第3章

需求分析

教学提示：本章内容是传统的软件生命周期开发计划中的分析时期的最后一个阶段，是生命周期中最重要的一个阶段。在这个阶段要弄清用户的真实需求，包括功能需求、性能需求、数据需求、运行需求和将来可能提出的要求；建立系统的逻辑模型，书写需求规格说明文档。

教学目标：掌握需求分析的方法、数据流图的画法和需求规格说明的书写方法。

需求分析是软件开发生存周期中介于系统分析和软件设计之间的重要阶段，它是以系统规格说明等作为分析活动的基础，并从软件的角度对系统进行检查和调整。良好的分析活动有助于避免或尽早剔除早期出现的错误，从而大幅提高软件生产率，降低开发成本，保证软件质量。

3.1 需求分析的任务、过程与原则

在进行需求分析时，必须理解并描述问题的信息域；定义系统要完成的功能；描述作为外部事件结果的软件行为等。

3.1.1 需求分析的任务

在需求分析阶段，不仅用户要弄清楚软件的功能和性能，并对软件的最终结构提出要求，而且软件分析人员也应该准确地回答“系统必须做什么”这个问题。

需求分析阶段要从可行性研究阶段的结果出发。根据现有系统的物理模型建立现有系统的逻辑模型，通过研究现有系统存在的问题修改现有系统的逻辑模型，进而得到开发系统的逻辑模型，从而设计出开发系统的物理模型。

弄清用户的需求及确定系统必须完成哪些工作也就是对目标系统提出完整、准确、清晰、具体的要求，建立系统的逻辑模型是需求分析阶段的主要任务。需求分析采用的主要方法是结构化分析方法。系统逻辑模型主要通过系统的数据流图和数据字典共同体现。

3.1.2 需求分析的过程

需求分析的过程主要分为5个步骤，分别是验证可行性研究阶段得到的结果，分析系统的主要要求，得到系统的逻辑模型，修正系统的开发计划和验证软件需求。

首先要验证可行性研究阶段得到的结果。在需求分析阶段要进一步与用户交流，获得用户的真实需求，对在可行性研究阶段得到的关于系统的初步开发计划和数据流图进行修改。

然后分析和确定系统的主要需求，包括用户对系统的功能要求、性能要求、运行要求和数据要求，为了使系统在较长时期内可用，还应将未来用户可能提出的要求考虑进来。

下一步是导出系统的逻辑模型。系统的逻辑模型由数据流图和数据字典共同构成。在需求分析阶段主要采用交流技术，如访谈、情景分析等与用户进行充分的交流，让用户弄清自己到底需要哪些功能，并且能清楚地表达出来，进而对可行性研究阶段已经得到的数据流图进行修改。同时为数据流图书写数据字典，以更充分地描绘系统的逻辑模型。对于主要的处理算法，应通过加工处理的描述进行说明。

接下来是要修正系统的开发计划。由于通过与用户的进一步交流挖掘出了用户的许多真实需求，在可行性研究阶段制定的计划不一定适合这次开发，因此要修正开发计划，重新制定开发进度、预算等。

最后是验证软件的需求。为了验证软件的需求，需要建立一个非常重要的文档，即需求规格说明书。需求规格说明书是开发方获得用户需求的表达，是用户检验产品的标准。

3.1.3 需求分析的原则

需求分析阶段最重要的一条原则是要弄清系统必须做什么，而不要考虑如何具体实现。当与用户进行交流时，一定要挖掘用户的真实需求，将用户的需求在这个阶段“冻结”，不要再改变，否则将会造成很大的损失。

3.2 需求分析的方法

与用户沟通获取需求的方法有很多，最初是用访谈的形式，现在比较常用的是结构化分析方法和原型化方法等。

3.2.1 结构化分析方法

1. 结构化分析方法的原理

结构化分析(SA)方法是指面向数据流自顶向下、逐步求精地进行需求分析。使用“分解”和“抽象”两个基本手段控制开发的复杂性。分解就是指把大问题分割成若干小问题，然后分别解决；分解也可以分层进行，即先考虑问题最本质的属性，暂时把细节略去，以后再逐步添加细节，直至设计到最详细的内容，这就是抽象。

2. 数据流图

数据流图是描绘系统的逻辑模型，其中没有任何具体的物理元素，只描绘了信息在系

统中流动和处理的情况。在需求分析阶段建立的数据流图是采用结构化分析方法的基本思想获得的分层细化的数据流图。

(1) 数据流图的组成

数据流图是由 4 个要素组成的,分别是外部实体(也就是数据的源点或终点)、处理、数据流和数据存储。

① 外部实体(数据的源点或终点)。指系统外与系统有联系的人或事物。表达该系统数据的外部来源或去处,如顾客、工人、单位或另一个系统等。

② 处理。指对数据的逻辑处理功能,也就是对数据的变换功能,任何改变数据的操作都是处理,如一系列程序、单个程序或程序的一部分。

③ 数据流。指处理功能的输入或输出,是运动中的数据,如信件、票据等。

④ 数据存储。是静止状态的数据,表示数据保存的"地方"。这个地方并不是指保存数据的物理地点或物理介质,而是指数据存储的逻辑描述,如可以表示一个文件、文件的一部分、数据库的元素或记录的一部分等。

(2) 数据流图的符号

数据流图中的符号主要包括基本符号和附加符号。

① 基本符号,如图 3.1 所示。

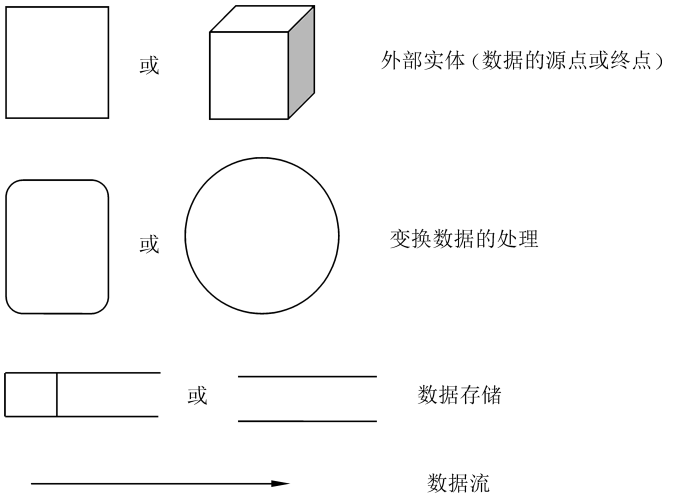


图 3.1　数据流图基本符号

② 附加符号,如图 3.2 所示。

"*"表示数据流之间是"与"的关系。

"+"表示数据流之间是"或"的关系。

"⊕"表示只能从中选择一个(互斥关系)。

(3) 分层数据流图

绘制分层细化的数据流图的基本原则是:一是要保持父图和子图信息的连续性,二是当进一步划分将涉及如何具体实现一个功能时,细化结束。

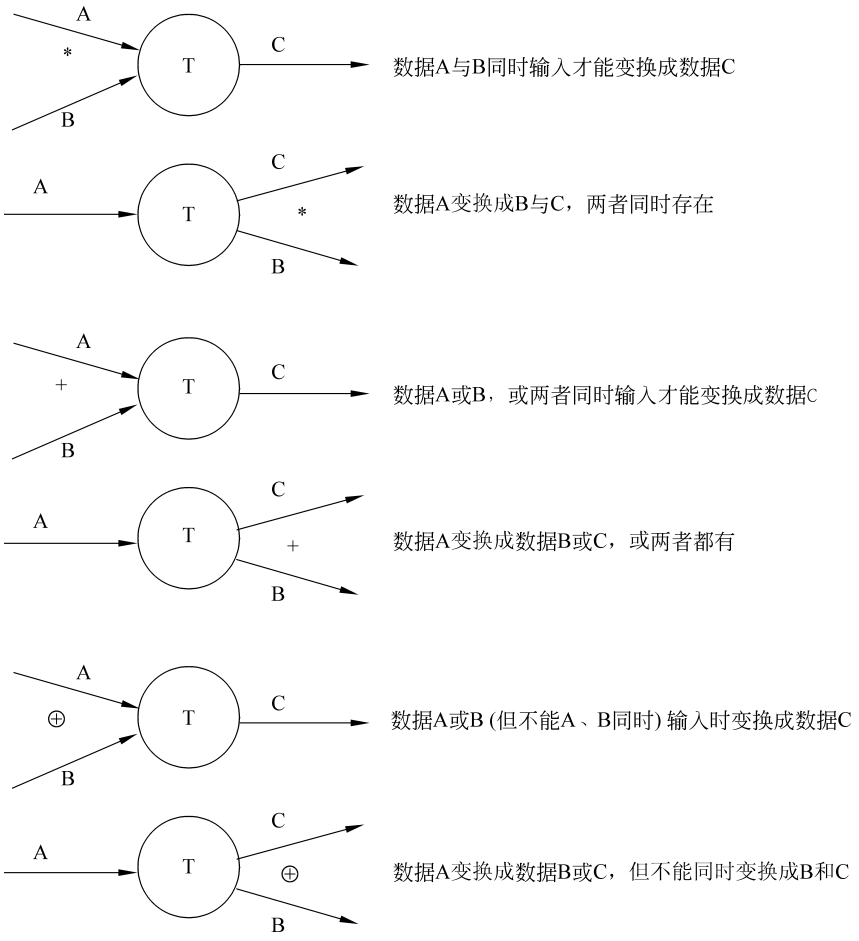


图 3.2 数据流图附加符号

【例 3.1】 画出××培训中心管理系统的数据流图，如图 3.3 至图 3.5 所示。

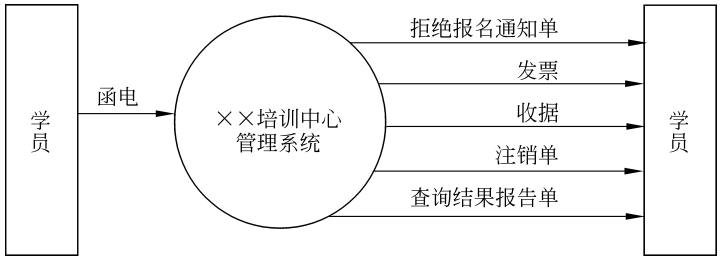


图 3.3 ××培训中心管理系统基本模型

问题描述：

××培训中心当有学员报名时，根据系统中的课程文件的内容，查看该门课程是否已经额满(额满不能报名，未满可以报名)，以及该学员是否已经进行了学员登记(即该学员是否为新学员)。报名后，为该学员打印一份报名单，标出最迟的交费日期及所报课程信

息。对于已经报过名的(即拿到报告单)的学员,付款时应根据发票上的应交学费金额收取学费并更新账目文件,复审无误后为学员开具收据。有的学员报名后,发现有的课程不想学了,当要注销时,系统对相关的课程文件和学员文件进行修改,并退款进行财务处理,更新账目文件。当学员咨询时,可以将其所要查询的内容打印成报告单。学员的报名、付款、查询以及注销统称为事务。

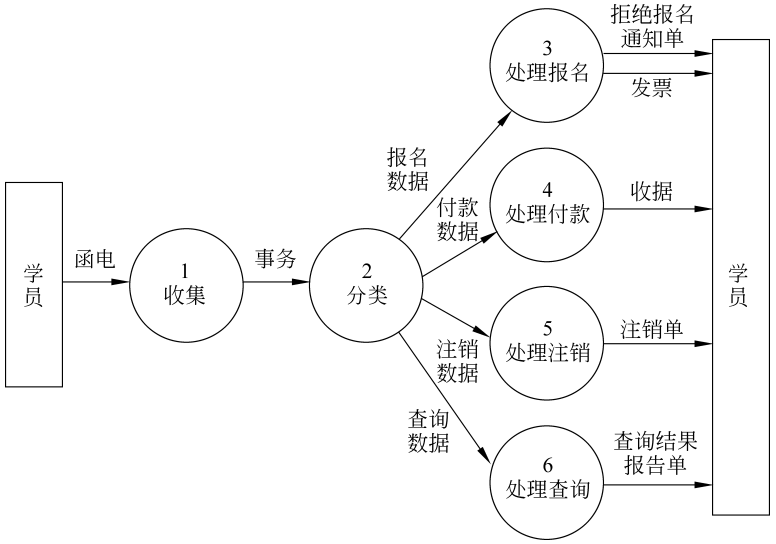


图 3.4　××培训中心管理系统功能级数据流图

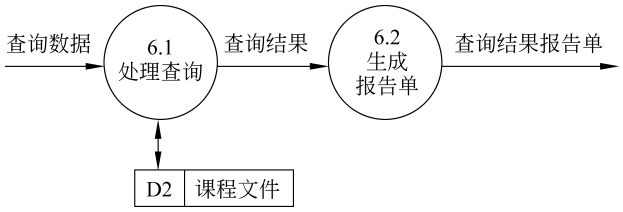


图 3.5　进一步细化后的数据流图

3. 数据字典

数据字典是关于数据的信息的集合,也就是对数据流图中包含的所有元素的定义的集合。数据字典与数据流图共同构成系统的逻辑模型,没有数据字典,数据流图就不严格;没有数据流图,数据字典也难以发挥作用。只有将数据流图和对数据流图中每个元素的精确定义(数据字典)放在一起,才能共同构成系统的规格说明。

数据字典由对下列 4 类元素的定义组成:数据流、数据流分量(即数据元素)、数据存储和处理(用 IPO 图或 PDL 描述更方便)。除了数据定义之外,还包含关于数据的一些其他信息:一般信息——名字、别名、描述;定义——数据类型、长度、结构;使用特点——值的范围、使用频率、使用方式;控制信息——来源、用户、使用它的程序、使用权、改变权;

分组信息——父结构、从属结构、物理位置(记录、文件、数据库)。

数据字典中的定义就是对数据自顶向下的分解,通过数据元素组成数据的方式和分类。

① 顺序:以确定的次序连接两个或多个分量。

② 选择:从两个或多个可能的元素中选取一个。

③ 重复:把指定的分量重复零次或多次。

④ 可选:一个分量是可有可无的(不重复或重复一次)。

数据字典中的常用符号见表 3.1。

表 3.1 数据字典中的常用符号

符 号	含 义	解 释
=	被定义为	
+	与	例如,x=a+b,表示 x 由 a 和 b 组成
[…,…] [… | …]	或	例如,x=[a|b]或 x=[a,b],表示 x 由 a 或 b 组成
{…}	重复	例如,x={a},表示 x 由 0 个或多个 a 组成
m{…}n	重复	例如,x=3{a}8,表示 a 在 x 中至少出现 3 次、至多 8 次
(…)	可选	例如,x=(a),表示 a 可在 x 中出现,也可不出现
"…"	基本数据元素	例如,x="a",表示 x 是取值为 a 的数据元素
…	连接符	例如,x=1…9,表示 x 可取 1～9 的任意一值

【例 3.2】 写出××培训中心管理系统的数据字典。

事务=[报名数据|付款数据|注销数据|查询数据]

报名数据=姓名+课程代号+电话+地址

付款数据=姓名+课程代号+课程名+学时数+学费+开课时间+最晚缴费时间

注销数据=姓名+课程代号+学费+备注

查询数据=[类别|课程号]

发票=姓名+课程代号+课程名+学时数+学费+开课时间+最晚缴费时间

收据=姓名+课程代号+课程名+交款额+交款日期

注销单=姓名+课程代号+课程名+注销日期

查询结果报告单=课程代号+课程名+学时数+开班人数+学费+开课时间+已有人数

学员文件=姓名+课程代号+电话+地址

课程文件=课程代号+课程名+学时数+开班人数+学费+开课时间+已有人数

账目文件=姓名+交款额+交款日期

拒绝数据=课程代号+开班人数+已有人数

拒绝报名通知单=课程代号+开班人数+已有人数

3.2.2 原型化方法

按照传统的软件开发方法，目标软件往往需要在等待漫长的开发时间之后才能得到目标软件的最初版本。此时，由于软件研制的各个阶段中各种错误和偏差的积累，用户常常会对目标软件提出许多修改意见，有时甚至会全盘否定，导致开发失败，这无疑会造成人力、物力和财力的巨大浪费。为了防止这种情况的发生，降低开发风险，在需求分析阶段常常采用原型化方法。

1. 原型化方法的基本思想

在软件开发的早期，快速建立目标软件系统原型可以让用户在对原型进行评估的同时提出修改意见。当原型几经改进最终确认后，它将从软件设计和编码阶段进化成软件产品；或者设计和编码人员遵循原型所确立的外部特征以实现软件产品。

2. 采用原型化方法的步骤

确定采用原型化方法之后，分析人员要遵循以下步骤。

① 采用一种分析技术或方法生成一个简化的需求规格说明。

② 对上述需求规格说明进行评审并通过后，生成设计规格说明。通常，为了快速生成原型，这种设计只注重软件的总体结构、用户界面和数据设计，不注重过程内部的控制流设计。

③ 使用可重用软部件库、用户界面自动生成器等工具生成可运行的软件原型并进行测试和改进。

④ 将原型提交给用户进行评估，同时征询改进意见。

⑤ 上述过程将反复进行，直到用户完全满意。此时的原型已全面、准确地反映了目标软件在外部行为方面的需求，可以作为需求规格说明的一部分而成为软件设计和编码的基础。

3. 快速原型开发模型

在获得用户需求时，分析人员不断与用户沟通交流，但获得的需求仍然十分有限。分析人员通过列出用户已经提出的需求和可能提出的需求、让用户重新判断等方法与用户沟通，但分析人员仍无法确定用户所描述的需求就是他们想要开发的系统，也就是无法获得最真实的需求信息。

此时，快速原型开发模型应运而生，它通过快速构建一个可运行的原型系统，让用户试用原型并收集用户反馈意见的方法获取用户的真实需求。快速原型模型有两种，一种是在与用户交流之前就已经建立的原型系统，将这个系统拿给用户看，让用户说出自己所要的系统与这个系统的主要区别，这种快速原型称为抛弃型。另一种称为进化型，这种快速原型先与用户交流，获得一定的信息（用户的需求），然后快速搭建一个原型系统交给用户看，用户从这个系统出发，找出这个系统与其心目中的系统的区别，开发人员从而获得更多的需求信息，进一步修改系统以满足用户的需求。

3.2.3 系统动态分析

在对系统进行需求分析时，不仅要建立系统的逻辑模型，还要建立系统的行为模型，也就是对系统进行动态分析。进行动态分析的主要工具是状态迁移图。

状态迁移图简称状态图，用来描绘系统的状态及引起系统状态转换的事件，并指明作为特定事件的结果系统将做出哪些动作。

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，它规定了系统对事件的响应方式。

初始状态简称初态，指系统启动时进入的状态。

最终状态简称终态，指系统运行结束时达到的状态。

事件是指在某个特定时刻发生的事情，它是对引起系统做出动作或(和)从一个状态转换到另一个状态的外界事件的抽象。

符号表示如图 3.6 所示。

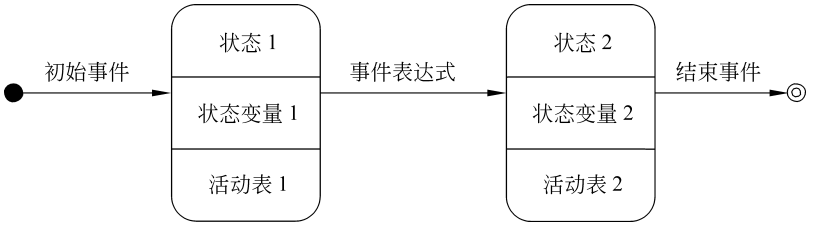


图 3.6 状态图中使用的主要符号

【例 3.3】 复印机的工作过程大致如下。未接到复印命令时处于闲置状态，一旦接到复印命令则进入复印状态，完成一个复印命令规定的工作后又回到闲置状态，等待下一个复印命令；如果执行复印命令时发现没纸，则进入缺纸状态，发出警告，等待装纸，装满纸后进入闲置状态，准备接收复印命令；如果复印时发生卡纸故障，则进入卡纸状态，发出警告，等待维修人员排除故障，故障排除后回到闲置状态。请用状态迁移图描绘复印机的行为。

答案如图 3.7 所示。

3.2.4 Jackson 系统开发方法和 Warnier 系统开发方法

Jackson 系统开发方法和 Warnier 系统开发方法都是结构化分析的主要方法。

1. Jackson 系统开发方法

Jackson 系统开发方法是英国人 Michael Jackson 在 1970 年提出的，它的发展分为 2 个阶段，前期(20 世纪 70 年代)主要研究以处理数据为主的结构化程序设计，称为 Jackson 结构程序设计方法，简称 JSP(Jackson Structured Programming)方法；后期(20 世纪 80 年代)吸收了软件工程的功能分割、逐步求精等设计思想，集中研究软件系统的开发，提出了 Jackson 系统开发方法，简称 JSD(Jackson System Development)方法。

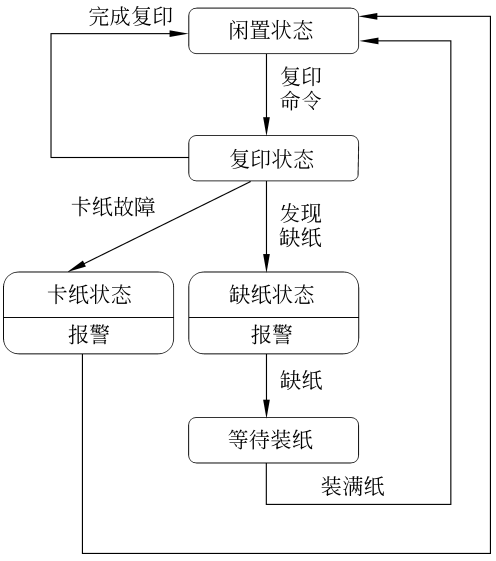


图 3.7　复印机的状态迁移图

Jackson 系统开发方法的基本思想是从数据结构出发建立对应的程序结构，所以这样的程序结构特别适合于设计企事业事务管理类的数据处理系统。

JSD 方法和面向数据流的结构化分析与设计方法(SADT)都是由信息驱动的，都是将信息转换为软件的程序结构。但是 JSD 方法不直接利用数据流图，因此不区分变换型结构或事务型结构。而且 JSD 方法的最终目标是生成软件的过程性描述，没有特别考虑程序模块化结构，模块只作为过程的副产品而出现，模块独立性也没有特别强调。

JSD 方法实际上是支持软件分析与设计的一组连续的技术步骤，具体操作如下。

① 实体动作分析：从问题的描述中提取软件系统要产生和运用的实体(人、物或组织)，以及现实世界作用于实体上的动作(事件)。

② 实体结构分析：把作用于实体的动作或由实体执行的动作按时间发生的先后次序排序以构成进程，并用一个层状的 Jackson 结构图表示。

③ 定义初始模型：把实体和动作表示成一个进程模型，定义模型与现实世界的联系。模型系统的规格说明可用系统规格说明图(System Specification Diagram，SSD)表示。

④ 功能描述：说明与已定义的动作相对应的功能，为已定义的动作加入功能函数。

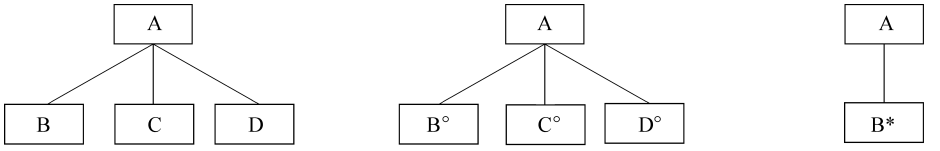
⑤ 决定系统时间特性：为进程加入时间因素，对进程调度特性进行评价和说明。

⑥ 实现：设计组成系统的硬件和软件，实现系统的原型。

JSD 方法的前 3 步属于需求分析阶段，后 3 步属于设计阶段。

与程序结构一样，Jackson 系统开发方法的数据结构也有顺序、选择和重复这 3 种类型。

① 顺序结构：顺序结构的数据由一个或多个数据元素组成，每个数据元素按确定次序出现一次，如图 3.8(a)所示。



(a) 顺序结构　　(b) 选择结构　　(c) 重复结构

图 3.8　Jackson 数据结构图

② 选择结构：选择结构的数据包含两个或多个数据元素，每次使用这个数据时，按一定条件从这些数据元素中选择一个，如图 3.8(b)所示，图中的“°”表示选择。

③ 重复结构：重复结构的数据根据使用时的条件由一个数据元素出现零次或多次构成，如图 3.8(c)所示，图中的“*”表示重复。

与 Jackson 结构图对应，可用伪码将 Jackson 结构图表示的 3 种程序结构转换为语言表示。

顺序结构：

```
A seq
B
C
D
A end
```

选择结构：

```
A select 条件 1
B
A        or 条件 2
C
A        or 条件 3
D
A end
```

重复结构：

```
A iter until(或 while)条件
B
A end
```

2. Warnier 系统开发方法

法国计算机科学家 Warnier 提出了表示信息层次结构的另外一种图形工具，Warnier 图应用树状结构描绘信息。用 Warnier 图可以表明信息的逻辑组织，它不仅可以指出一类信息或一个信息量是重复出现的，也可以表示特定信息在某一类信息中是有条件出现的。因为重复条件约束是说明软件处理的基础，所以 Warnier 图成为软件设计的工具。

图 3.9 所示是用 Warnier 图描述一类软件产品的例子，它说明了这种图形工具的用法。图中的花括号用来区分数据结构的层次，在一个花括号中的所有名字都属于一类信息；异或信息(⊕)表明一类信息或一个数据元素在一定条件下才出现，而且在这个符号的上方和下方的 2 个名字所代表的数据只能出现一个；在一个名字下面或右边的括号中的数字 P_i 说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。

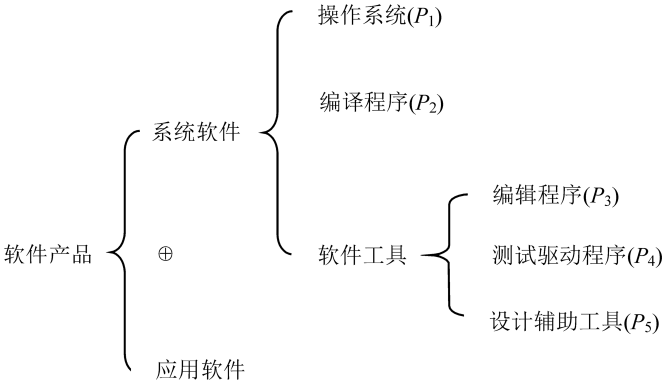


图 3.9　Warnier 图的一个例子

3.3　需求规格说明与评审

在进行需求分析时要将需求分析的有关数据记录下来，并按照有关标准和原则形成需求规格说明书。

3.3.1　需求规格说明书

需求规格说明书是需求分析阶段的最后成果，是软件开发中的重要文档之一。

需求规格说明书的内容是把软件计划中确定的软件范围加以展开，制定出完整的信息描述、详细的功能说明、恰当的检验标准及其他与要求有关的数据。

(1) 概述

从系统的角度描述软件的目标和任务。

(2) 数据描述

对软件系统所必须解决的问题做出详细说明。

- 数据流图
- 数据字典
- 系统接口说明
- 内部接口

(3) 功能描述

描述为解决用户问题所需要的每项功能的过程细节，对每项功能给出处理说明和在设计时需要考虑的限制条件。

- 功能
- 处理说明
- 设计的限制

(4) 性能描述

说明系统应达到的性能和应满足的限制条件,以及检测的方法和标准、预期的软件响应和可能需要考虑的特殊问题。

- 性能参数
- 测试种类
- 预期的软件响应
- 应考虑的特殊问题

(5) 参考文献目录

应包括与该软件有关的全部参考文献,其中包括前期的其他文档、技术参考资料、产品目录手册以及标准等。

(6) 附录

包括一些补充资料,如列表数据、算法的详细说明、框图、图表和其他资料。

需求规格说明书的作用主要有以下三方面。

作为用户和软件人员之间的共同文档,为双方相互了解提供基础;反映用户问题的结构,可以作为软件人员进行设计和编码的基础;作为验收的依据,即作为选取测试用例和进行形式验证的依据。

3.3.2 需求评审

需求分析文档完成后,应由用户和系统分析员共同进行需求评审。需求评审需要有用户方和承包商方的人员共同参与,检查文档中的不规范之处和遗漏之处。

在需求评审过程中,需要检查的主要内容如下。

- 系统定义的目标是否与用户的要求一致。
- 系统需求分析阶段提供的文档资料是否齐全。
- 文档中的所有描述能否完整、清晰、准确地反映用户要求。
- 与所有其他系统成分的重要接口是否都已经描述。
- 被开发项目的数据流与数据结构是否足够。
- 所有图表是否清楚,在不做补充说明时能否理解。
- 主要功能是否已包括在规定的软件范围之内,是否都已充分说明。
- 软件的行为和它必须处理的信息及必须完成的功能是否一致。
- 设计的约束条件或限制条件是否符合实际。
- 是否考虑了开发的技术风险。
- 是否考虑了软件需求的其他方案。
- 是否考虑了将来可能会提出的软件需求。
- 是否详细制定了检验标准,它们能否对系统定义成功进行确认。
- 有没有遗漏、重复或不一致的地方。

- 用户是否审查了初步的用户手册或原型。
- 软件开发计划中的估算是否受到了影响。

首先在宏观上进行评审，保证规约是完整、一致、精确的，然后细致地评审每个域，不仅要检查概要描述，还要检查需求被陈述的方式。

最后把需求评审期间找出的冲突、矛盾、错误和遗漏正式记录下来，由系统用户、系统购买者和开发商共同协商解决这些问题的方案。

小　结

本章主要阐述了需求分析的过程和在分析工程中应用的基本原理和主要图形工具。本章重点介绍了数据流图的画法、数据字典的写法、需求规格说明书的内容等。在学习数据流图的过程中，不仅要掌握基本元素及图形符号，还要掌握绘制分层数据流图的步骤以及在细化过程中的指导原则。数据字典的定义要规范，要使用恰当的符号。需求规格说明书是需求分析阶段的重要文档。

习　题

1. 根据问题描述绘制数据流图，写出数据字典。

为方便储户，某银行拟开发计算机储蓄系统。储户填写的存款单或取款单由业务员输入系统，如果是存款，则系统记录存款人姓名、住址、存款类型、存款日期、利率等信息，并打印存款单给储户；如果是取款，则系统计算利息并打印利息清单给储户。

2. 根据问题描述绘制数据流图，写出数据字典。

某装配厂有一存放零件的仓库，仓库中现有的各种零件的基本信息及库存量临界值都记录在库存清单主文件中。当库存量有变化时，应及时修改库存清单主文件；若库存量小于临界值，则报告给采购部门，每天送一次订货报表。该厂使用一台计算机处理和更新库存清单主文件和产生订货报表的任务。零件库存量的每次变化称为一个事务，由 CRT 终端输入计算机；系统重点库存清单程序对事务进行处理，更新并存储在磁盘上的库存清单主文件中，并且把必要的订货信息写在磁带上。最后，每天由报告生成程序读一次磁带，并打印订货报告。

3. 说明绘制分层数据流图的步骤。

4. 说明绘制分层数据流图的指导原则。