

# 第4章 软件需求分析

---

本章内容：

4.1 需求分析

4.2 结构化分析

4.3 系统流程图

4.4 数据流图

4.5 数据字典

# 第4章 需求分析

---

软件生存周期由三个时期组成:软件定义、软件开发和软件维护。

软件定义时期通常又可分为三个阶段，即问题定义、可行性研究和需求分析阶段。

其中，需求分析与定义处于整个软件正式开发的起始阶段，其成功与否直接关系到整个软件的开发与维护，有着举足轻重的作用。

# 4.1 需求分析

---

需求分析是软件定义时期的最后一个阶段，其基本任务是回答“系统必须做什么”这个问题。

需求分析不是确定系统怎样完成工作，而是确定系统必须完成哪些工作。

需求分析阶段结束时，要提交详细的数据流图、数据字典和算法描述。需求分析的结果是系统开发的基础，它关系到系统的质量和成败的关键。因此，必须用行之有效的方法进行严格的审查验证。

---

## **4.1.1 需求分析的特点**

---

在进行可行性研究和项目开发计划以后，如果确认开发一个新的软件系统是必要的而且是可能的，那么就可进入需求分析阶段。

---

需求分析虽处于软件开发过程的开始阶段，但它对于整个软件开发过程以及软件产品质量是至关重要的。需求分析是指开发人员要进行细致的调查分析，准确理解用户的要求。将用户非形式的需求陈述转化为完整的需求定义，再由需求定义转换到相应的形式功能规约的过程。它的难点主要体现在以下几个方面：

---

---

## 1. 需求易变性

用户在开始时提出一些功能需求，当对系统有一定的理解后，会提出一些需求。以后随着理解的深入而不断提出新的需求。用户需求的变动是一个极为普遍的问题。即使是部分变动，也往往会影响到需求分析的全部，导致不一致性和不完备性。

---

## 2. 问题的复杂性

一方面是由用户需求所涉及的因素繁多引起的，如运行环境和系统功能等；另一方面是扩展的应用领域本身的复杂性。

---

### 3. 交流障碍

需求分析涉及人员较多，系统分析员要与软件系统用户、问题领域专家、需求工程师和项目管理员等进行交流。但是这些人具备不同的背景知识，处于不同的角度，扮演不同角色，造成了相互之间交流的困难。

---

## 4. 不完备性和不一致性

由于用户各类人员对于系统的要求所处的角度不一样，所以对问题的陈述往往是不完备的，其各方面的需求还可能存在着矛盾。需求分析要消除其矛盾，形成完备及一致的定义。

为了克服需求分析的困难，人们展开的各种研究都是围绕着需求分析的方法、自动化工具（如CASE技术）及形式化需求分析等方面进行研究。需求分析的方法在应用中已有丰富的应用经验。

---

## 4.1.2 需求分析的原则

---

为使需求分析科学化，在软件工程的分析阶段中提出了许多需求分析方法。在已提出许多软件需求分析与说明的方法中，每一种分析方法都有独特的观点和表示法，但都适用下面的基本原则。

---

(1) 可以把一个复杂问题按功能进行分解并可逐层细化。通常，如果软件要处理的问题涉及面太大，关系太复杂就很难理解。若划分成若干部分，并确定各部分间的接口，那么就可完成整体功能。在需求分析过程中，软件领域中的数据、功能和行为都可以划分。

---

(2) 必须能够表达和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。其中数据流是数据通过一个系统时的变化方式。功能域则是反映数据流、数据内容和数据结构三方面的控制信息。

---

(3) 建立模型。所谓模型就是所研究对象的一种表达形式。因此，模型可以帮助分析人员更好地理解软件系统的信息、功能和行为，这些模型也是软件设计的基础。

在软件工程中著名的结构化分析方法和面向对象分析方法都遵循以上原则。

## 4.1.3 需求分析的任务

---

需求分析的基本任务是要准确地理解旧系统，定义新系统的目标。为了满足用户需要，回答系统必须“做什么”的问题。在可行性研究和项目开发计划阶段对这个问题的回答是概括的、粗略的。需求分析的任务还不是确定系统怎样完成它的工作，仅仅是确定系统要完成哪些工作，也就是对系统提出完整、准确、清晰、具体的要求。

---

这个时期的工作可以从可行性阶段的数据流图等文档出发，划分出系统必须完成的许多基本功能，研究这些功能并进一步具体化。要实现详细的数据流图、数据字典和算法描述。需求分析阶段的结果是开发的基础，关系到系统的成败和质量。要完成好下面的任务：

---

## 1. 问题明确定义

在可行性研究的基础上，双方通过交流，对问题都有进一步的认识，所以可确定对问题的综合需求。这些需求包括以下几种。

(1) 功能需求：指所开发的软件必须具备什么样的功能。

(2) 性能需求：要开发软件的技术性能指标，如访问时延、存储容量、运行时间等限制。

(3) 环境需求：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。

(4) 用户界面需求：用户操纵界面的形式、输入/输出数据格式、数据传递的载体等。

(5) 系统的可靠性、安全性、可移植性和可维护性等方面的需求。

---

## 2. 导出软件的逻辑模型

分析人员根据前面获取的需求资料，要进行一致性的分析检查，在分析、综合中逐步细化软件功能，划分成各个子功能。同时对数据域进行分解，并分配到各个子功能上，以确定系统的构成及主要成分。最后要用图文结合的形式，建立起新系统的逻辑模型。

---

### 3. 编写文档

通过分析确定了系统必须具有的功能和性能，定义了系统中的数据，描述了数据处理的主要算法。应该把分析的结果用正式的文件记录下来，作为最终软件的部分材料。编写文档的步骤如下。

---

(1) 编写“需求说明书”，把双方共同的理解与分析结果用规范的方式描述出来，作为今后各项工作的基础。主要描述系统的概貌、功能要求、性能要求、运行要求和将来可能提出的要求。用图形工具描述数据流图和系统的主要算法。要包括用户需求和系统功能之间的参照关系、设计约束等。

要描述建立起来的数据字典、数据结构的层次，以及对于存储分析的结果。

(2) 编写初步用户使用手册，要从用户使用系统的角度来描述系统的用户要求。着重反映被开发软件的用户功能界面和用户使用的具体要求、使用系统的主要步骤与方法、系统用户的责任等。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) 修改完善项目开发计划。在需求分析阶段对开发的系统有了更进一步的了解，所以能更准确地估计开发成本、进度及资源的使用计划要求。因此对原计划要进行适当修正。

## 4.1.4 需求分析的方法

---

需求分析就是研究问题域，产生一个满足用户需求的系统模型。这个系统模型应能正确地描述问题域和系统责任，并使后续开发阶段的有关人员能根据这个模型继续进行工作。软件分析方法比较多，其中最有影响的是功能分解法、数据流法、信息建模法和面向对象的分析。前三种分析方法在历史上发挥过应有的作用，直到今天仍然被许多开发者所采用。

---

## 1. 功能分解方法

功能分解 = 功能

- + 子功能
- + 功能接口

功能分解法（Function Decomposition）以系统需要提供的功能为中心来组织系统。首先定义各种功能，然后把功能分解为子功能，同时定义功能之间的接口。

---

## 2. 结构化分析方法

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

信息建模 = 实体（对象）

- + 属性
- + 关系
- + 父类型/子类型
- + 关联对象

---

## 4. 面向对象的分析

面向对象 = 对象、类

- + 结构与连接
- + 继承
- + 封装
- + 消息通信

---

以上公式仅仅表示面向对象分析方法中几项最重要的特征，全面的论述将在后面展开。简单说来，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）。

## 4.2 结构化分析

---

在结构化方法的发展历程上，它是随着结构化程序设计（Structured Programming，简称SP）方法的提出、结构化设计（Structured Design，简称SD）方法的出现直到结构化分析（Structured Analysis，简称SA）方法提出才逐渐形成的。

---

结构化方法是分析、设计到实现都使用结构化思想的软件开发方法，实际上它由三部分组成：结构化分析、结构化设计和结构化程序设计。它也是一种实用的软件开发方法。它是根据某种原理，使用一定的工具，按照特定步骤工作的软件开发方法。它遵循的原理是自顶向下、逐步求精，使用的工具有数据流图（DFD）、数据字典、判定表、判定树和结构化语言等。

---

---

## 1. 基本思想

结构化方法总的指导思想是自顶向下、逐步求精，它的两个基本原则是抽象与分解。

## 2. 特点

结构化方法具有以下特点：

- (1) 它是使用最早的开发方法，使用时间也最长。
- (2) 它应用最广，特别适合于数据处理。
- (3) 相应的支持工具多，发展较为成熟。

---

### 3. 优点

结构化方法一经问世，就显示出了它的以下几大优点：

- (1) 简单、实用。
  - (2) 适合于瀑布模型，易为开发者掌握。
  - (3) 成功率较高，据美国1000家公司统计，该方法的成功率高达90.2%，名列第二，仅次于面向对象的方法。
  - (4) 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。
-

---

## 4. 存在问题

结构化方法存在以下一些问题。

- (1) 对于规模大的项目，特别复杂的应用不太适应。
- (2) 难以解决软件重用的问题。
- (3) 难以适应需求的变化。
- (4) 难以彻底解决维护问题。

## 4.2.1 自顶向下逐层分解

---

在面对一个复杂的问题进行分析时，如果既要考虑问题的各方面又要分析问题的每一个细小环节。那么越想搞清楚问题，就越搞不清楚。因此，一开始就不不是考虑到问题所有方面和问题的所有细节，而采取的策略往往是分解。

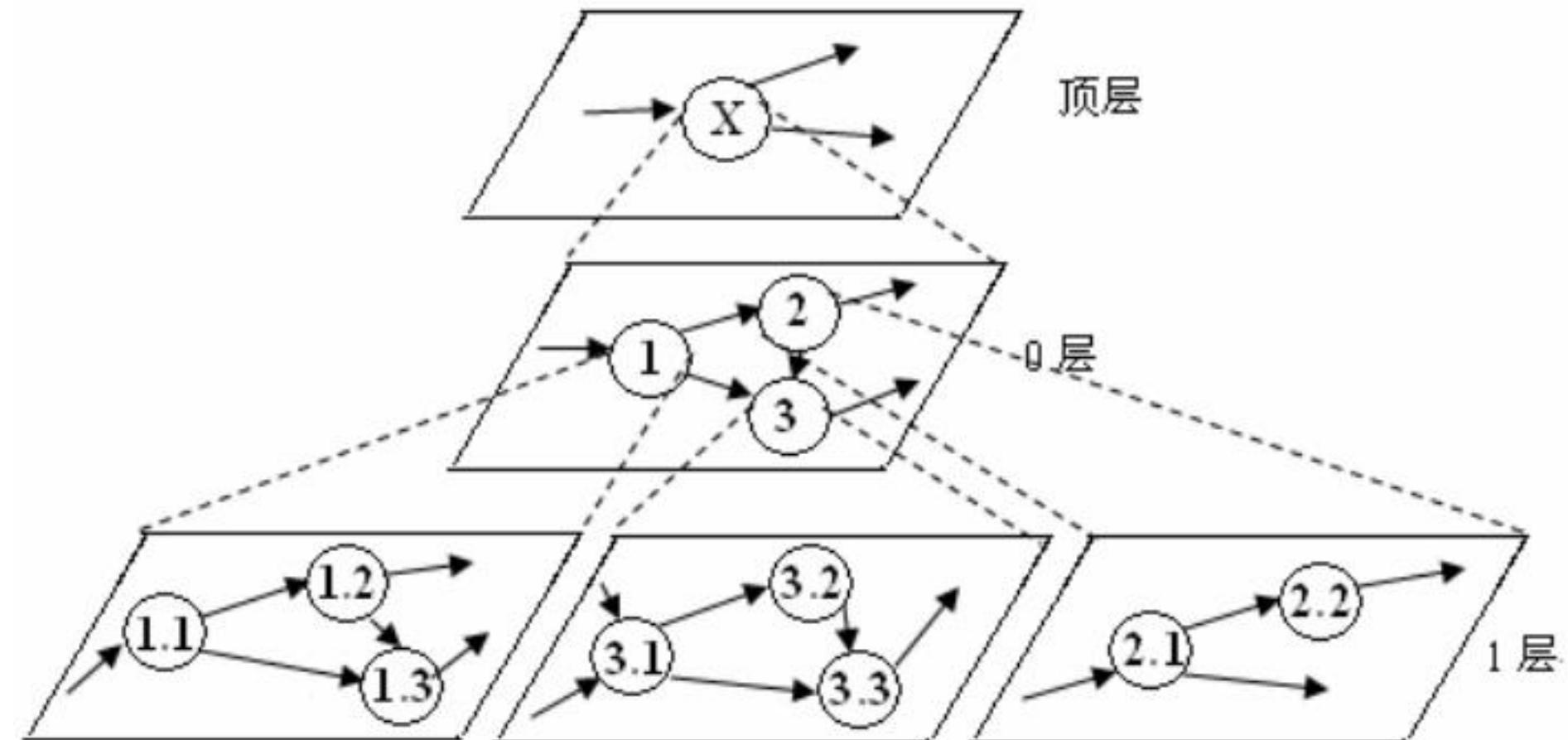


图4-1 对一个问题的逐层分解

---

图4-1的顶层抽象地描述了整个系统，底层具体地画出了系统的每一个细节，而中间层是从抽象到具体的逐步过渡。这种层次分解使分析人员分析问题时不至于一下子陷入细节，而是逐步地去了解更多的细节，如在顶层，只考虑系统外部的输入和输出，其他各层反映系统内部情况。

## 4.2.2 结构化分析步骤

---

要对一个系统进行结构化分析，首先要明确这一阶段的任务是要搞清楚“做什么”。为此就要对现行系统有一定了解，在此基础上修改要变化的部分而形成新系统。具体步骤如下。

# 1. 建立现行系统的物理模型

所谓现行系统（也称当前系统）指目前正在运行的系统，也是需要改进的系统。现行系统可能是正在计算机上运行的软件系统，也可能是人工的处理系统。通过了解现行系统的工作过程，对现行系统进行详细调查，收集资料，将看到的、听到的、收集到的信息和情况用图形或文字描述出来。也就是用一个模型来反映自己对现行系统的理解，如画系统流程图（后面介绍）。这一模型包含了许多具体因素，反映现实世界实际情况。

## 2. 抽象出现行系统的逻辑模型

在系统分析中需要建立功能模型时，可以通过上述建立的物理模型，它反映了系统过去“怎么做”的具体实现。要构造新的逻辑模型就要去掉物理模型中非本质的因素（如物理因素），抽取出本质的因素。所谓本质的因素是指系统固有的、不依赖运行环境变化而变化的因素，任何实现均这样做。非本质的因素不是固有的，随环境不同而不同，随实现不同而不同。运用抽象原则对物理模型进行认真的分析，区别本质因素和非本质因素，去掉非本质因素，形成现行系统的逻辑模型。

### 3. 建立目标系统的逻辑模型

目标系统是指待开发的新系统。有了现行系统的逻辑模型后，就将目标系统与现行系统逻辑进行分析，比较其差别，即在现行系统的基本上决定变化的范围，把那些要改变的部分找出来，将变化的部分抽象为一个加工，这个加工的外部环境及输入输出就确定了。然后对“变化的部分”重新分解，分析人员根据自己的经验，采用自顶向下、逐步求精的分析策略，逐步确定变化部分的内部结构，从而建立目标系统的逻辑模型。

---

## 4. 进一步补充和优化

目标系统的逻辑模型只是一个主体，为了完整地描述目标系统，还要作一些补充。补充的内容包括它所处的应用环境及它与外界环境的相互联系；说明目标系统的人机界面；说明至今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。

## 4.3 系统流程图

---

在可行性研究时需要了解和分析现行系统，概括对现行系统的认识。进入设计阶段后要把新系统的逻辑模型转变成为物理模型，需要描述未来新系统的概貌。如何描述该系统的概貌，系统流程图是描述的传统工具。其基本思想是用图形符号以黑盒方式描述系统的每个部件。系统流程图表达的是系统各部件间的流动情况，不是对信息进行加工处理的控制过程。

---

## 1. 系统流程图的作用

用系统流程图来描述物理系统。所谓物理系统，就是一个具体实现的系统，也就是描述一个单位、组织的信息处理的具体实现的系统。在可行性研究中，对于旧系统的理解和对于新系统的构想，可以通过画出系统流程图来表示要开发项目的大致处理流程、范围和功能等。系统流程图不仅能用于可行性研究，还能用于需求分析阶段。

---

## 2. 系统流程图的符号

项目小组开始工作时，制定的系统标准包括各种表示符号。系统流程图的符号一般使用如表4-1所示的内容。

符号	名称	说明
	处理	能改变数据值或位置的加工，例如，程序模块、处理机等都是处理。
	输入/输出	表示输入或输出，是一个广义的不指明具体设备的符号。
	连接	指出转到图的另一部分或从图的另一部分转来，通常在同一页。
	换页连接	指出转到另一页图上或由另一页图转来。
	数据流	用来连接其他符号，指明数据流动方向。
	文档	通常表示打印输出，也可表示用打印终端输入数据。
	联机存储	表示任何种类的联机存储，包括磁盘、软盘和海量存储器件等。
	磁盘	磁盘输入/输出，也可表示存储在磁盘上的文件或数据库。
	显示	CRT 终端或类似的显示部件，可用于输入或输出，也可既输入又输出。
	人工输入	人工输入数据的脱机处理，例如，填写表格。
	人工操作	人工完成的处理，例如，会计在工资支票上签名。
	辅助操作	使用设备进行的脱机操作。
	通信链路	通过远程通信线路或链路传送数据。

表4-1 系统流程图的符号

---

### 3. 系统流程图的示例

下面以某企业的库房管理为例，说明系统流程图的使用。

某企业有一个库房，存放该厂生产需要的物品，库房中的各种物品的数量及各种物品库存量临界值等数据记录在库存文件上，当库房中物品数量有变化时，应更新库存文件。若某种物品的库存量少于库存临界值，则报告采购部门以便其订货，每天向采购部门送一份采购报告。

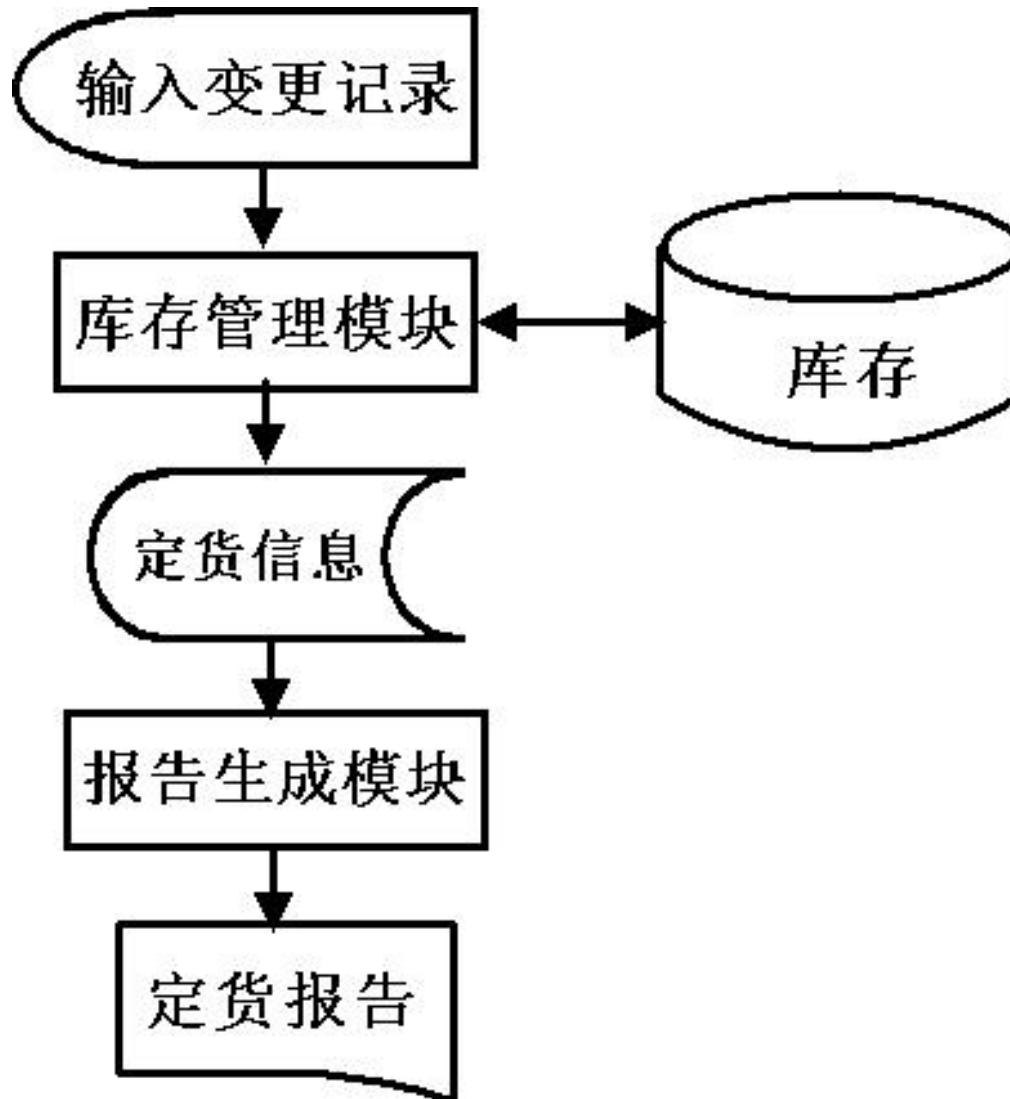


图4-2 库存管理系统的系统流程图

## 4.4 数据流图

---

数据流图（Data Flow Diagram，简称DFD）是结构化分析的最基本的工具。数据流图描述系统的分解，即描述系统由哪几部分组成，各部分之间有什么联系等。数据流图描述的是系统的逻辑模型，图中没有任何具体的物理元素，只是描绘信息在系统中的流动和处理情况。

---

因为数据流图是逻辑系统的图形表示，即使是非计算机专业的人员也能理解，所以是极好的通信工具。它以图形的方式描绘数据在系统中流动和处理的过程。由于它只反映系统必须完成的逻辑功能，所以它是一种功能模型。

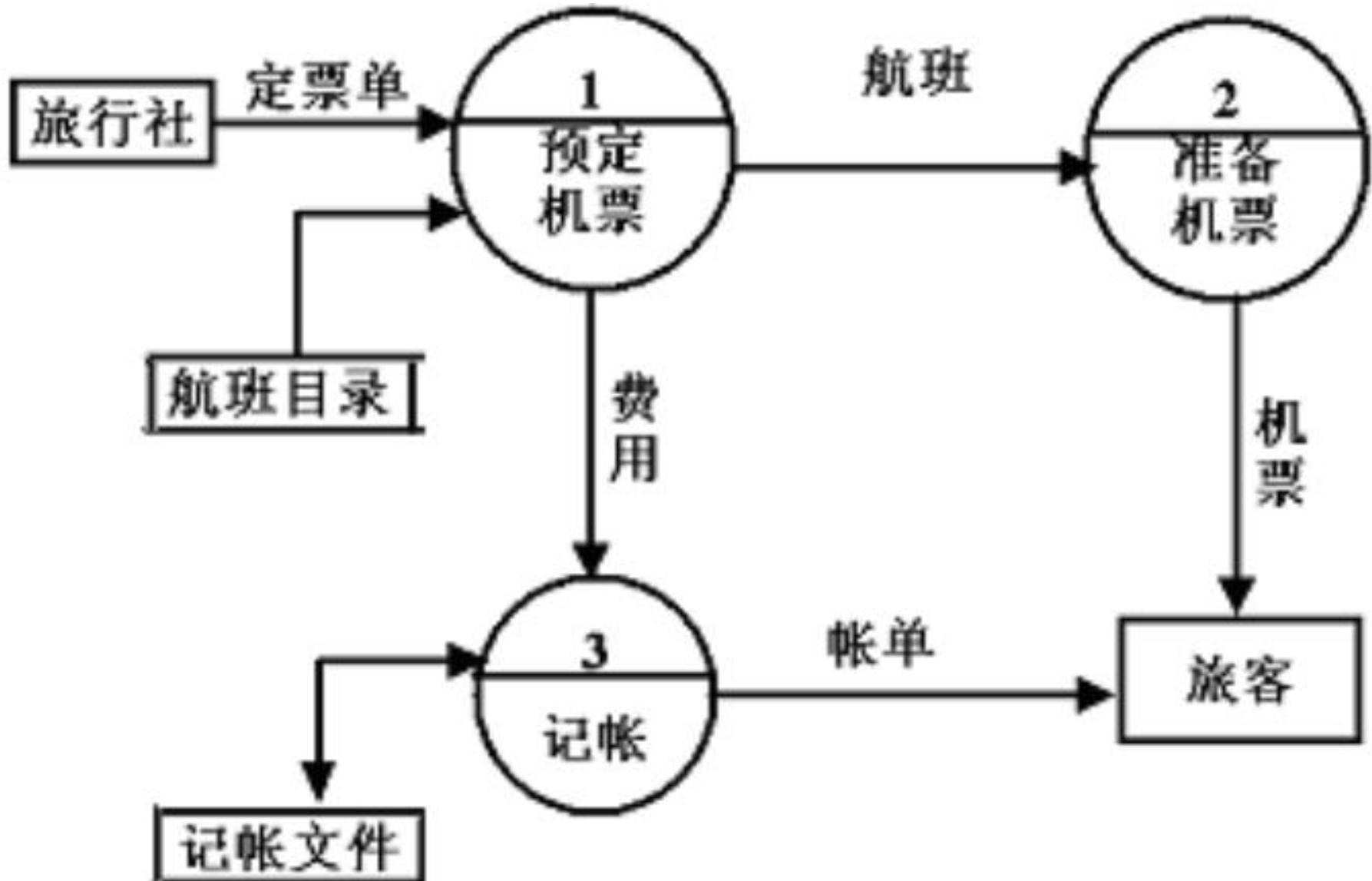


图4-3 飞机票预订系统

## 4.4.1 基本图形符号

---

数据流图要应用一些符号，有些表示的意义相同但是符号不一样。归纳起来数据流图只有4种基本符号元素：数据流（Data Flow）、数据处理（Process）、数据存储（Data Store）和外部实体（External Entity）。

数据流图有以下4种基本图形符号：

→：箭头，表示数据流。

○：圆或椭圆，表示加工。

=：双杠，表示数据存储。

□：方框，表示数据的源点或终点。

数据流图的同一种意义的表示方法有多种，如图4-4所示给出了三种常见的符号表示。在数据流图中由于用圆圈表示数据处理，整幅图看起来就好像有许多水泡泡，所以也称泡泡图

（Bubble Chart）。数据流图的特点是符号简单，使用方便，可以不考虑布局。图4-4是一个泡泡图实例。

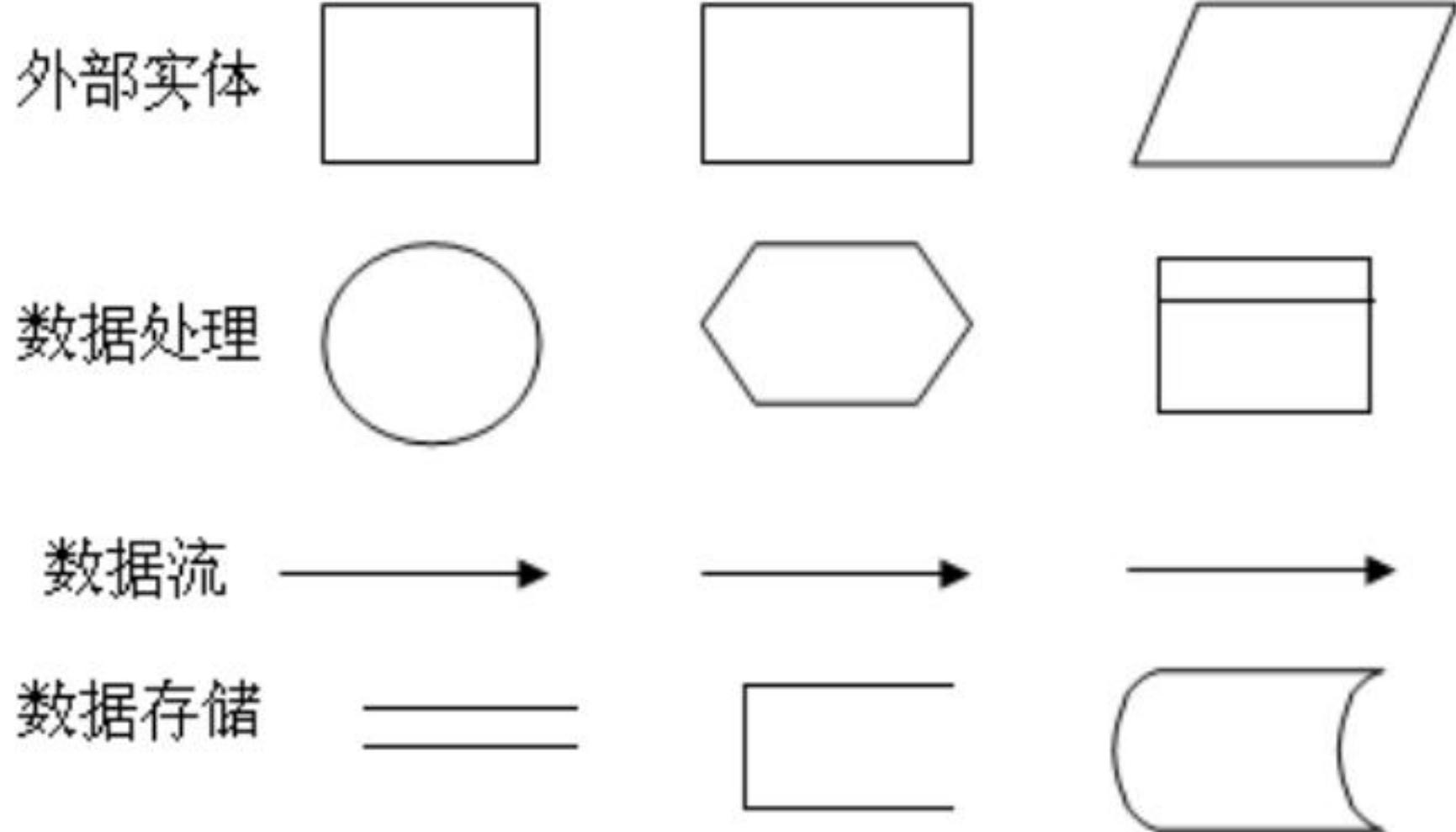


图4-4 数据流基本元素的三种不同表示

---

## 1.数据流

用箭头表示数据流，箭头方向表示数据流向，数据流名标在数据流线上面。数据流由一组数据项组成，但在数据流图中只有其名称，所以，应尽量准确地给数据流命名。数据流是数据在系统内传播的路径，由一组成分固定的数据项组成。

---

## 2. 加工

加工也称为数据处理，或称为变换，是对数据进行处理的单元。数据处理名称写在方框内。它对数据流进行某些操作或变换。每个加工也要有名字，通常是动词短语，简明地描述完成什么加工。在分层的数据流图中，加工还应编号。

---

### 3.数据存储

数据存储是由若干数据元素组成的，它为数据处理提供数据处理所需要的输入流或为数据处理的输出数据流提供储存“仓库”。

数据存储指暂时保存的数据，它可以是数据库文件或任何形式的数据组织。流向数据存储的数据流可理解为写入文件，或查询文件，从数据存储流出的数据可理解为从文件读数据或得到查询结果。

---

## 4.数据源点和终点

任何一个系统的边界定义后，就有系统内外之分，一个系统总会与系统外部的实体有联系。这种联系的重要形式就是数据。数据源点和终点是软件系统外部环境中的实体（包括人员、组织或其他软件系统），统称外部实体。它们是为了帮助理解系统界面而引入的，一般只出现在数据流图的顶层图中，表示了系统中数据的来源和去处。

## 4.4.2 画数据流图

---

用数据流图来表示系统中某一个层面的数据处理过程是很方便的。如果将一个复杂问题的全部用一幅数据流图来表示就困难了。为了表达较为复杂问题的数据处理过程，用一张数据流图是不够的。要按照问题的层次结构进行逐步分解，并以一套分层的数据流图反映这种结构关系。

---

## 1. 画系统的输入/输出

最初，把系统视为一个整体，看这个整体与外界的联系。分析有哪些内容是要通过外界获取的，就是系统的输入；有哪些是要向外界提供服务的，就是系统的输出。画系统的输入/输出即先画顶层数据流图。

机票预订系统的顶层图描述了机票预订系统与外界的简单关系。



图4-5 飞机票预订系统顶层图

---

## 2. 画系统内部

数据流图主要是用于描述系统内部的处理过程。有些内部处理过程比较简单，有些则相当复杂。描述系统内部即画下层数据流图。一般方法是将层号从0开始编号，采用自顶向下，由外向内的原则。

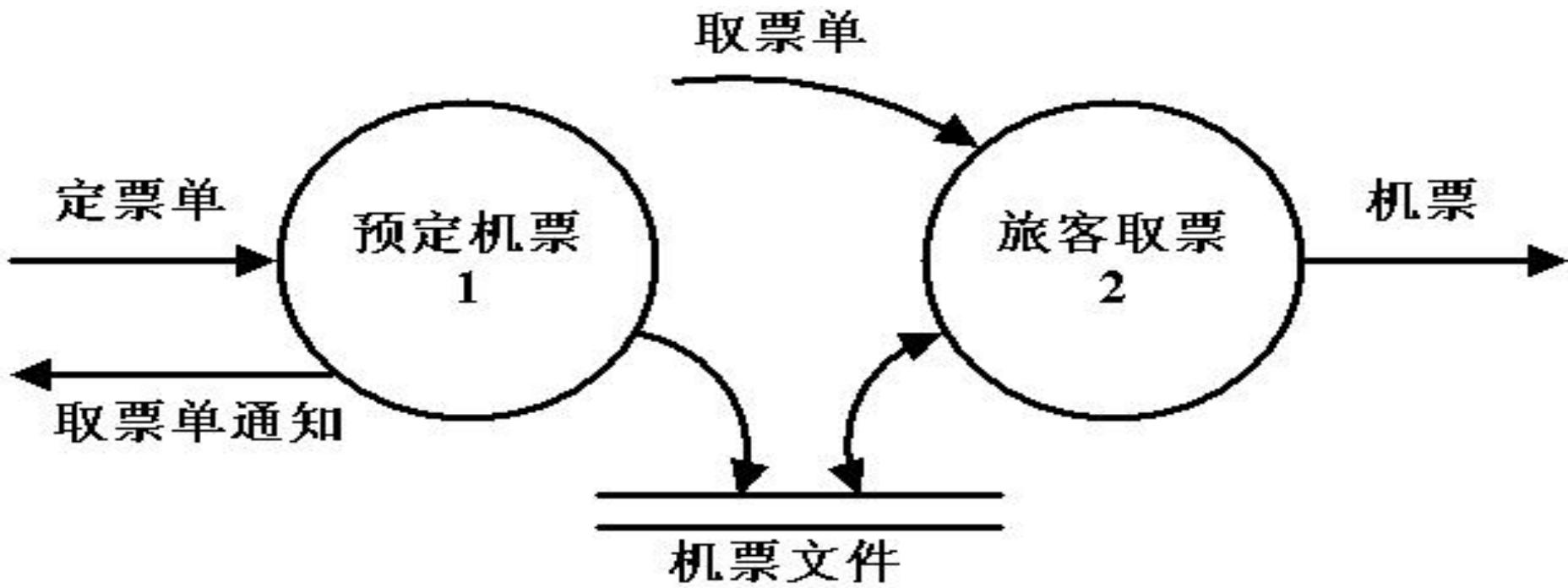


图4-6 飞机票预订系统0层图

---

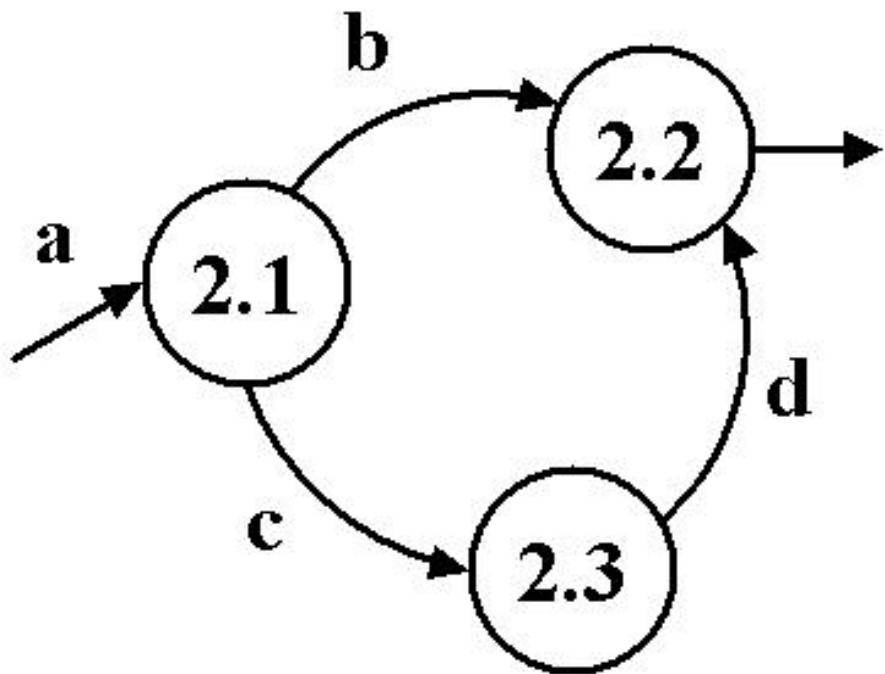
### 3. 注意事项

在软件的系统分析之前，系统的技术负责要制订一个系统的标准。其内容之一就是画数据流图的规范。画数据流图要注意以下几点。

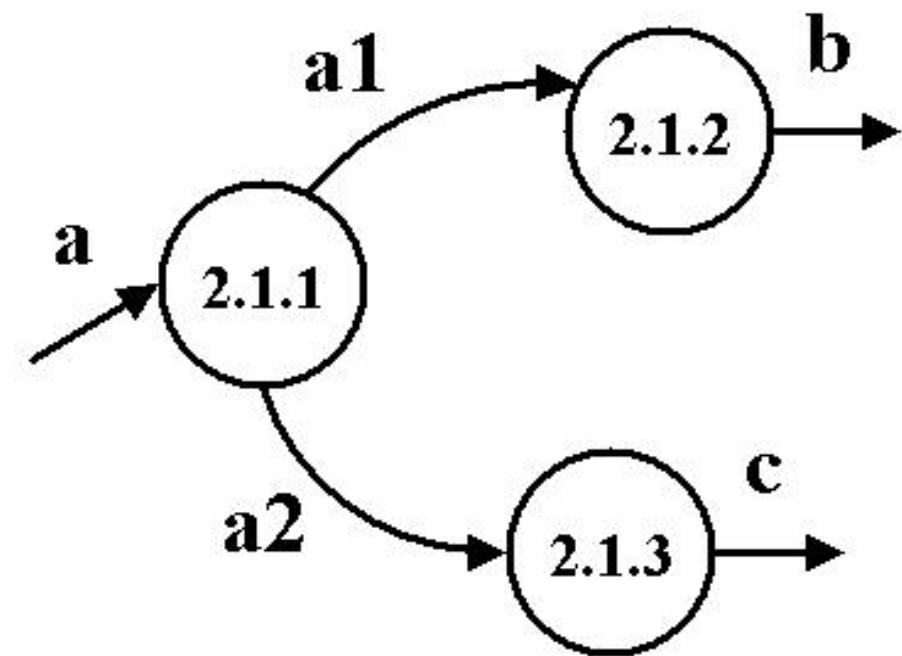
(1) 命名：在画数据流图中，不论数据流、数据存储还是加工的命名要合适，要易于理解其含义。数据流的名字代表整个数据流的内容，不仅仅是它的某些成分。命名时不能使用抽象含义的名字，比如“数据”、“信息”等等。加工名的命名也要反映其处理的功能，不能使用“处理”、“操作”这些笼统的词。

(2) 在画数据流图时要注意不是画控制流。数据流图反映的是系统“做什么”，不反映“如何做”，因此箭头上的数据流名称只能是名词类，整个图中不反映加工的执行顺序。

- (3) 每个加工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。
- (4) 加工点的编号：如果一张数据流图中的某个加工点要分解成另一张数据流图时，则上层图为父图，直接下层图为子图，父、子图上的所有加工都应编号。子图的编号是父图中相应加工的编号的扩充，子图上加工的编号的方法是由父图号、小数点及子图的局部号组成，如图4-7所示。



图a



图b

图4-7 父图与子图的编号

(5) 系统分析中要区别物流和数据流。数据流反映能用计算机处理的数据，并不是实物。因此在目标系统的数据流图上一般不要画物流，如机票预订系统中，人民币也在流动，但并未画出，因为交款是“人工”行为。

(6) 在数据流图表示系统的数据流向时，一般都要用到父图与子图来描述不同的层次。这时要注意父图与子图的平衡。子图的输入、输出数据流同父图相应加工的输入、输出数据流必须一致，即父图与子图的平衡。

在描述数据流时，为了考虑平衡可能要忽略一些枝节性的数据流（如出错处理）。因为父图与子图的平衡是分层数据流图中的重要性质，保证了数据流图的一致性。除了便于分析人员的阅读与理解外，在后面的变换与设计中也是非常重要的。

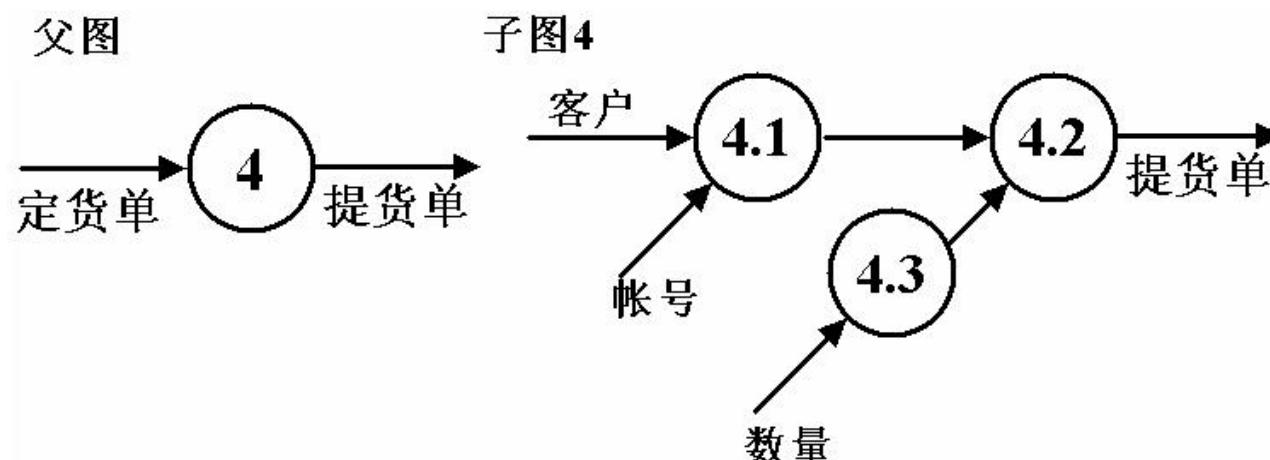


图4-8 父图与子图的平衡

(7) 在分层处理的过程中，当某层数据流图中的数据存储不是父图中相应加工的外部接口，而只是本图中某些加工之间的数据接口，则称这些数据存储为局部数据存储。如果是一个局部数据存储，只有当它作为某些加工的数据接口或某个加工特定的输入或输出时，就把它画出来，这样有助于实现信息隐蔽。

(8) 数据流图作为以后设计和与用户交流的基础，其易理解性极为重要。因此在分层中要注意合理分解，要把一个加工分解成几个功能相对独立的子加工，这样可以减少加工之间输入、输出数据流的数目，增加数据流图的可理解性。

---

不同的系统分析员在数据流图中使用的元素符号也有差别，这里给出了描述数据流图的一套基本符号：

- ：表示数据流，只能水平或垂直画。 ↵
- ：表示加工。 ↵
- ：表示数据存储。 ↵
- ：表示源点或终点。 ↵

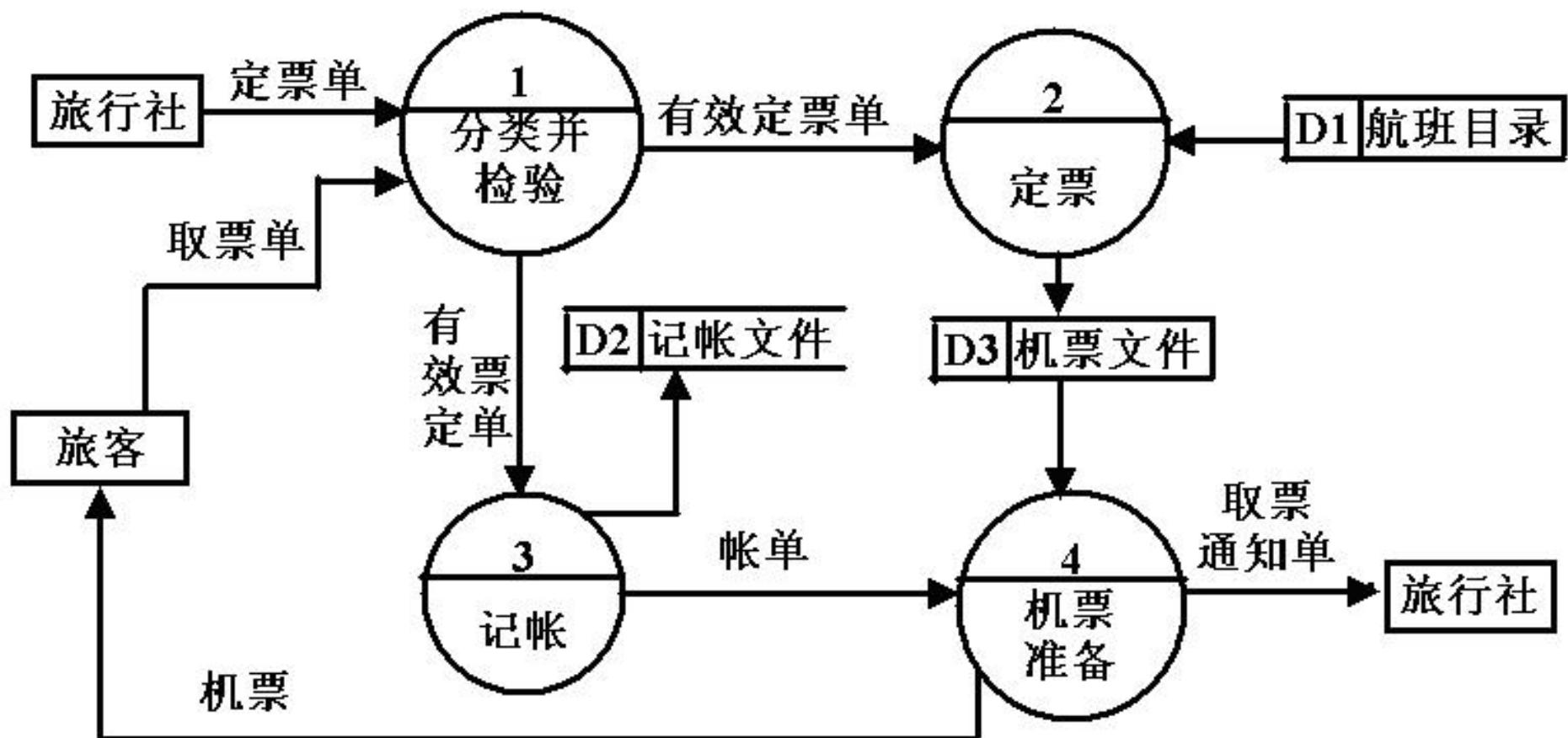


图4-9 与图4-3等价的DFD

## 4.5 数据字典

---

数据字典（Data Dictionary，简称DD）是关于数据的信息的集合，是对数据流图中包含的所有元素的定义的集合。它定义了数据流图中的数据的加工。它是数据流条目、数据存储条目、数据项条目和基本加工条目的汇集。

## 4.5.1 内容及格式

---

数据字典是为分析人员查找数据流图中有关名字的详细定义而服务的，因此也像普通字典一样，要把所有条目按一定的次序排列起来，以便查阅。数据流和数据字典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。在数据流图中的源点、终点不在系统之内，故一般不在数据字典中说明。数据字典有以下4类条目：数据流、数据项、数据存储及基本加工。其中数据项是组成数据流和数据存储的最小元素。

---

---

## 1. 数据流条目

要定义DFD中的数据流就要用数据流条目。定义方法通常列出该数据流的各组成数据项。在定义数据流或数据存储组成时，要用到一些符号。下面给出在数据字典的定义式中出现的符号。

- (1) =: 被定义为。
- (2) +: 与。例如,  $X=a+b$  表示x由a和 b组成。
- (3) [...]//: 或。例如,  $X=[a|b]$  表示x由a或b组成。
- (4) {...}: 重复。例如,  $X=\{a\}$  表示x由0个或多个a组成。
- (5)  $m\{...\}n$ 或 $\{...\}_m^n$ : 重复。例如:  $X=2\{a\}5$ 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。
- (6) (...): 可选。例如,  $x= (a)$  表示a可在x中出现, 也可不出现。

---

(7) “...”：基本数据元素。例如， $x = "a"$ ，表示 $x$ 是取值为字符 $a$ 的数据元素。

(8) “..”：连接符。例如， $x = 1..9$ ，表示 $x$ 可取1到9中任意一个值。

## 2. 数据项条目

数据流的组成成员是数据项，数据项条目是不可再分解的数据单位。其定义格式及举例如下。

数据项名称：货物编号。

别名：W-No, W-num, GW-No。

简述：公司内部所有货物的编号。

类型：字符串。

长度：10。

取值范围及含义：第1位：进口/国产。

第2~4位：类别。

第5~7位：规格。

第8~10位：产品编号。

### 3. 数据存储条目

与数据流条目一样。对存储数据的定义用数据存储条目。数据存储条目主要内容及举例如下。

数据存储名称：顾客记录。

别名：无。

简述：存放顾客的信息。

组成：姓名+编号+航班+目的地+身份证号码。

组织方式：索引文件，以姓名编号为关键字。

查询要求：要求能立即查询。

---

## 4. 加工条目

在DFD中有许多基本加工的处理逻辑。这些加工处理逻辑的说明是用加工条目。在各层都有加工处理逻辑。但是由于下层的加工是由上层的基本加工分解而来。因此，只要有了基本加工的说明，就可理解其他加工。

加工条目的主要内容及举例如下。

加工名：能否提供机票。

编号：1.2。

激发条件：接收到合格订票单时。

优先级：普通。

输入：合格订单。

输出：能提供机票、不能提供机票。

加工逻辑：根据库存记录。

IF 订单项目的数量 < 该项目库存量的临界值

THEN 提供机票处理

ELSE 此订单缺票，登录，待有票后再处理

ENDIF

---

## 4.5.2 数据字典的实现

---

数据字典除了概念和技术上的问题外，工作量是非常大的。由于是系统的一项基础工作，所以数据字典的实现因环境的不同而采用不同的实现方法。早期人们用手工建立数据字典，现在一般是利用计算机辅助建立并维护数据字典。步骤如下。

(1) 编制一个“字典生成与管理程序”，可以按规定的格式输入各类条目，能对字典条目增、删、改，能打印出各类查询报告和清单，能进行完整性、一致性检查等。

(2) 利用已有的数据库开发工具，针对数据字典建立一个数据库文件，可将数据流、数据项、数据存储和加工分别以矩阵表的形式来描述各个表项的内容，如数据流的矩阵表为：

编号	名称	来源	去向	流里	组成
... <sup>②</sup>					

---

然后使用开发工具建成数据文件，便于修改、查询，并可随时打印出来。另外，有的DBMS本身包含一个数据字典子系统，建库时能自动生成数据字典。

目前一些软件开发小组都习惯于用计算机中小的数据库软件作为数据字典的实现工具。最方便的可以用微软的Access和Excel。计算机辅助开发数据字典比手工建立数据字典有更多的优点，能保证数据的一致性和完整性，使用也方便。

---

# 4.6 成绩管理系统结构化分析

## 4.6.2 业务需求

### 1 系统用户

1) 名字解释

2) 系统用户结构



---

## 2 业务描述

1) 超级管理员:

2) 普通管理员:

3) 普通用户:

3 业务流程:

1) 超级管理员的基本业务流程如图4-11所示

2) 普通管理员的基本业务流程如图4-12所示

---

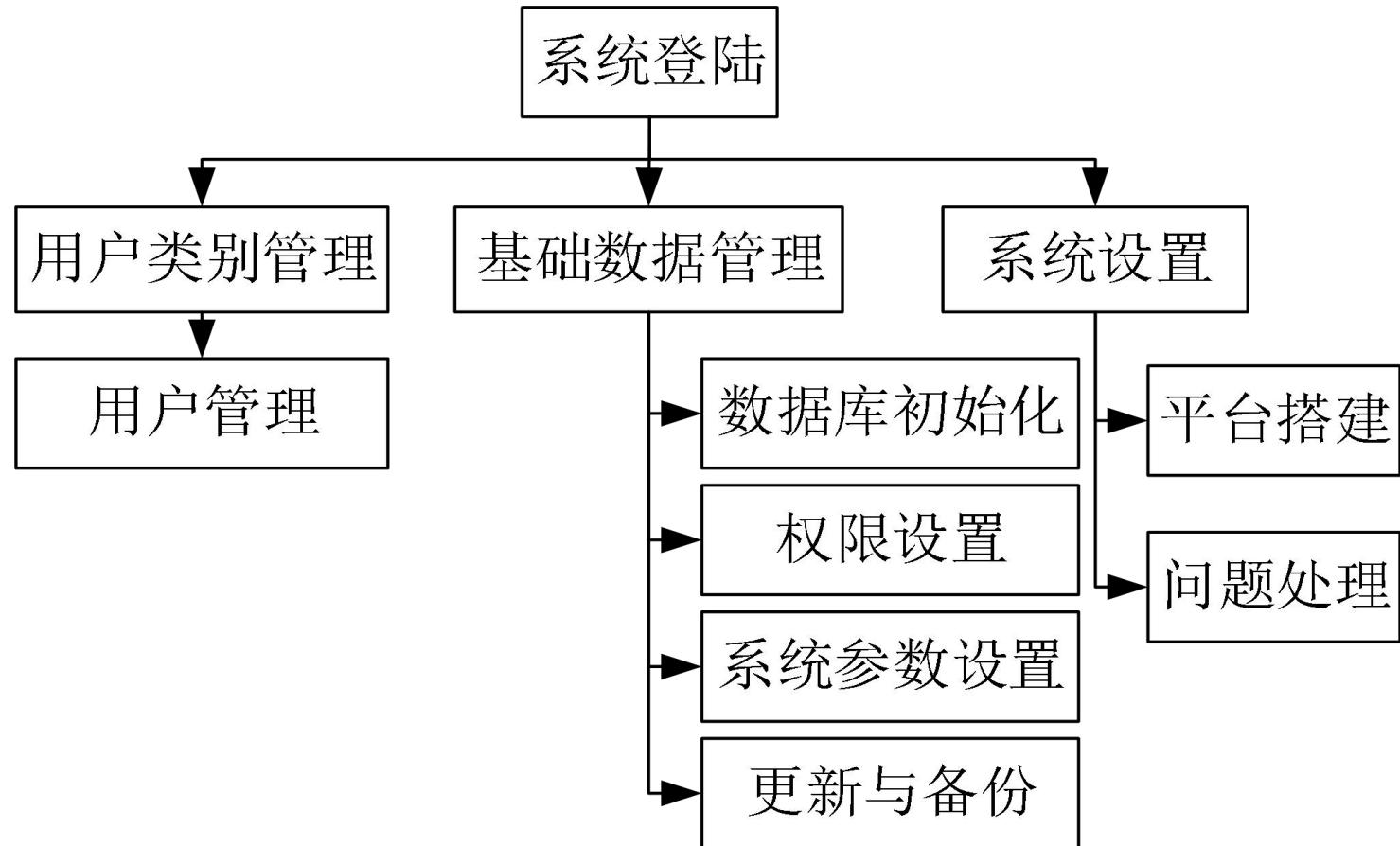


图4-11 超级管理员的基本业务流程图

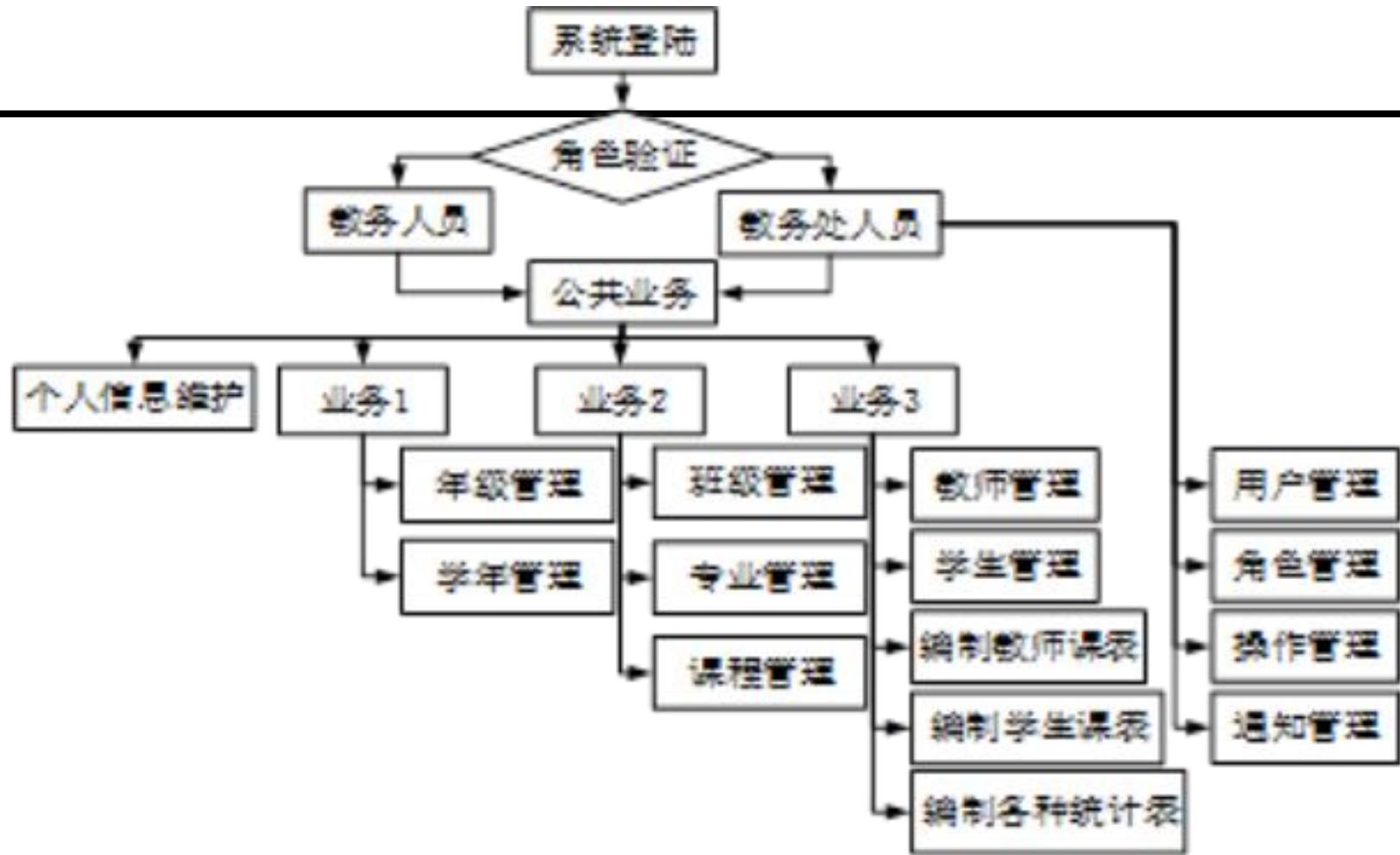
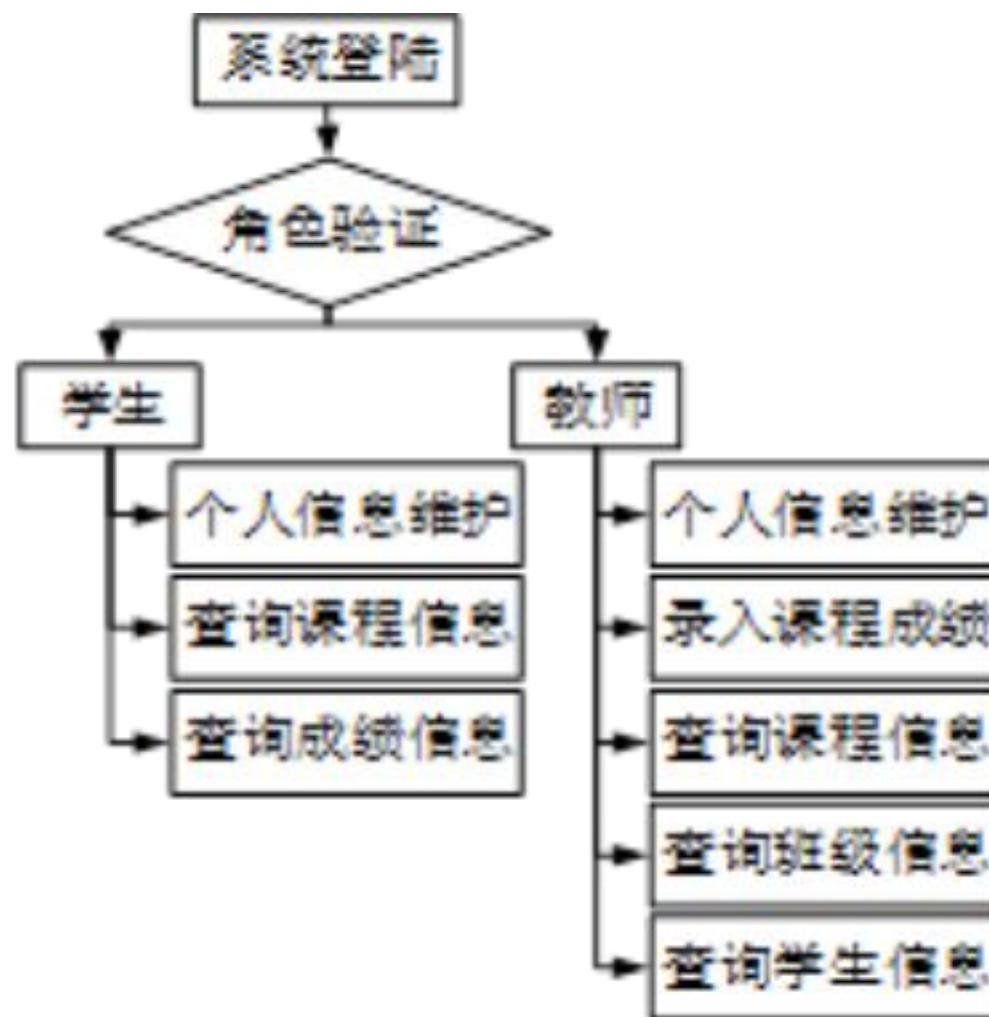


图4-12 普通管理员的基本业务流程图

### 3) 通用用户的基本业务流程如图4-13所示

---



---

### 4.6.3 需求分析

一、系统目标

二、业务分析

1、系统外部实体

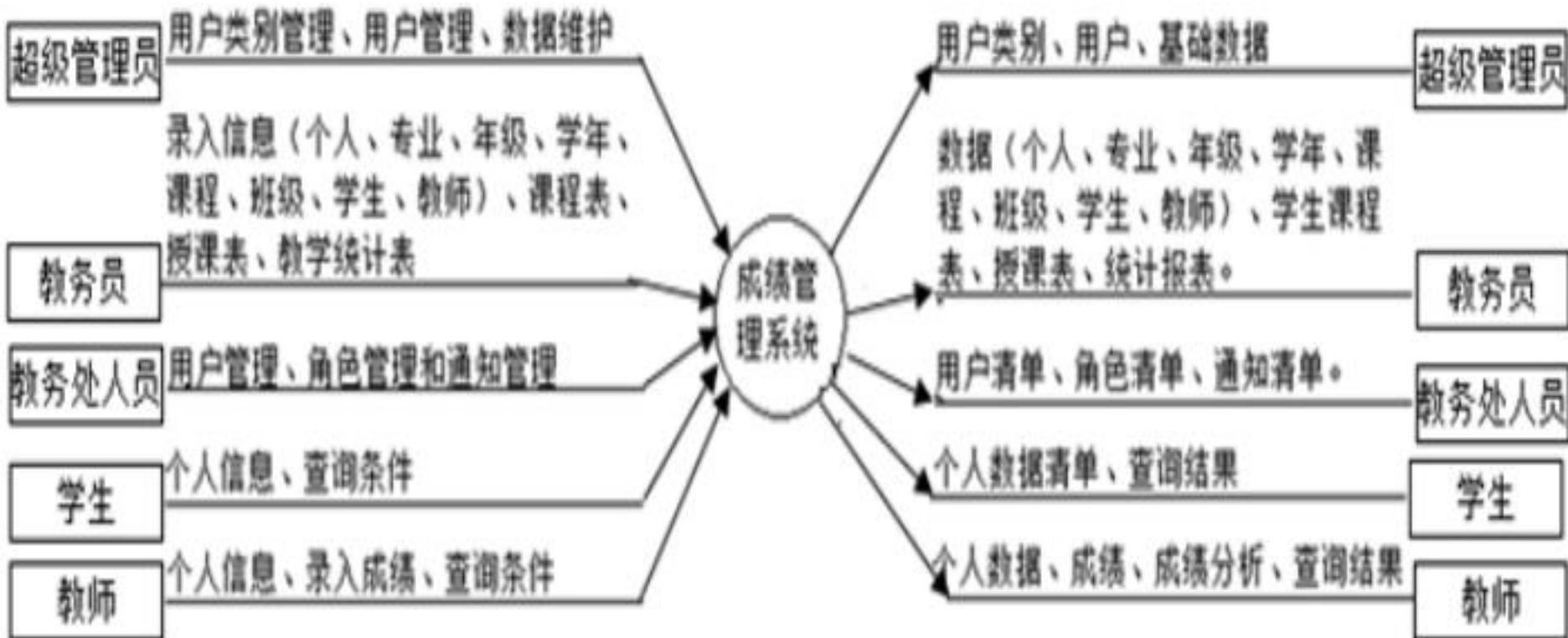
2、系统输入与输出

1) 系统输入

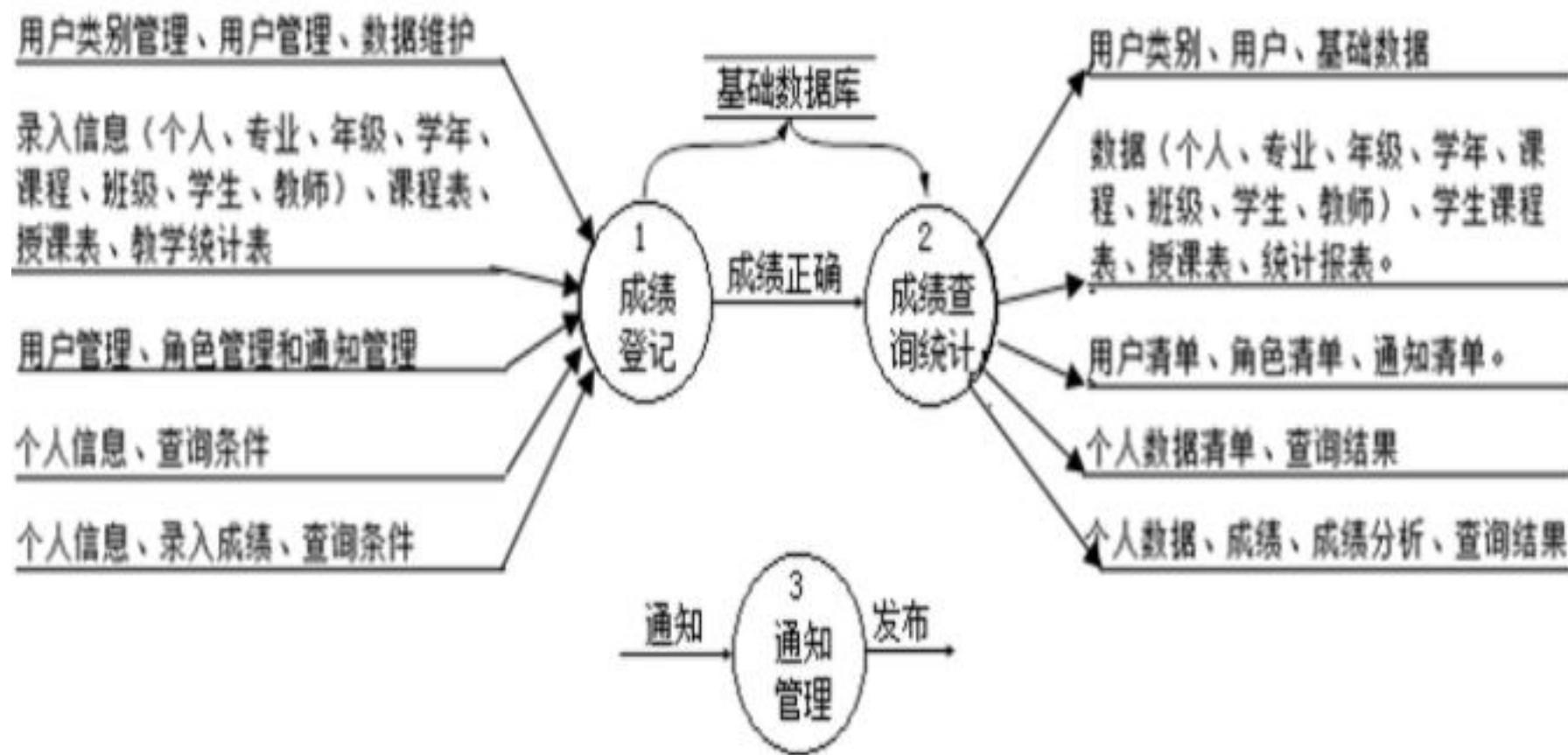
2) 系统输出

# 三、分析结果

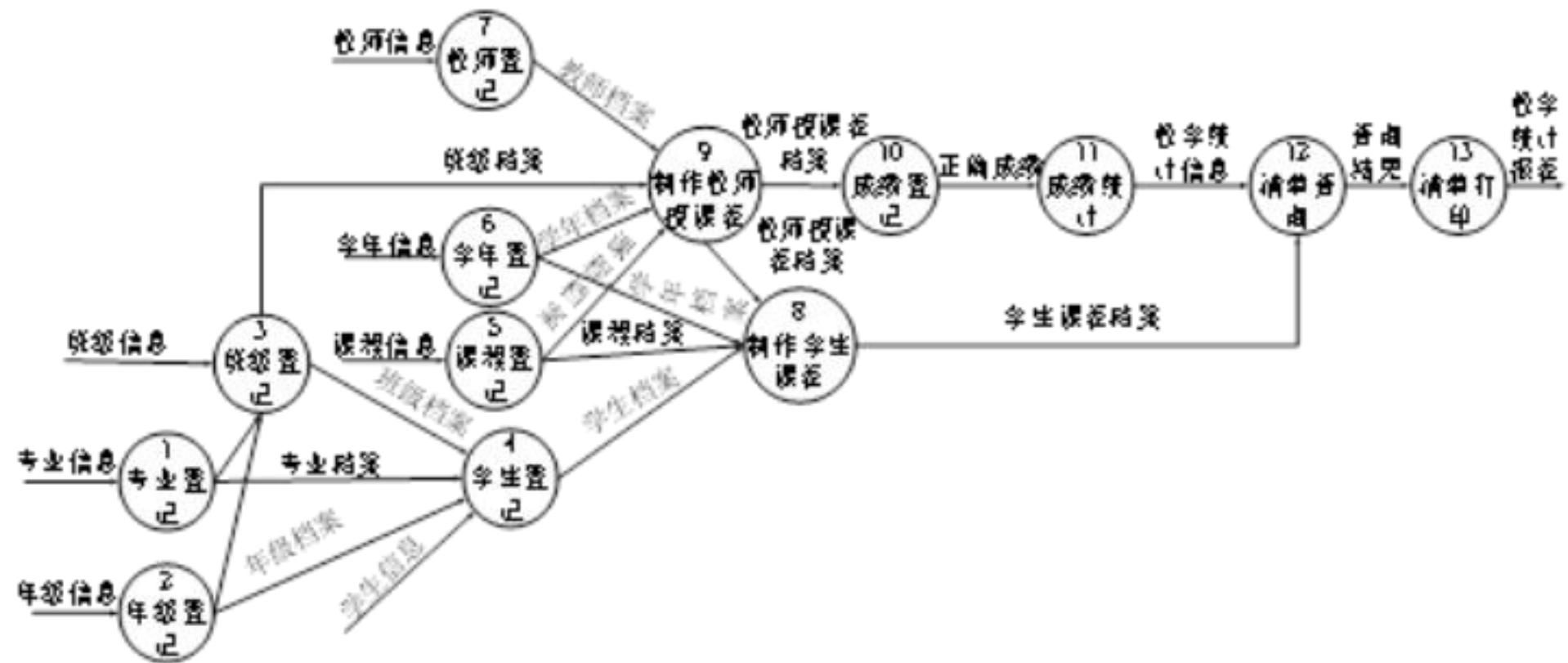
## 1、顶层数据流图

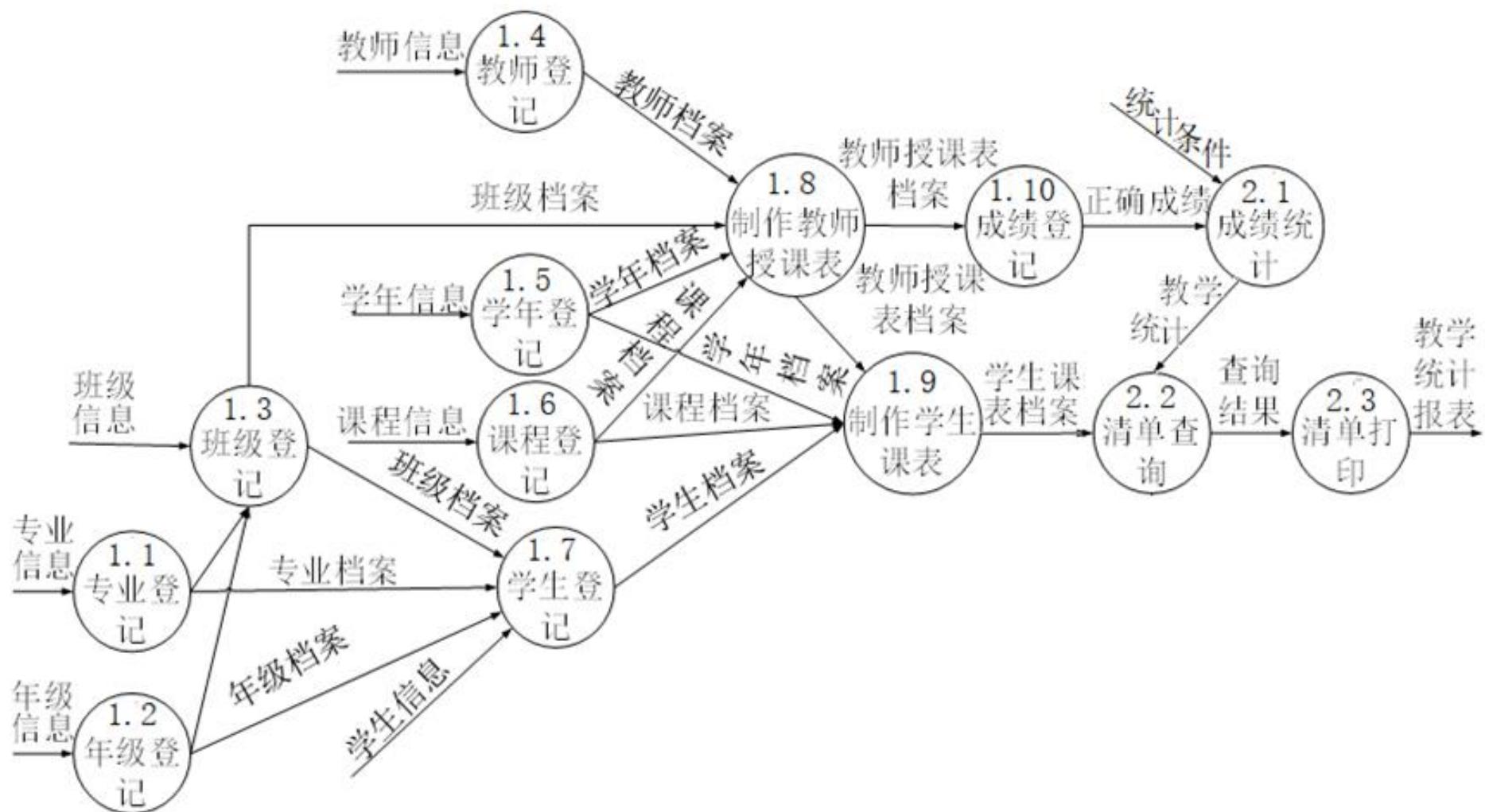


## 2、0层数据流图



### 3、1层数据流图





# 5、数据字典

---

1)、数据流条目

(1) 数据流“用户信息”：

编 号： DF13-00

数据流： 用户信息

说 明： 系统用户输入的用户详细信息

数据流来源： 系统用户输入

数据流去向： 录入用户。

组成： { 用户ID+用户登陆密码+用户类别 }

流量： 最小1条/次； 最大1个班级的人数/次

(2) 数据流“用户档案”：

---

## 2、数据项条目

### (1) 数据项“用户ID”

编 号： D13-001

数据项： 用户ID

含义说明： 唯一标识每个用户

别名： 用户编号

类型： 字符型

长度： 8（由学号的长度决定）

取值范围： 00000000至99999999（或学号的长度）

取值含义： 可用学号作为用户ID， 学号的每一段代表不同的含义。其他数据项“用户登陆密码”、数据项“用户类别”、数据项“启用标志”

### 3、加工条目

---

#### (1) 加工“录入用户”

编号: P13-01

加工: 录入用户

说明: 向目标系统录入用户信息，并将用户信息存储于用户表中

输入: 用户信息(或学生档案或教师档案)、用户类别档案

输出: 用户档案

处理: 对输入信息进行识别，由学生或教师档案导入；用户类别档案由用户类别表中导入，每一用户只能是一个类别。将正确的用户信息存储于用户表中，输出用户档案，用户档案作为修改用户和删除用户的输入数据。用户密码和用户类别不可为空，“被删除”类别输入时不可以选择，新输入的用户系统默认启用标志设置为“启用状态”。其他加工（略）。

---

## 四、系统功能与能力

成绩登记、成绩查询统计和通知管理。目标系统具备的能力包括：数据能够录入、修改、删除、查询和打印；正常运行和维护。

## 五、系统非功能性需求

---

- 1、安全、安全保密性需求：用户密码加密等级为普密级和服务器数据定期备份。
  - 2、接口需求：目标系统能够从外部导入Word、Excel文档或导出Word、Excel文档。
  - 3、性能需求：数据准确；运行稳定；系统能够支持高峰期10万个虚拟用户同时访问；系统兼容性和可扩展性较高，易移植。
  - 4、运行与维护需求：硬件环境与软件环境。
  - 5、设计约束和合格性需求：操作界面人性化设计与输入数据简化处理。
-

# 小结

---

本章主要介绍需求分析的特点、原则、任务和方法。在需求分析方法中说明了功能分解方法、结构化分析方法、信息建模方法和面向对象的分析。为了便于在实际系统分析中处理数据关系。

# 小结

---

软件体系结构的设计是软件的关键。为此，还介绍的是从需求到软件体系结构的方法。主要介绍RTRSM的基本元素和RTRSM到体系结构的转换步骤。也说明了基于软件体系结构开发方法和 软件体系结构求精方法。这样做的目的是为了在软件结构设计时利用用户需求的分析可以求出体系结构。但是这些方法目前还在进一步深入研究中。

---

# 谢谢