

【学习重点】

- (1) 理解结构化分析的基本概念。
- (2) 理解结构化分析的基本步骤。
- (3) 理解结构化分析的主要方法。

5.1 结构化分析概述

结构化分析方法是一种传统的系统化的软件建模技术,其过程包括创建描述软件中所涉及的信息和行为,依据功能和行为对软件进行功能划分,建立软件的数据模型和数据处理模型,并描述模型所涉及的软件要素。

在需求工程中,分析师需要创建软件的各种模型,以便更好地理解数据流与控制流、处理功能与操作行为及相关信息,并综合考虑软件的功能需求、非功能需求和数据要求的分析结果,导出软件的逻辑模型。例如,使用精化的软件分解方法,建造软件处理的数据、功能与行为模型,为软件设计者提供可被翻译成数据、体系结构、界面和行为设计的模型,以及通过需求规格说明文档为开发者和客户提供软件质量评估的依据等。

结构化分析(Structured Analysis,SA)方法是 20 世纪 70 年代由 E. Yourdon 等倡导的一种适用于大型数据处理系统的、面向数据处理的软件需求分析方法。结构化分析方法一般采用以下指导性活动。

(1) 充分理解问题。人们通常急于求成,甚至在问题未被很好地理解之前,就产生了一个解决错误问题的软件。

(2) 开发快速原型。原型的目的是使用户能够了解将如何进行人机交互(推荐使用原型技术)的过程(包括输入、输出信息等),理解软件功能需求与业务需求的差别。

(3) 描述软件需求。记录每个需求的起源和原因,这样能有效地保证需求的可追踪性和可回溯性。

(4) 建立软件高层模型。使用多个分析模型,建立软件所涉及的数据、功能和行为的模型,为软件工程师提供不同的视图,有效避免可能会忽略某些软件要素,并增加识别出存在不一致性的地方。

(5) 确定软件需求优先级。赋予软件需求优先级,优先开发重要的功能,提高软件开发生产效率。

(6) 验证软件需求。软件需求常用自然语言描述,存在含糊的可能,这可以通过复审发现问题,删除需求描述含糊或不一致的地方。



视频讲解

5.2 结构化分析模型

结构化分析方法是一种半形式化的分析与建模技术,其过程包括对软件相关的各种信息进行分析,抽取其本质特征,创建描述数据、功能和行为的模型。

软件模型不是软件的替代表示,而是抛弃了具体细节的一个软件功能抽象。在理想情况下,软件需求描述需要给出软件中各种实体的全部信息,而软件功能抽象就是挑选出软件中最突出的特征进行简要描述。所以,软件模型可以从不同的角度表达软件的特征。

结构化分析模型的主要目标是描述客户的需要,为软件设计建立基础,定义在软件开发完成后可以确认的一组需求。结构化分析模型有面向数据的软件模型和面向行为的软件模型两大类。

(1) 面向数据的软件模型主要刻画软件中所涉及的数据及其关系,用来确定软件的数据结构和存储结构模型。实体关系模型就是一个面向数据的软件模型,其关心的是寻找软件中的数据及其之间的关系,却不关心软件的功能。

(2) 面向行为的软件模型又包括两类。一类是面向数据处理流的软件模型,也称为数据流模型,用来描述软件中处理数据的过程。数据流模型集中在数据的流动和数据转换功能方面,而不关心数据结构细节。另一类是状态转换模型,用来描述软件是如何对内外部事件做出响应的。这两种模型可以单独使用,也可以一起使用,要视软件的具体情况而定。结构化分析模型分别用数据流图、数据字典、状态转换图、实体-关系图等描述,其组成形式如图 5.1 所示。

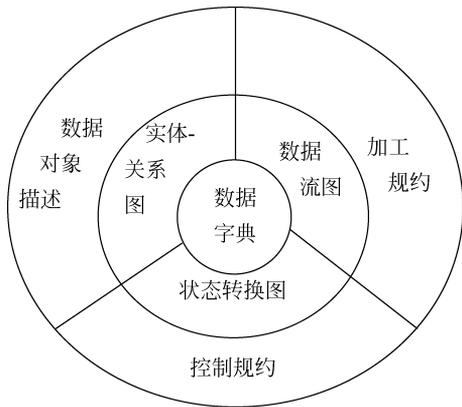


图 5.1 结构化分析模型组成形式

结构化分析模型组成的核心是数据字典(Data Dictionary, DD),包含了软件使用或开发的所有数据描述的中心库。结构化分析模型有 3 类视图。

(1) 数据流图(Data Flow Diagram,DFD)。其服务于两个目的:一是指明数据在软件中移动时如何被变换,二是描述软件对数据进行变换的功能和子功能。数据流图提供了附加信息,它们可以用于信息域的分析,并作为软件功能建模的基础。

(2) 实体-关系图(Entity-Relationship Diagram,E-RD)。实体-关系图用于描述数据对象的属性、数据对象之间的关系等。



视频讲解

(3) 状态转换图(State Transition Diagram,STD)。状态转换图用于指明作为外部事件的结果,软件将如何动作。状态转换图可表示软件的各种行为模式,以及在状态间转换的方式,是行为建模的基础。

结构化分析模型组成的外层是描述。在实体-关系图中出现的每个数据实体及其属性可以使用数据对象来描述。在数据流图中出现的每个加工/处理的功能描述包含在加工规约中。软件控制方面的附加信息包含在控制规约中。

5.3 数据流分析方法

面向数据流是一种结构化需求分析方法,也是一种软件建模活动,简称数据流分析方法。该方法使用简单易读的符号,根据软件内部数据传递和变换关系,自顶向下逐层分解,描绘满足用户要求的软件行为模型。

5.3.1 数据流图

数据流分析方法采用数据流图来表示软件的行为模型。用数据流图描述软件的处理过程是一种很直观的方式。在需求分析中用数据流图来建立现存目标软件的数据处理模型,描述数据被手工或计算机处理或转换的过程。当数据流图用于软件设计时,这些处理或转换在最终生成的程序中将是若干功能模块。

数据流图有 4 种基本符号,如图 5.2 所示。

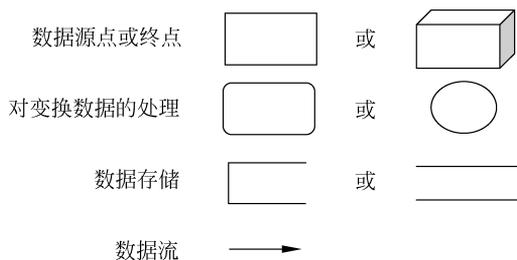


图 5.2 数据流图的基本符号

- 矩形或立方体表示数据源点或终点。
- 圆角矩形或圆代表对变换数据的处理。
- 开口矩形或两条平行线代表数据存储。
- 箭头表示数据流。

这里,处理可以是一个过程或一系列程序组成的过程,甚至可以由人工完成的过程。一个数据存储可以表示一个文件或文件的一部分、数据库的元素或记录的一部分等;数据可以存储在磁盘、主存等任何介质上。数据存储是处于静止状态的数据,而数据流是处于活动中的数据。数据源点和终点有时会是同一个实体或对象,若只用一个符号代表数据的源点和终点,则至少有两个箭头与这个符号相连。

数据流图的基本要点是描绘软件“做什么”,而不考虑“怎样做”。通常数据流图会忽略出错、打开和关闭文件之类的内部处理等软件细节的处理方面。

数据流图通常作为与开发人员交流的工具,是软件分析和设计的工具,也有助于进行更

详细的设计。

5.3.2 数据字典

64

数据字典是分析模型中出现的所有名字的集合,并包括有关命名实体的描述。如果名字是一个复合对象,它还应有对其组成部分的描述。数据字典在软件模型开发中非常有用,它可以管理各种数据流图、实体-关系图和状态转换图中的各种信息。使用数据字典有以下两个作用。

(1) 它是所有名字信息管理的有效机制。在一个大型软件系统中,需要给模型中的许多实体和关系命名,而这些名字在软件中必须保持一致且不能出现冲突。数据字典可以检查名字的唯一性。

(2) 作为连接软件分析、设计、实现等阶段的纽带,以阶段进化的方式进行信息存储。随着软件的改进,字典中的信息也会发生相应的变化,新的信息会随时加入进来。

在描述数据流图涉及的信息时,数据字典应该由数据流、数据流分量、数据存储和处理4类元素的定义组成。除数据定义之外,数据字典还应该包括关于数据的其他信息。

- (1) 一般信息,包括名字、别名、描述等。
- (2) 定义,包括数据类型、长度、结构等。
- (3) 使用特点,包括取值的范围、使用频率、使用条件、使用方式、条件值等。
- (4) 控制信息,包括用户、使用特点、改变数、使用权等。
- (5) 分组信息,包括文档结构、从属结构、物理位置等。

在数据字典中,应对组成的数据元素定义进行自顶向下的分解。分解的原则是当包含的元素不需要进一步定义,且每个和工程有关的人都清楚时为止。

由数据元素组成数据的方式有3种基本类型,可以使用这3种类型的任意组合定义数据字典中的任何条目。

- (1) 顺序:顺序连接两个或多个分量元素,一般用加号表示顺序连接关系。
- (2) 选择:从两个或多个可选的分量元素中选取一个,选择运算符用方括号表示,对于多个可供选择的元素,用“|”符号分隔。例如, $[A-1|A-2|A-3]$ 表示三个可选数据元素。
- (3) 重复:描述的分量元素重复零次或多次,重复运算符用花括号表示,并与重复的上下限同时使用。如果上下限相同,则表示重复次数固定;如果上下限分别为0和1,则表示分量可有可无。

数据字典描述通常采用卡片形式,一张卡片上应包含名字、别名、描述、定义、位置等信息。

5.3.3 数据流分析方法的步骤

数据流分析方法的步骤分为数据流图要素分析、构建数据流图和建立数据字典3个步骤。

- (1) 数据流图要素分析。根据问题确定数据的源点、终点、数据流、数据存储和处理等。
- (2) 构建数据流图。根据数据流图要素绘制数据流图。
- (3) 建立数据字典。对数据流分析中所涉及的所有要素进行详细的规格描述。

前两个步骤是一个逐步求精的过程,一开始从整体的角度分析数据流处理过程,然后针对每个数据流处理过程进行多次分解,精化数据流图,直到不能分解为止。

【例 5.1】 订货软件的数据流图构建。

设某工厂的采购部每天需要一张订货报表。订货零件的信息包括零件编号、零件名称、数量、价格、供应者等。零件的入库、出库事务通过计算机终端输入给订货系统。当某零件的库存数少于给定的库存量临界值时,就应该再次订货。

(1) 数据流分析。

根据订货软件需求的描述,可以抽取以下要素。

- ① 数据源点有仓管员(负责入库或出库事务,并上报信息至订货系统);
- ② 数据终点有采购员(接收每天的订货报表);
- ③ 数据流有事务和订货报表;
- ④ 数据存储有订货信息和库存清单;
- ⑤ 处理有处理事务和产生报表。

(2) 构建基本软件模型。

软件的特征就是将输入信息转换成输出信息,任何软件的基本模型都由若干数据源点或数据终点和一个对数据加工处理组成。图 5.3 所示的是订货软件基本模型的数据流图。



图 5.3 订货软件基本模型的数据流图

(3) 第 1 次求精。

很明显,订货软件基本模型的数据流图非常抽象,因此需要把订货软件中的处理进一步细化,确定出软件的主要功能。订货软件细化后的数据流图如图 5.4 所示,可分为处理事务和产生报表两个主要功能,同时增加了库存清单和订货信息两个数据存储,并对应出现了事务、库存信息、订货信息、订货报表 4 个数据流。

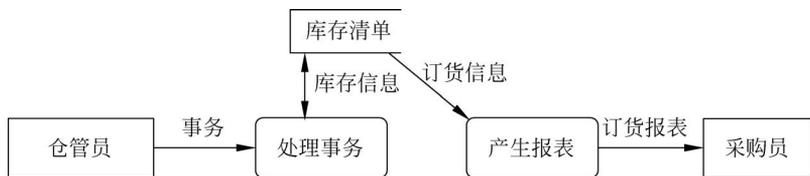


图 5.4 订货软件数据流图：第 1 次求精

(4) 第 2 次求精。

对订货软件的数据流图进一步细化。订货软件第 2 次求精的数据流图如图 5.5 所示。当发生一个事务时,接收它并按照事务的内容修改库存清单,然后根据库存临界值确定是否订货。考虑到入库、出库是不同的事务处理,于是把处理事务进一步分为处理入库和处理出库。

产生报表实际上包含了订货和生成订货报表两个过程,因此进一步将产生报表进一步分成处理订货和生成报表两个处理。

注意,数据流和数据存储的命名必须有具体含义。处理的命名应反映整个处理的功能,通常是由一个动词加上一个具体的宾语组成。

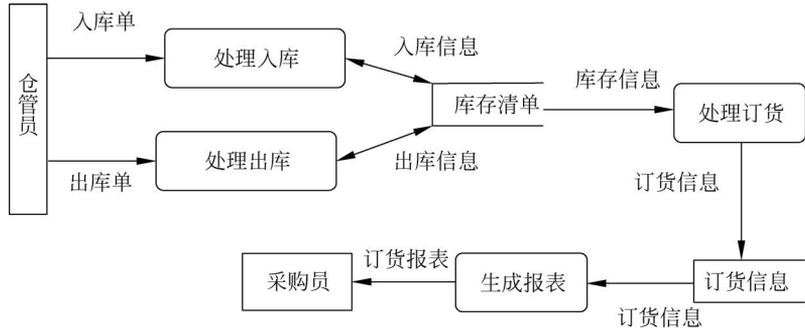


图 5.5 订货软件数据流图：第 2 次求精

订货系统中部分卡片形式的数据定义如下。

名字：订货报表

别名：订货信息

描述：每天一次需要订货的零件表

定义：订货报表=1{零件信息}N

位置：输出到打印机

其中：

零件信息=零件编号+零件名称+数量+价格+1{供应者}3

分量定义为：

零件编号=8 位字符

零件名称=20 位字符

数量=[1 | 2 | 3 | 4 | 5]

价格={零件单价}

数据字典应能产生交叉参照表,有错误检测、一致性校验等功能。

5.4 数据建模分析方法

数据建模分析方法通常开始于数据建模,是定义在软件内部处理的所有数据实体、数据实体之间的关系以及其他与这些关系相关的信息。

软件建模的一个重要方面是定义软件处理的数据逻辑结构。最广泛采用的数据建模技术是实体-关系模型,它描述数据实体、实体之间关系及实体属性。

1. 数据建模过程

实体是软件能够理解的复合信息的表示,是具有若干不同的特征或属性的事物,是实现世界中存在的且可相互区分的事物。实体可以是外部事物,也可以是发生的事件、组织单位、地点或结构等。

建立实体-关系模型分为抽取实体和建立实体-关系图两个步骤。抽取实体的方法是采用领域分析的方法。首先,要进行领域分析,获取领域所涉及的各种名词和术语,这些都有可能成为实体。然后,针对这些候选的实体,剔除与系统无关的或不密切的实体,或者属于其他实体属性的实体等。

实体之间往往是有联系的。例如,在图书馆管理系统中,“借书者”实体与“图书”实体之



视频讲解

间就存在“借”或“还”的关系。实体可以以多种不同的方式相互连接。要确定这种关系,需要理解所创建的软件环境中借书者与图书的角色,可以用一组“实体/关系”对来定义有关的关系,例如借书的人、买书的人等。如果在图书馆管理系统中,可以理解为借书的人,而在书店中可理解为买书的人。

建立实体-关系图就是根据实体及它们之间的关系构建实体及其关系的表达。首先,分析这些实体,确定它们之间的关系和这些关系的重数。然后,绘制完整的实体-关系图。最后,编写实体-关系图中所涉及元素的规格说明。

2. 实体-关系图

实体-关系模型可用实体-关系图来表示。实体-关系图由实体、实体之间的联系和实体属性组成。

(1) 实体是客观世界中存在的且可相互区分的事物。实体可以是人或者物,也可以是具体事物,还可以是抽象概念。例如,职工、学生、课程等都是实体。

(2) 实体之间的关系描述了实体与实体之间的联系。在实体-关系图中,关系用一个菱形表示,实体与实体通过关系相连。关系还需要理解实体 X 出现的次数与实体 Y 的多少次出现相关,称为关系的基数。关系的基数分为 3 类:

- 一对一(1:1)关系,表示一个实体只能和一个实体关联。
- 一对多(1:N)关系,表示一个实体可以和很多实体关联。
- 多对多(M:N)关系,表示一个实体和另一个实体的多次出现关联。

关系的基数可以定义为能够参与一个关联的最大实体实例数。例如,教师与课程的“教”是一对多的关系,每位教师可能教多门课程,但每门课程只能由一位教师教。学生和课程的“学”是多对多关系,一名学生可以学多门课,而每门课可以有 multiple 学生来学。

(3) 属性是实体或关系所具有的性质。通常一个实体有若干属性。例如,“学生”实体有系、年级、学号、性别等属性。关系也可以有属性。例如,学生“学”课程所取得的成绩,既不是学生的属性也不是课程的属性。因为“成绩”既依赖于某名学生,又依赖于某门具体的课程,所以它是学生与课程之间“学”关系的属性。

【例 5.2】 图书馆管理系统的实体-关系图。

根据图书馆管理系统的问题描述,可得到以下实体:图书、借书者、管理员、借书目录、预约记录、书目。图书馆管理系统初步的实体-关系图如图 5.6 所示。

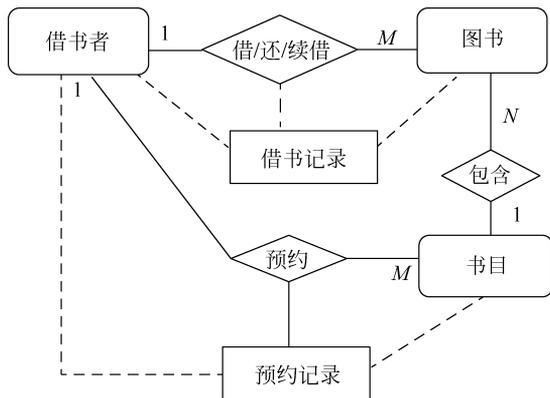


图 5.6 图书馆管理软件系统的部分实体-关系图

借书者可以借/还/续借一本图书；图书可包含在书目中；借书者可预约一个书目(抽象的书)。另外,借/还/续借的结果是产生删除/更新一条借书记录；预约产生一条预约记录。借书记录和预约记录是关联的结果,而虚线指明借书记录是借书者与图书的共同关联实体。同理,预约记录是借书者与书目的共同关联实体。

特别说明的是,图书和书目的概念不同。图书指的是图书馆管理系统中具体的一本书,可由每本书上的编号或条码号区分,而书目是该书的描述,它可以包含几本相同书名的图书。因此,预约只能发生在书目上,而不是图书上,因为如果预约某本具体的图书,那么即便同类书目的其他书归还,预约者还不能借,原因是预约的不是那本书,这显然不符合实际情况。

这些实体的主要属性如下。

借书者：借书者编号、姓名、性别、借书数、最大借书数、罚金金额、有限期。

图书：图书号、书目号。

书目：书目号、书名、作者、出版社、丛书名、收藏数、在馆数、预约数。

借书记录：图书号、借书者编号、借出日期、应还日期、续借次数。

预约记录：书目号、借书者编号、预约日期。

这里需特别注意借书记录和预约记录分别是借/还/续借关系的结果,而且这种关系是一对多的关系,因此,可以将这种一对多的关系通过借书者编号和图书号转换为一对一的关系,如图 5.7 所示。

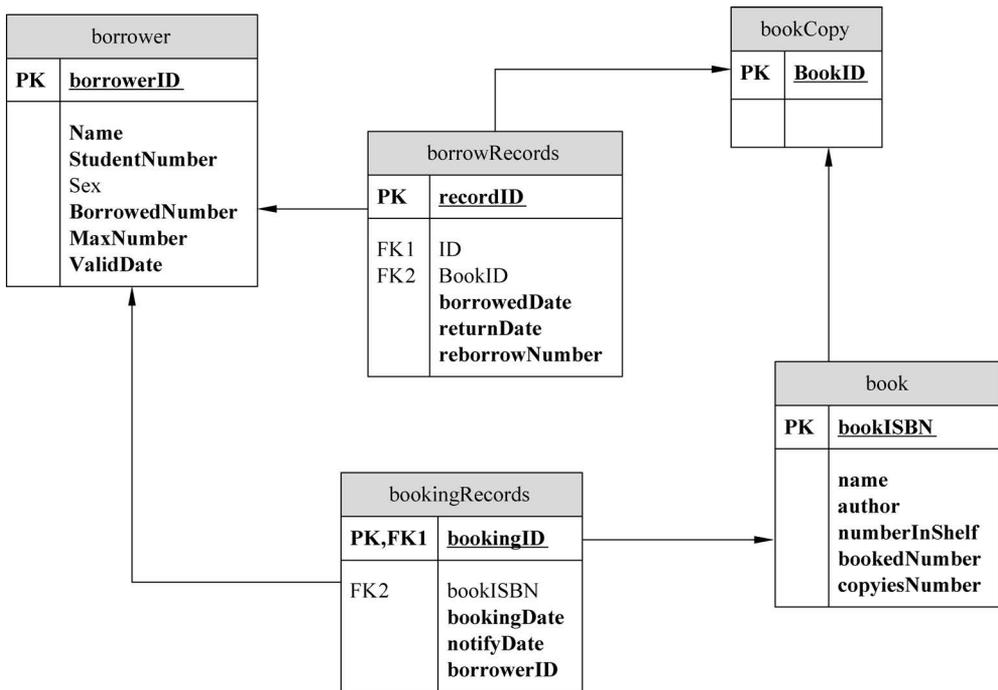


图 5.7 图书馆管理系统调整后的实体-关系图



5.5 状态分析方法

状态分析方法的主要任务是建立软件的状态模型。状态模型特别适合于具有复杂状态的软件建模,可以刻画软件处于不同状态时对事件的响应行为,尤其是对控制系统和网络通信协议的分析特别有效。

状态模型是一种描述系统对内部或者外部事件响应的行为模型。它描述软件的状态和事件,以及事件引发软件在状态间的转换。这种模型适用于描述实时软件系统,因为实时软件系统往往是由外界环境的激励而驱动的。

状态模型一般采用状态转换图(简称状态图)的标记方法。状态图描述了系统中某些复杂对象的状态变化,主要有状态、变迁和事件 3 种符号。

- (1) 状态是可观察的行为模式,用圆角矩形表示。
- (2) 变迁表示状态的转换,用箭头表示。
- (3) 事件是引发变迁的消息,用箭头上的标记表示。

状态图还可以用事件后的方括号表示先决条件,只有当这个条件为真时,才会发生状态变迁。当用状态自身的弧线箭头表示先决条件不为真时,状态不会变迁。

状态建模方法分为以下两个步骤。

(1) 软件的状态、行为与事件分析。软件的状态、行为与事件分析确定软件有哪些状态、行为和事件。可以通过软件运行场景来确定,也可以人为划分为一些逻辑状态。状态分析确定状态之间的转移和导致转移的事件。行为与事件分析需要确定事件导致什么样的状态转移,以及事件发生的条件等。

(2) 构建状态图。利用步骤(1)中分析的结果绘制状态图。

【例 5.3】 电梯系统状态分析。

在一幢有 m 层的大厦中安装一套由 n 部电梯组成的电梯系统,按照下列条件求解电梯在各楼层之间移动的逻辑关系:每部电梯有 m 个按钮,每个按钮代表一个楼层。当按下一个按钮时该按钮指示灯亮,同时电梯驶向相应的楼层,当到达相应楼层时指示灯熄灭。除了最底层和最高层之外,每一层楼都有两个按钮分别指示电梯上行和下行。按下按钮后指示灯就开始亮,当电梯到达此楼层时指示灯熄灭,并向所需要的方向移动。当电梯无升降运动时,关门并停在当前楼层。

通过上述的问题描述,经分析可知,电梯系统主要是控制电梯的运行,而电梯的运行比较复杂,具有很多状态。因此,重点分析电梯系统的各种状态和行为。电梯在运行过程中一般具有下列状态。

- 空闲:无请求时,电梯处于休息状态。
- 暂停:上下乘客时,电梯处于暂停状态。
- 上行:电梯处于向上运行状态。
- 下行:电梯处于向下运行状态。
- 处于第一层:初始启动,电梯会在第一层处于等待状态。
- 向第一层移动:当电梯长时间没有请求而处于空闲状态时,电梯会移动到第一层。

电梯系统具有下列事件。

- 向上：驱动电梯向上运行。
- 向下：驱动电梯向下运行。
- 停止：电梯停止运行。
- 无请求：没有乘客请求乘坐电梯。
- 长时间无请求：长时间没有乘客请求乘坐电梯。

绘制状态图时,一般要确定软件的初始状态和终止状态。初始状态是指系统加电或启动时要进入的第一个状态。电梯系统的初始状态是电梯处于第一层。终止状态是指结束时系统的状态,即软件系统最后一个转向到终止的状态。电梯系统没有终止状态,除非停止系统运行。一般,起始状态只有一个,而终止状态可以有多个,也可以没有。电梯系统的状态图如图 5.8 所示。

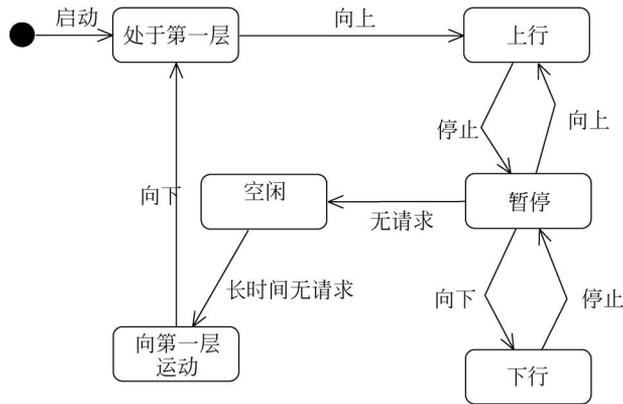


图 5.8 电梯系统的状态图

复杂的状态还可进一步分解为子状态。例如,“暂停”状态就是一个复合状态,其可以进一步分为“电梯门打开”状态、“系统计时”状态和“电梯门关闭”状态,如图 5.9 所示。

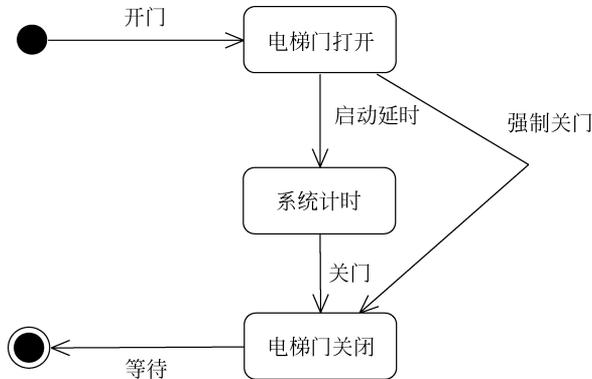


图 5.9 “暂停”复合状态的子状态图

5.6 结构化分析过程

结构化分析过程主要包括问题描述、绘制数据流图、确定需要计算机化部分、定义数据、定义处理逻辑、定义物理资源、确定输入/输出规格说明、确定数据规模与访问频率、确定硬



视频讲解

件需求和撰写软件需求规格说明文档等过程。

【例 5.4】 书店图书销售软件结构化分析。

(1) 问题描述。

书店从各出版社购买图书,并将其销售给学校、公司和个人客户。书店除订购流行的图书外,还根据需要订购其他图书。书店提供学校订购服务,并根据客户和订购量予以优惠。现在书店希望实现计算机化管理,需如何做?

仔细分析上述问题,确定需要具备哪些商务功能(入账、出账和库存),系统是采用批处理方式还是联机方式,硬件设备情况,实现计算机化管理的目标等。可以看出,其目的是销售图书、管理图书以及管理账目。

为了有效地分析软件的需求,采用逐步求精的结构化分析技术。结构化的分析,可以通过数据流图的描述确定逻辑数据流。

(2) 绘制数据流图。

图书销售例子的数据源为“顾客”;数据流为“订单”“发货清单”;数据存储为“图书数据”和“顾客数据”计算机文件,处理为“处理订单”操作。

画数据流图需要逐步求精,得到一套分层的数据流图。图 5.10 为第 1 次求精的数据流图。

当顾客有一个“图书需要”而书店没有时,图书细节将放入“待订单”(可以表示为计算机文件)。计算机每天扫描这些订单,如果向一个供应商发出的订单足够多,就批量订购;如果一个订单已等待了 5 天,则必须立即订购;如果订单有效,则处理订单并发货。图 5.11 为第 2 次求精的数据流图。第 2 次求精没有考虑的财务功能,即入账和出账等,可在第 3 次求精中加入。

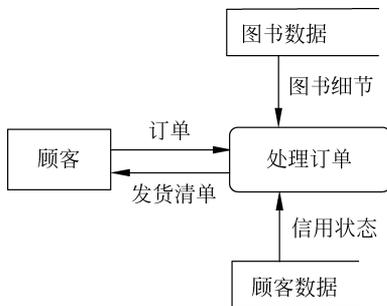


图 5.10 数据流图：第 1 次求精

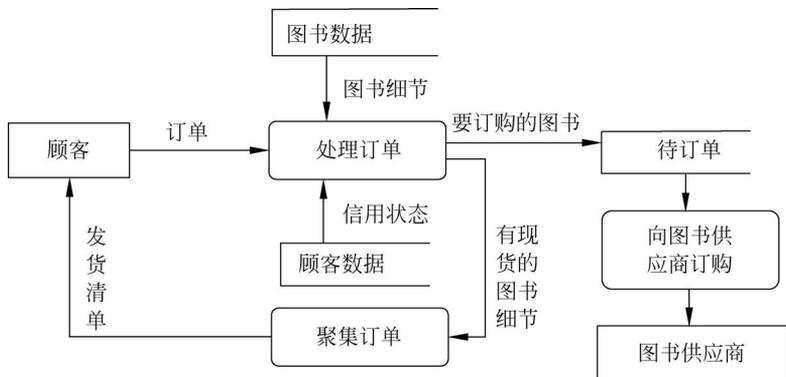


图 5.11 数据流图：第 2 次求精

图 5.12 仅给出了第 3 次求精的一部分。最终求精的数据流图可能更大,此时需要一个有层次结构的数据流图来描述。

(3) 确定需要计算机化部分。

系统的自动化方案选择,取决于客户的投资和目标。一般必须利用成本-效益分析方法

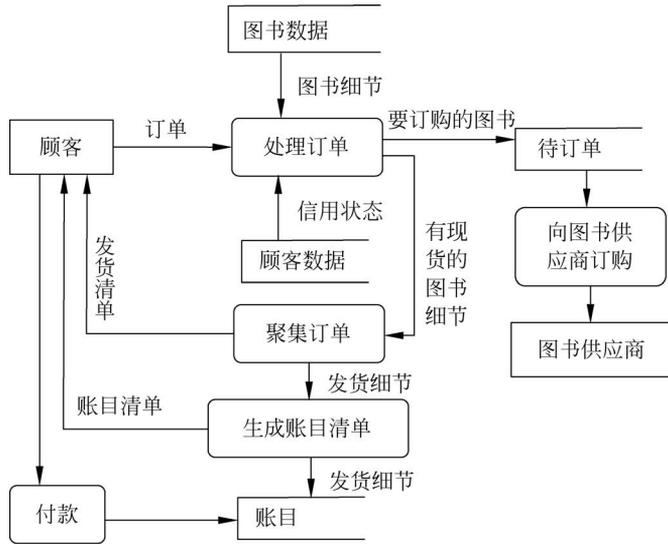


图 5.12 数据流图：第 3 次求精

对实现各个部分计算机化的各种可能方案进行分析。对数据流图各个部分的操作，必须决定是以批处理方式还是以联机方式执行。一般地，在需要处理大批数据和需要严密控制的情况下，批处理方式较好；而在处理数据量较小和使用内部机器的情况下，联机方式更好。

方案 1 是以批处理方式处理出账和订购图书，用联机方式处理订单的有效性检查、聚集订单和开发货清单。方案 2 是除发货票据使用联机方式或批处理方式外，其余都使用联机方式。具体采用哪种方案，需要经过后续分析得出。

(4) 定义数据。

在设计一个大型软件时，数据字典保存了所有数据元素信息。首先，对涉及的每个数据流、数据存储进行标识，然后再逐步求精细化每个数据流、数据存储。

本例的数据流和数据存储有“订单”“图书细节”“顾客细节”“图书数据”“顾客数据”“账目”“发货清单”等。“订单”可细化为“订单标识”“顾客细节”“图书细节”，其中“顾客细节”和“图书细节”还可进一步细化。

(5) 定义处理逻辑。

确定了软件的数据元素，就可以分析每个处理具体做什么了。例如，分析“生成账目”中“给教育部门打折扣”细节。书店提供向学校打折扣的细节是：若少于 10 本则给予 10% 的折扣；多于 10 本且少于 50 本则给予 15% 的折扣；多于 50 本则给予 20% 的折扣。这个处理逻辑适合用一个判定树来描述。

(6) 定义物理资源。

开发人员根据联机需求和数据元素，做出有关硬件的情况。此外，对每个文件必须指定一些信息，包括文件名、组织结构(排序、索引等)、存储介质和记录(字段数)。如果使用一个数据库管理系统，还要指定每个表的相关信息。

(7) 确定输入/输出规格说明。

必须指定数据输入格式，即使没有详细的布局，至少也要指定输入哪些内容。输入屏幕和打印输出格式也需要详细地确定。

(8) 确定数据规模与访问频率。

确定有关数值数据,包括输入量(每日或每小时)、打印报表的频率、每个报表的最后期限、CPU 和大容量存储之间传输的每一类记录的大小和数量,以及每个文件的大小。

(9) 确定硬件需求。

计算出磁盘大小和存储容量、备份存储要求,确定输出设备,以及给出购买设备的建议。

(10) 撰写系统需求规格说明文档。

根据以上结构化分析和建立的模型,建立系统标准化的需求规格说明文档,并进行评审。

5.7 软件需求规格说明文档

软件需求分析的描述通常采用一种规范的需求规格文档。软件需求规格说明文档即软件需求规格说明书(Software Requirement Specification, SRS),是需求分析任务的最终“产品”。软件需求规格说明文档通过分配给软件的功能和性能,建立完整的数据描述、详细的功能和行为描述、性能需求和设计约束的说明、合适的校验标准,以及其他和需求相关的信息。

1. 需求规格说明文档的读者

软件需求规格说明文档有一个很广泛的读者范围,从订购软件的高级管理者到负责开发软件的软件工程师。图 5.13 所示为软件需求规格说明文档的用户。

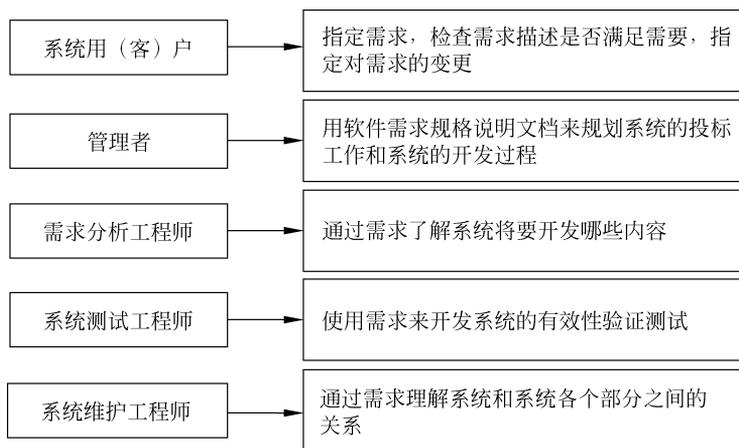


图 5.13 软件需求规格说明文档的用户

对于开发工程师们来说,软件需求规格说明文档一方面必须是清晰和具备可理解性的,另一方面必须是完备的和详细的。因为它是拟定设计的唯一可用的信息源。如果该文档中包含了遗漏、矛盾、模糊不清等错误,则设计中的这些错误将不可避免地被带到实现中。因此,需要能以一种非技术的描述形式有效地表述目标产品,以便于理解;而且应足够准确,以保证在开发周期的最后,交给客户的产品是无缺陷的。

2. 软件需求规格说明文档描述要求

软件需求规格说明文档是软件工程项目中最重要的一个文档。它相当于客户和开发商之间的一项合同。它精确地描述让软件做什么,以及软件的约束条件等。它还给软件设计

提供了一个蓝图,给软件验收提供了一个验收标准集。所以,软件需求规格说明文档应满足以下各个方面的描述要求。

- 只叙述软件的外部行为。
- 定义软件实现上的约束。
- 是容易改变的。
- 成为软件维护人员的参考工具。
- 记录软件的整个生命周期。
- 应对未料到的事件给出可接受的反应。

3. 软件需求规格说明文档标准

软件需求规格说明文档包括软件的用户需求和一个详细的软件需求描述。一般情况下,用户需求和软件需求被集中在一起描述。若是较简单,则用户需求在软件需求的引言部分给出;若是有很多项需求,则详细的软件需求可被分开到不同的文档中单独描述。

IEEE/ANSI 830—1993 标准和我国 GB/T 9385—2008 国家标准给出了关于软件需求规格说明文档的内容框架。下面是软件需求规格说明文档各个部分的简要介绍。

(1) 引言部分。陈述关于计划文档的背景和为什么需要该软件,解释软件是如何与其他软件协同工作的。

(2) 任务概述。陈述软件的目标、运行环境和软件的规模等。

(3) 数据描述。给出软件必须解决的问题的详细描述,并记录信息内容和关系、输入/输出数据及其结构。

(4) 功能描述。给出解决问题所需要的每个功能,包括每个功能的处理过程、设计约束等。

(5) 性能需求。描述性能特征和约束,包括时间约束、适应性等。

(6) 运行需求。给出交互的用户界面要求,与其他软/硬件的接口,以及异常处理等。

(7) 其他需求。给出软件维护性各个方面的要求。

5.8 小 结

结构化分析方法是一种典型的分析建模技术,已经获得了广泛应用。抽象和分解是结构化分析的指导思想。模型是忽略软件所有细节的抽象视图。结构化分析模型工具有数据流图、数据字典、实体-关系图和状态转换图等。数据流图和数据字典常用于结构化分析,用于描述软件的数据处理及其流程的信息,它们一起被称为软件逻辑模型。实体-关系图和状态转换图辅助描述软件所涉及的实体动态变化行为。

习 题

1. 什么是结构化分析?
2. 阐述结构化分析要创建的模型。
3. 给出面对面结对软件系统的用户管理的数据模型。
4. 给出网上书店的主要功能和优先级。

5. 给出网上书店的实体-关系图。
6. 给出电子表软件的状态图(提示:电子表具有3种状态,分别为显示时间、设置小时、设置分钟。模式按钮是外部事件,导致电子表发生状态变化)。
7. 图书馆管理系统中在检查读者能否借书时要考虑哪些规则?
8. 借书功能的可借性是否要考虑预约?
9. 请建立图书馆管理系统的实体-关系模型。