- ◆ 认识事物之间行为的依赖关系;
- ◆ 认识行为的起因,区分主动行为和被动行为;
- ◆ 认识系统的并发行为;
- ◆ 认识对象状态对行为的影响。

(4) OOA 过程

OOA 过程包括以下主要活动。

① 发现对象、定义它们的类。

② 识别对象的内部特征。

◆ 定义属性;

◆ 定义服务。

③ 识别对象的外部关系。

◆ 建立一般/特殊结构;

◆ 建立整体/部分结构;

◆ 建立实例连接;

◆ 建立消息连接。

④ 划分主题,建立主题图。

⑤ 建立详细说明。

这是对模型的详细定义与解释,可以作为一个独立的的活动,更自然的做法是分散在其他活动之中。

⑥ 原型开发。

这项活动可在 OOA 过程中反复进行。

2.3.6 面向对象常用的图形工具

1. 用例图

用例图为需获取和建模提供了一种有效的方法,可以作为分析建模的基础,用例图描述了软件系统和外部角色之间的交互关系。通过用例图可以自然地描绘出系统的边界。用例图的组成元素有角色、用例和关联。角色是可以与系统进行信息交换的任意对象,包括人员、软件、硬件、组织机构等,一个单一的用例可能包括与完成某些任务逻辑相关的一系列任务。用例图的图形符号如图 2-14 所示。其中,用大矩形框表示系统,在框的上方标注系统的名称;系统内包含一个或多个用例,一个用例用椭圆表示,椭圆内标注用例的名称;角色用

一个人形图形表示,画在表示系统的矩形外面;角色与用例之间的关联用直线段表示,它们之间的关系可以是多对多的。

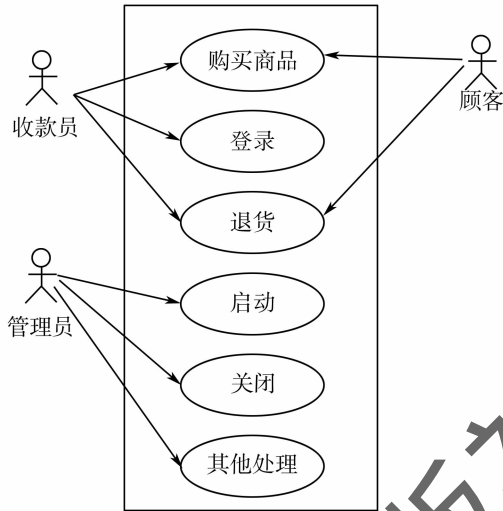


图 2-14 POS 系统用例图

2. 类图

类图用来描述类和它们之间的关系。类用矩形框表示。矩形框一般分成三个区域,上面的区域标注类的名字,中间区域标识该类的属性,下面区域标识类的操作,如图 2-15 所示。



图 2-15 类图

其中,类的名字是一个字符串,用于区分其他的类,给类命名时最好能够反映类所代表的问题域中的概念;属性用来描述该类的对象所具有的特征;操作描述对数据的具体处理方法。

3. 状态图

状态图也称为状态转换图,主要用来描述对象子系统的生命周期,通过状态图可以了解到一个对象所能到达的所有状态以及对象收到的事件(收到消息、超时、错误条件满足)对对象状态的影响等。对所有的类而言,只要它有可标记的状态和复杂的行为都应该有一个状态图。状态图指定对象的行为以及根据不同的当前状态行为之间的差别,同时它还能说明事件是如何改变一个类的对象的状态。状态是对象操作的前一次活动的结果。通常情况下,状态由类中的指定属性来说明。改变对象状态的事情称为“事件”。

状态转换图的节点是对象的状态,其有向边是标有事件或事件相应行为的状态转换。由当前状态和接受到的事件决定下一个状态。状态转换图的例子如图 2-16 所示。该图为 POS 机的状态图,该图有关机、待机和在工作三种状态,off 和初始化是相应的命令事件,当 POS 机超过其处理容量时发出报警。

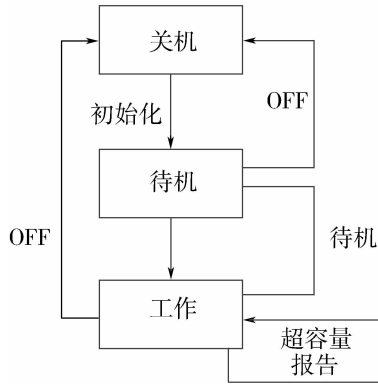


图 2-16 POS 机状态转换图

改变对象状态常用下面三种标准事件：进入、退出、做。其中进入事件用来指定进入一个状态的动作，如给属性赋值或发送一条消息，退出事件用来指定退出一个状态的动作，做事件则用来指定在该状态下的动作，例如发一条消息、等待或计算。

4. 序列图

序列图描述消息是如何在对象间发送和接收的。序列图有两个坐标轴，纵坐标显示时间，横坐标显示对象。序列图也显示特殊情况下的对象交互，在系统执行期间的某一时间点发生在对象间的特殊交互（如当用到一个特殊功能时）。在序列图的横坐标上是与序列有关的对象，同时有一条纵向的虚线表示对象在序列中的执行情况（即发送和接收消息对象的活动），这条虚线称为对象的生命线。对象间的通信用对象的生命线之间的水平消息线来表示。消息线的箭头类型说明消息的类型。图 2-17 是文档打印系统的部分序列图。

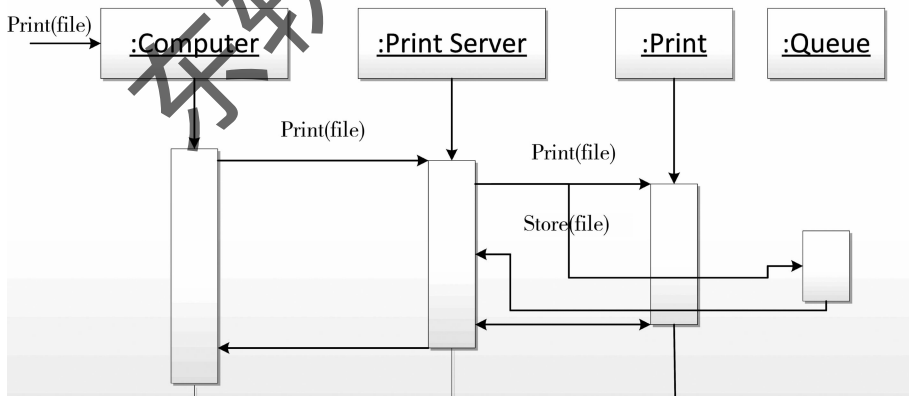


图 2-17 序列图

2.3.7 ATM 系统案例分析

自动取款机系统如图 2-18 所示。

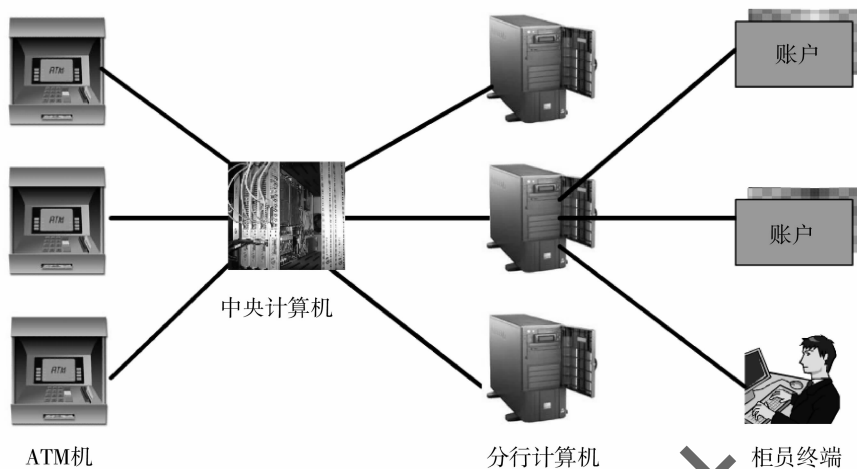


图 2-18 ATM 系统

1. ATM 需求陈述

(1) 银行拟开发一个自动取款系统,它是由自动取款机、中央计算机、分行计算机及柜员终端组成的网络系统。ATM 和中央计算机由总行投资购买。总行拥有多台 ATM,分别设在全市各主要街道上。分行负责提供分行计算机和柜员终端。柜员终端设在分行营业厅及分行下属的各个储蓄所内。该系统的软件开发成本由各个分行分摊。

(2) 银行柜员使用柜员终端处理储户提交的储蓄事务。储户可以用现金或支票向自己拥有的某个账户内存款或开新账户。储户也可以从自己的账户中取款。通常,一个储户可能拥有多个账户。柜员负责把储户提交的存款或取款事务输入柜员终端,接收储户交来的现金或支票,或付给储户现金。柜员终端与相应的分行计算机通信,分行计算机具体处理针对某个账户的事务并且维护账户。

(3) 拥有银行账户的储户有权申请领取银行卡。使用银行卡可以通过 ATM 访问自己的账户,用银行卡可在 ATM 上提取现金,或查询有关自己账户的信息(如账户上的余额)、办理转账、存款等事务。

(4) 所谓银行卡就是一张特制的磁卡,上面有分行代码和卡号。分行代码唯一标识总行下属的一个分行,卡号确定了这张卡可以访问哪些账户。每张银行卡仅属于一个储户所有,同一张卡可能有多个副本。

(5) 当用户把银行卡插入 ATM 之后,ATM 就与用户交互,以获取有关这次事务的信息,并与中央计算机交换关于事务的信息。首先,ATM 要求用户输入密码,接下来 ATM 把从这张卡上读到的信息以及用户输入的密码传给中央计算机,请求中央计算机核对这些信息并处理这次事务。中央计算机根据卡上的分行代码确定这次事务与分行的对应关系,并且委托相应的分行计算机验证用户密码。如果用户输入的密码是正确的,ATM 就要求用户选择事务类型(取款、查询等)。当用户选择取款时 ATM 请求用户输入取款额。最后,ATM 从现金出口吐出现金,并且选择是否打印出账单。

2. 确定类和对象

(1) 找出候选的类和对象

① 正式分析法: 包括可感知的物理实体、人或组织的角色、应该记忆的事件、两个或多个对象的相互作用、需要说明的概念等。

② 非正式分析法: 以需求陈述的名词为依据, 找出候选的类。

(2) 筛选出正确的类和对象

① 冗余: 两个类表达了同样的信息, 则保留在此问题域中最富于描述力的名称。(如储户、用户这两个可去除用户)

② 无关: 仅需要把与本问题密切相关的类和对象放进目标系统中。

③ 笼统: 把笼统或模糊的类去掉。

④ 属性: 有些名词实际上描述的是其他对象的属性, 应该去掉这些名词。

⑤ 操作: 有可能使用一些既可作为名词, 又可作为动词的词, 应该慎重考虑它们在本问题中的含义。

⑥ 实现: 在分析阶段不应该过早地考虑怎样实现目标系统。

经过初步筛选, 剩下的类和对象有: ATM、中央计算机、分行计算机、柜员终端、总行、分行、柜员、储户、账户、事务、银行卡。

3. 确定关联

(1) 直接提取动词短语得出的关联

① ATM、中央计算机、分行计算机及柜员终端组成网络

② 总行拥有多台 ATM

③ ATM 设在主要街道上

④ 分行提供计算机和柜员终端

⑤ 柜员终端设在分行营业厅及储蓄所内

⑥ 分行分摊软件开发成本

⑦ 储户拥有账户

⑧ 分行计算机处理针对账户的事务

⑨ 分行计算机维护账户

⑩ 柜员终端与分行计算机通信

⑪ 柜员输入针对账户的事务

⑫ ATM 与中央计算机交换关于事务的信息

⑬ 中央计算机确定事务与分行的对应关系

⑭ ATM 读银行卡

⑮ ATM 与用户交互

⑯ ATM 吐出现金

⑰ ATM 打印账单

⑱ 系统处理并发的访问

(2) 找出需求陈述中隐含的关联

① 总行由各个分行组成

- ②分行保管账户
 - ③总行拥有中央计算机
 - ④系统维护事务日志
 - ⑤系统提供必要的安全性
 - ⑥储户拥有银行卡
- (3)根据问题域得出的关联
- ①银行卡访问用户
 - ②分行雇用柜员
- (4)筛选关联

筛选时根据下述标准删除候选的关联：

- ①删除已删去的类之间的关联。
 - ②与问题无关的或应在实现阶段考虑的关联删去。
 - ③删去瞬时效应。
 - ④删去三元关联。将 3 个或 3 个对象以上的关联分解为二元关联。
 - ⑤删去派生关联。去掉那些可以用其他关联定义的冗余关联。
 - ⑥进一步完善余下的关联。
- (5)继续完成对象模型

4. 确定主题

确定主题的方法为：

- (1)将每一种结构(包括整体一部分结构和一般化—特殊化结构)中最上层的类提升成为主题。
- (2)将各不属于任何结构的类提升主题。
- (3)检查在相同或类似的问题领域中以前做面向对象分析的结果,看是否有可直接复用的主题。

5. 确定属性

确定一个属性有 3 个基本准则：

- (1)对相应对象或分类结构的每一个实例是否都适用？
- (2)在现实世界中它与这种事物的关系是否最密切？
- (3)确定的属性应当是一种相对的原于概念,不依赖于并列的其他属性就可以理解。

6. 对象模型的精化

利用继承关系,按照自顶向下或者自低向上的方式对系统中众多的对象类进行重组。

- (1)以自底向上的方式抽象出具有若干对象类共同性质的父亲(泛化)。
- (2)以自顶向下的方式把当前类细化成更具体的子类(分解)。
- (3)合并。

ATM 的对象模型图如图 2-19 所示。

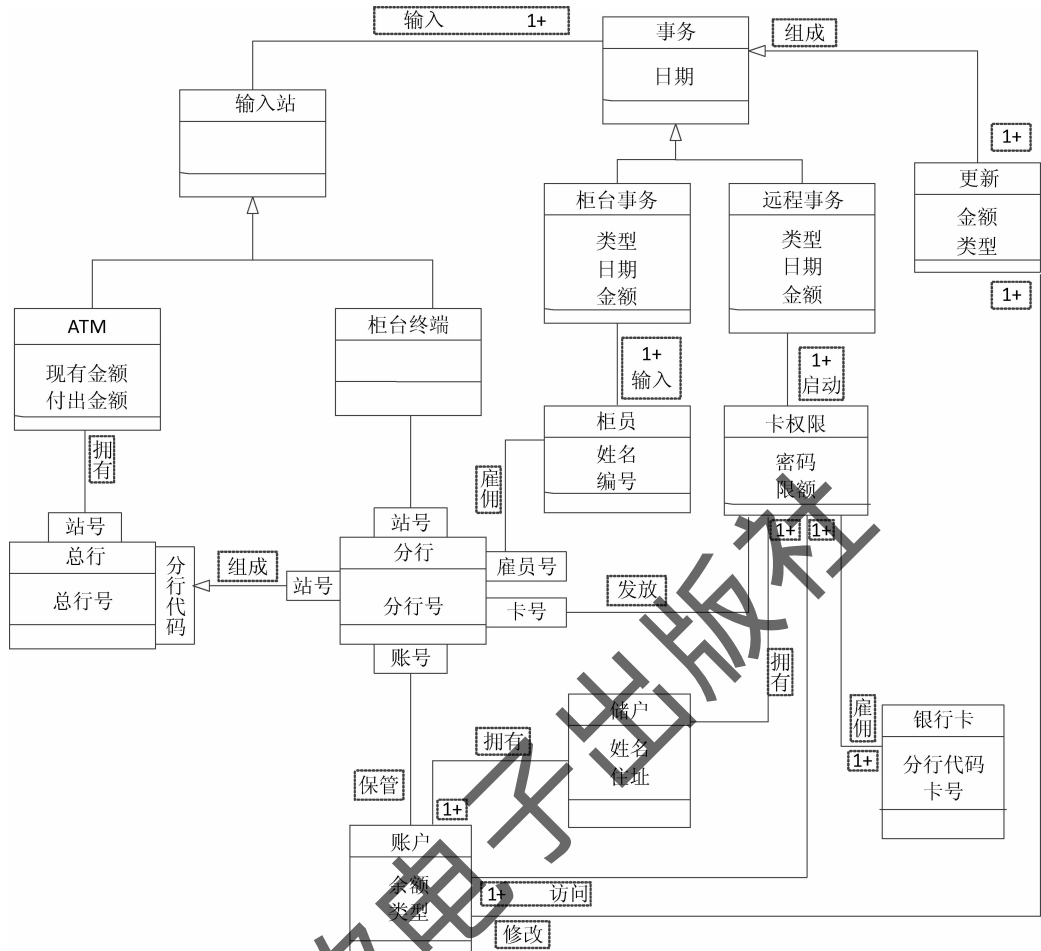


图 2-19 ATM 对象模型图

任务4 需求规格说明与评审

【任务分析】

通过完成需求规格说明与评审这个任务,我们可以了解如何书写系统需求规格说明书及如何对系统进行需求分析的评审等。

【知识点分析】

需求规格说明与评审这个任务所涉及知识点包括需求规格说明的主要内容、需求分析的评审标准等。

2.4.1 需求规格说明的主要内容

软件需求规格说明是分析任务的最终产物,它是软件开发过程中的一份关键性的技术

文档。它的作用是：

- (1) 作为软件人员和用户之间事实上的技术合同说明。
- (2) 作为软件人员下一步进行设计和编码的基础。
- (3) 作为测试和验收的依据。

软件需求规格说明主要内容包括：引言、任务概述、需求规定、运行环境规定等。

2.4.2 需求分析的评审

在将需求分析规格说明书提交给设计阶段之前，必须进行需求评审。如果在评审过程中发现说明书存在错误或缺陷，应及时进行更改或弥补，重新进行相应部分的初步需求分析、需求建模、修改需求规格说明书，并再行评审。

衡量需求规格说明书好坏的标准按重要性次序排列为正确性、无歧义性、完全性、可验证性、一致性、可理解性、可修改性、可追踪性。下面依次介绍这些评审标准的主要内涵。

(1) 正确性

需求规格说明书中的功能、行为、性能描述必须与用户对目标软件产品的期望相吻合。

(2) 无歧义性

对于用户、分析人员、设计人员和测试人员而言，需求规格说明书中的任何语法单位只能有唯一的语义解释。确保无歧义性的一种有效措施是在需求规格说明书中使用标准化术语，并对术语的语义进行明确的、统一的解释。

(3) 完全性

需求规格说明书不能遗漏任何用户需求。具体地说，目标软件产品的所有功能、行为、性能约束以及它在所有可能情况下的预期行为，均应该完整地包含在需求规格说明书中。

(4) 可验证性

对于规格说明书中的任意需求，均应存在技术和经济上可行的手段进行验证和确认。

(5) 一致性

需求规格说明书的各部分之间不能相互矛盾。这些矛盾可以表现为术语使用方面的冲突，功能和行为特征方面的冲突，以及时序方面的前后不一致。

(6) 可理解性

追求上述目标不应妨碍需求规格说明书对于用户、设计人员和测试人员的易理解性，不宜在说明书中使用太多的专业化词汇。

(7) 可修改性

需求规格说明书的格式和组织方式应保证能够比较容易接纳后续的增加、删除和修改，以便修改后的说明书能够较好地保持其他各项性能。

(8) 可追踪性

需求规格说明书必须将分析后获得的每项需求与用户的原始需求项联系起来，并为后续开发和其他文档引用这些需求项提供便利。

为保证软件需求定义的质量，评审应由专门指定人员负责，并按规程严格进行。评审结束应有评审负责人的结论性意见及签字。除分析员之外，用户/需求者、开发部门的管理者，以及软件设计、实现、测试人员都应当参加评审工作。一般地，评审的结果都包含了一些修改意见，待修改完后再经评审通过，才可能进入设计阶段。

2.4.3 学生心理动态跟踪评价系统需求规格说明书

软件需求规格说明书

1 引言

本文是学生心理动态跟踪评价系统 SPDTES (Students' Psychological Dynamic Tracking Evaluation System) 的软件需求规格说明书。

大学生心理健康问题已经成为当今社会的重要问题之一。大学生的心理问题复杂、多变,其诱发原因也是多种多样。对高校大学生心理进行动态跟踪,及时把握学生的心理动向,扩充学生咨询心理问题的渠道,才能有效解决大学生的心理健康问题。

学生心理动态跟踪评价系统 SPDTES (Students' Psychological Dynamic Tracking Evaluation System) 是根据某高校的实际需求,依据某高校心理辅导站的实际管理过程开发的,采用了目前最流行的 Internet/Intranet 基于浏览器/服务器 (Browser/Server) 结构体系的主流技术 ASP.NET 技术作为运行平台。

1.1 编写目的

SPDTES 以某高校现行使用的《学生心理档案系统》检测新生心理的数据为研究基础,以全员参与心理健康教育为目的,提供及时收集学生心理动态信息,开发密切关注学生动态发展的师生互动平台。开发学生心理动态跟踪评价系统,高校心理健康管理部门可以及时把握学生心理动态发展,制定应对教育措施;学生可及时获取相关教育信息,反馈教育效果。该系统以校园网为平台,不仅扩充学生咨询心理的渠道,而且更能通过网络虚拟消除学生心理障碍,便于学生咨询,提升高校学生心理管理的水平和层次。

1.2 背景

本项目的名称:学生心理动态跟踪评价系统 SPDTES (Students' Psychological Dynamic Tracking Evaluation System)。

本项目的用户是高校学生处、学院心理辅导站、二级学院党书记、心理辅导员、班主任、班级心理委员。系统能及时了解学生心理变化,对学生心理不健康数据进行及时、准确、高效的采集,能及时获取对心理不健康学生进行干预的最佳时间,实现了系统化和网络化,提高工作时间和效率,节约了管理成本。

1.3 定义

学生心理动态跟踪评价系统 (Students' psychological dynamic tracking evaluation system),简称 SPDTES,是学院心理辅导站对班级心理咨询、管理的网络软件。

1.4 参考资料

- [1] 李绍原. 数据库原理及应用. 北京:科学出版社,1999
- [2] 郑人杰,殷人昆,陶永雷. 实用软件工程. 北京:清华大学出版社,2000
- [3] Jesse Liberty. C#程序设计. 刘基诚,译. 北京:中国电力出版社,2002
- [4] 宋翔,聂雪. 图形制作与处理. 西安:西安电子科技大学出版社,2003
- [5] 郭庚麒. 软件工程基础教程. 北京:科学出版社,2004
- [6] 李明刚,肖建. ASP.NET Web 站点高级编程范例. 北京:清华大学出版社,2004
- [7] 黎卫东. ASP.NET 网络开发入门与实践. 北京:人民邮电出版社,2006
- [8] 吴晨,张亮,张静. ASP.NET+SQL Server 数据库开发与实例. 北京:清华大学出

出版社,2006

[9] 精英科技. SQL Server 2005 高级开发指南. 北京:中国电力出版社,2008

[10] 邵良彬,刘好增,马海军. ASP.NET(C#)实践教程. 北京:清华大学出版社,2012

2 任务概述

2.1 目标

2.1.1 该软件的开发意图

- (1) 对学生心理委员工作用网络化管理代替人工管理;
- (2) 使学院心理辅导老师对学生心理工作管理更加规范、便捷,信息采集更加准确、及时;
- (3) 减轻心理辅导站老师的工作负担;
- (4) 适应网络发展的需要、社会和谐发展的需要,使学生心理健康信息更加公开、管理过程更加透明。

2.1.2 该软件的应用目标

通过本软件的开发,减轻心理辅导站的工作强度、实现网络化、无纸化管理,方便班级心理委员实时汇报情况、能有效监控学生心理工作情况。

2.1.3 该软件的作用及范围

本软件针对于我校学生心理管理工作,也普遍适用于高校学生心理管理,它比较独立和完善的高校大学生心理管理软件,对学生心理管理可不受时空限制,管理过程具有实时记录,便于查询。

2.1.4 该软件开发的背景

随着学校规模的日益扩大,学生人数的增多,高校社会化日益严重,学生心理负担越来越多,出现心理不健康状况频率有增多态势。心理辅导站工作人员少,工作强度大,同时缺少一种对学生心理日常工作管理软件,为了更好地服务师生、更加及时、准确把握学生心理健康动态,应高校的要求开发学生心理动态跟踪评价系统。

2.2 用户的特点

本软件的使用对象是学生处、心理辅导站、班主任及班级心理委员,都具备一般计算机的基本操作知识,能进行简单网络操作,因此所有用户均能正确使用本软件。

2.3 假定与约束

一般约束:

项目的开发经费、时间等见软件开发合同书;

在管理方面,硬件的限制,并行操作安全和保密方面无约束。

假设:

假设开发过程中管理不完善,设计时系统功能未能得到全面考虑,对于不可预知的情况(如部分学生托管、专业归属发生改动等),本项目的开发都将受到很大的影响。

3 需求规定

3.1 对功能的规定

学生心理动态跟踪评价系统具有输入、输出、增加、删除、修改、查询、传送等多项功能。

3.2 对性能的规定

(1) 精度

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

(2) 时间特性要求

在软件方面,响应时间,更新处理时间都比较快且迅速,完全满足用户要求。

(3) 灵活性

当用户需求,如操作方式、运行环境、结果精度、数据结构与其他软件接口等发生变化时,设计的软件要做适当调整,灵活性非常大。

3.3 故障处理要求

(1) 内部故障处理

在开发阶段可以立即修改数据库里表的相应内容或者调整代码。

(2) 外部故障处理

可增加一项程序功能去修改导入表的内容。

3.4 其他专门要求

(1) 保密性

本软件作为学生心理管理工作辅助工具,可限定一个程序中某些区域的规约,给不同的用户角色分配不同的功能。

(2) 可使用性

该系统采用了 B/S 结构,可使用性非常强,开发、维护等几乎所有工作都集中在服务器端。同时当课题组对网络应用进行升级时,只需更新服务器端的软件,而不必更换客户端软件,减轻了系统维护与升级的成本与工作量,方便普通用户使用。

(3) 可维护性

本软件开发平台采用 Microsoft Visual Studio 2008,后台代码支撑语言选择 C# 语言,数据库使用 SQL SERVER 2005,将脚本语言内嵌在 HTML 文档中,将数据连接与调用方法存放在类文件,数据读取简单、清晰、代码可读性好、容错性强、具有较高的可维护性。数据采用 SQL SERVER 2005,数据独立于系统,与源代码分开,数据安全性强。

4 运行环境规定

4.1 设备

本系统是采用使用市场占有率很大的基于 B/S 结构的 ASP.NET 动态网络技术设计,对一般客户端的计算机配置要求仅为一台具有基本网络配置、Windows 操作系统和网络浏览器的个人计算机,而且升级容易,仅仅需要对服务器端的系统进行升级就可以完成整个系统的升级。具体配置要求如下:

客户机:×××××

内 存:512MB 以上

硬 盘:10GB 以上

显示器:彩显

鼠 标:Microsoft 兼容鼠标

服务器(建议配置):×××××

内 存:4GB 以上

硬 盘:40GB(10G 闲置空间)以上

4.2 软件环境

学生、教师客户端软件环境:

中文 Windows XP/ Windows 2000/ Windows NT/ Windows 7/ Windows 8

配置 TCP/IP 协议

服务器软件环境:

.NET Framework 和 IIS 运行

中文 Windows NT 4.0+ IIS 4.0 或 Windows 2000+IIS 5.0 及以上

数据库 SQL server 2005

4.3 接口

(1) 用户接口

本产品的用户一般需要通过终端进行操作,进入主界面后点击相应的窗口,分别进入相对应的界面(如:用户登录界面)。用户对程序的维护,最好要有备份。

(2) 软件接口

Windows XP/ Windows 2000/ Windows NT/ Windows 7/ Windows 8 操作系统。

4.4 控制

本系统的开发平台是 Microsoft Visual Studio 2008,选用 C# 语言为后台编程语言。

习题二

1. 可行性研究主要研究哪些问题? 请试说明。
2. 需求分析的任务和步骤是什么?
3. 什么是结构化分析方法? 结构化分析方法的结果是什么?
4. 数据流图的作用是什么? 它有哪些基本成分?
5. 数据词典的作用是什么? 它有哪些基本条目?
6. 常用的加工逻辑的描述方法有哪些?
7. 软件需求规格说明的主要内容是什么? 它的作用是什么?
8. 什么是面向对象的需求分析方法?