

第3章 软件需求获取与结构化分析方法

- 需求获取与需求分析阶段的任务
- 结构化分析方法
- 系统需求规格说明
- 需求评审
- 需求管理

示例

- 基于欧洲航天局 (ESA) 的太空火箭 Ariane-5 中的嵌入式软件。1996 年 6 月 4 日，Ariane-5 在其首飞中发射，并完美运行了大约 40 秒。然后，它开始偏离航线。在 Ariane 地面控制员的指示下，火箭被遥控摧毁。这枚火箭被毁，不仅火箭本身受损，而且火箭上搭载的四颗卫星也受损；这场灾难的总损失高达 5 亿美元（Newsbytes 主页 1996 年；Lions 等人 1996 年）。
- 使用 16 位无符号整数，我们可以存储 0 到 65535 之间的任意值。而如果把首位用来存放符号（正 / 负），那么 16 位有符号整数就能涵盖从 -32768 到 +32768 的任意值（实际可用数位只有 15 位）。任何超出这个范围的值都无法正常使用。
- 另一方面，浮点数的存储规则略有不同，强调的是在相同的位数中覆盖更大范围的数字。例如，即使是 16 位（双精度）浮点数，也能存储从 $-1.8e+308$ 到 $-2.2e-308$ 之间的大量值。可见，要把其中的某个值转换成 16 位有符号整数，则很可能会超出后者的支持范围。那如果是 64 位浮点数呢？结果只会更糟。
- 一旦这种不可避免的事态成真，会有怎样的后果？在使用 16 位有符号整数时，从浮点数到整数的转换会引发我们熟知的整数溢出。现在只剩最后一个问题了：整数溢出，对于火箭发射意味着什么？
- 制导系统会读取火箭的水平速度数据（64 位浮点数），并尝试将其转换为 16 位整数以发送至主计算机。但转换未能成功。

示例

- 一旦这种不可避免的事态成真，会有怎样的后果？在使用 **16 位有符号整数** 时，从浮点数到整数的转换会引发我们熟知的整数溢出。现在只剩最后一个问题了：整数溢出，对于火箭发射意味着什么？
- 制导系统会读取火箭的水平速度数据（**64 位浮点数**），并尝试将其转换为 **16 位整数** 以发送至主计算机。但转换未能成功。很明显，因为读数大于 **16 位整数** 所能表示的最大值，所以转换失败。一般来讲，设计良好的系统会内置一个程序来处理溢出错误，并向主计算机发送一条合理的消息。但阿丽亚娜并不是这样……制导系统会持续发送错误消息，于是主计算机不但接收不到正确的水平速度值，制导系统那边还被立即关闭了。

通往设计和施工的桥梁

- 1994 年，Standish Group 对 350 多家公司进行了调查，调查了它们 8000 多个软件项目，以了解它们的进展情况。结果令人担忧。31% 的软件项目在完成之前就被取消了。此外，在大型公司中，只有 9% 的项目按时交付并按预算完成，而在小型公司中，只有 16% 的项目符合这些标准 (Standish 1994)。
- 为了了解原因，Standish (1995) 要求调查对象解释项目失败的原因。
 - 据报道，主要因素包括
 - 需求不完整 (13.1%)
 - 缺乏用户参与 (12.4%)
 - 缺乏资源 (10.6%)
 - 不切实际的期望 (9.9%)
 - 缺乏高管支持 (9.3%)
 - 需求和规范不断变化 (8.7%)
 - 缺乏规划 (8.1%)
 - 不再需要系统 (7.5%)

通往设计和施工的桥梁



客户如此描述需求



项目经理如此理解



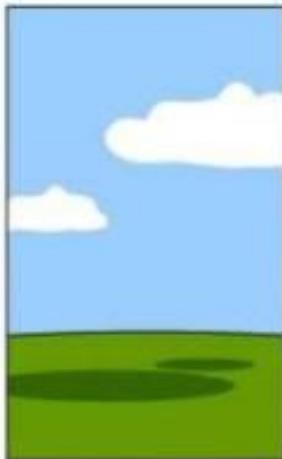
分析员如此设计



程序员如此编码



商业顾问如此诠释



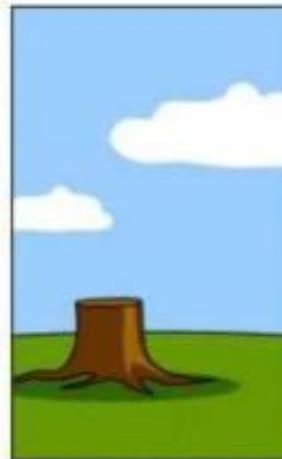
项目文档如此编写



安装程序如此“简洁”



客户投资如此巨大



技术支持如此肤浅



解密：
实际需求—原来如此

通往设计和施工的桥梁



3.1 需求获取与需求分析阶段的任务

- 需求获取的任务和原则
- 需求获取的过程
- 软件需求分析阶段的任务

需求获取的任务和原则

- **需求获取的主要任务是与客户或用户沟通，了解系统或产品的目标是什么？客户或用户想要实现什么？系统和产品如何满足业务的要求，最终系统或产品如何用于日常工作？**
- **获取并理解用户的需求是软件工程师所面对的最困难的任务之一。**

需求获取的任务和原则

- 导出需求变得如此困难的原因归为以下几个方面的问题：
 - **系统的目标或范围问题；**
 - **需求不准确性问题；**
 - **需求的易变问题；**

需求获取除了需要有专业的系统分析师，还需要通过有效的客户 / 开发者的合作才能成功。

需求获取的任务和原则

1. 需求获取的任务

- (1) 发现和分析问题，并分析问题的原因/结果关系。**
- (2) 与用户进行各种方式的交流，并使用调查研究方法收集信息。**
- (3) 按照三个成分观察问题的不同侧面：即数据、过程和接口。**
- (4) 将获取的需求文档化，形式有用例、决策表、需求表等。**

需求获取的任务和原则

2. 需求获取应遵循的原则

- (1) 深入浅出的原则。**就是说，需求获取要尽可能全面、细致。获取的需求是个全集，目标系统真正实现的是个子集。
- (2) 以流程为主线的原则。**在与用户交流的过程中，应该用流程将所有的内容串起来。如信息、组织结构、处理规则等。这样便于交流沟通。流程的描述既有宏观描述，也有微观描述。

需求获取的过程

1. 开发高层的业务模型
2. 定义项目范围和高层需求
3. 识别用户类和用户代表

系统的不同用户之间在很多方面存在差异，例如：

- (1) 使用产品的频率；**
- (2) 用户在应用领域的经验和使用计算机系统的技能；**
- (3) 所用到的产品功能；**
- (4) 为支持业务过程所进行的工作；**
- (5) 访问权限和安全级别**

需求获取的过程

4. 获取具体的需求

确定了项目范围和高层需求，并确定了用户类及用户代表后，就需要获取更具体、完整和详细的需求。具体需求的来源可以来自以下几种典型的途径。

- (1) 与用户进行交流。**
- (2) 现有产品或竞争产品的描述文档。**
- (3) 系统需求规格说明。**
- (4) 当前系统的问题报告和改进要求。**
- (5) 市场调查和用户问卷调查。**
- (6) 观察用户如何工作。**

需求获取的过程

5. 确定目标系统的业务 workflows

具体到当前待开发的应用系统，确定系统的业务 workflows 和主要的业务规则，采取需求调研的方法获取所需的信息。例如，针对信息系统的需求调研方法如下：

(1) 调研用户的组织结构、岗位设置、职责定义，从功能上区分有多少个子系统，划分系统的大致范围，明确系统的目标。

(2) 调研每个子系统的工作流程、功能与处理规则，收集原始信息资料，用数据流来表示物流、资金流、信息流三者的关系。

(3) 对调研内容事先准备，针对不同管理层次的用户询问不同的问题，列出问题清单。将操作层、管理层、决策层的需求既联系又区分开来，形成一个需求的层次。

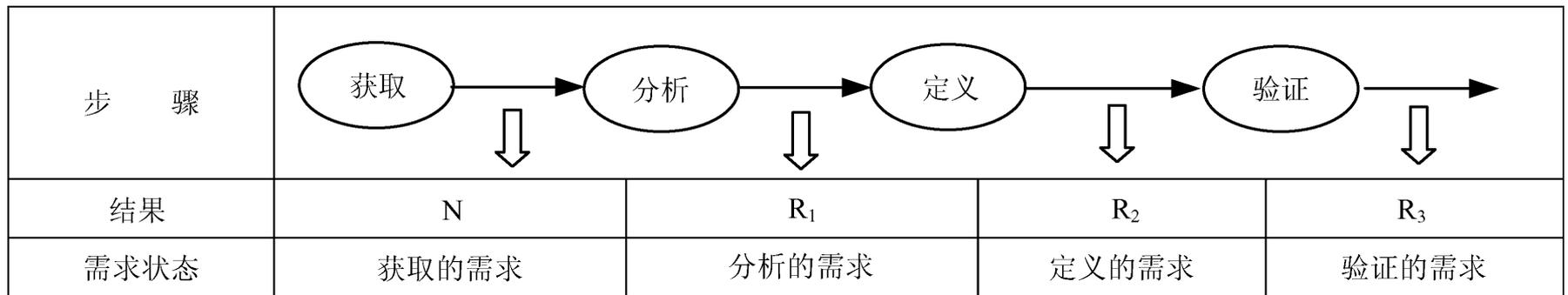
需求获取的过程

6. 需求整理与总结

- **必须对上面步骤取得的需求资料进行整理和总结，确定对软件系统的综合要求，即软件的需求。**
- **并提出这些需求实现条件，以及需求应达到的标准。**
- **这些需求包括功能需求、性能需求、环境需求、可靠性需求、安全保密要求、用户界面需求、资源使用需求、软件成本消耗与开发进度需求等。**

软件需求分析阶段的任务

- 可以把软件需求分析阶段的工作分为4个步骤，即**获取需求**、**分析需求**、**定义需求**和**验证需求**，如图所示。



软件需求分析阶段的工作步骤

软件需求分析阶段的任务

1. 需求获取

通过启发、引导从客户（或用户）那里得到的原始需求是他们的业务要求（needs），简称为N。

这是分析之前获取的需求，其中可能存在一些实际问题，这些问题只有通过分析才能得到解决，直接把获取的需求作为软件设计阶段的依据将会导致严重的后果。

软件需求分析阶段的任务

2. 需求分析

认真研究获取的需求，必须考虑以下几方面：

- (1) **完整性**：每项获取的需求都应给出清楚描述，使得软件开发工作能够取得设计和实现该功能所需要的全部必要信息。
- (2) **正确性**：获取的每项需求必须是准确无误的，并且需求描述无歧义性。
- (3) **合理性**：各项需求之间、软件需求与系统需求之间应是协调一致的，不应存在矛盾和冲突。

软件需求分析阶段的任务

2. 需求分析

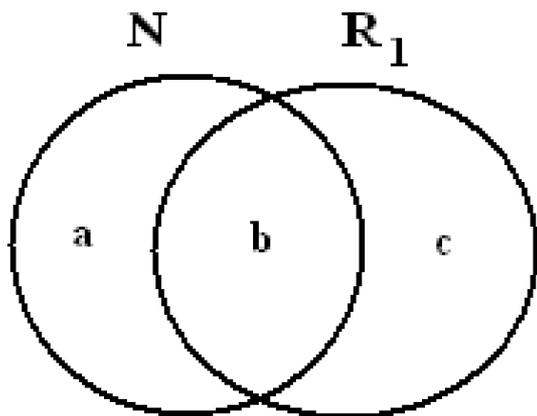
(4) **可行性**：包括技术可行性、经济可行性、社会可行性。

(5) **充分性**：获取的需求是否全面、周到。

软件需求分析阶段的任务

2. 需求分析

由于分析的过程会对获取的需求做部分调整，也即从获取的需求 N 中去掉了一些 a ，又补充了一些 c ，从而得到的是分析的需求 R_1 ($b+c$)。



N: 获取的需求

R_1 : 分析的需求

a: 经分析去掉的部分

b: 经分析保留的部分

c: 经分析补充的部分

软件需求分析阶段的任务

3. 需求定义

将已经过分析的需求清晰、全面、系统、准确地描述成为正式的文档，这一步定义需求的工作就是编写需求规格说明。

软件需求分析阶段的任务

4. 需求验证

为了确保已定义的需求（需求规格说明）准确无误，并能为客户（或用户）理解和接受，需要对其进行严格的评审。

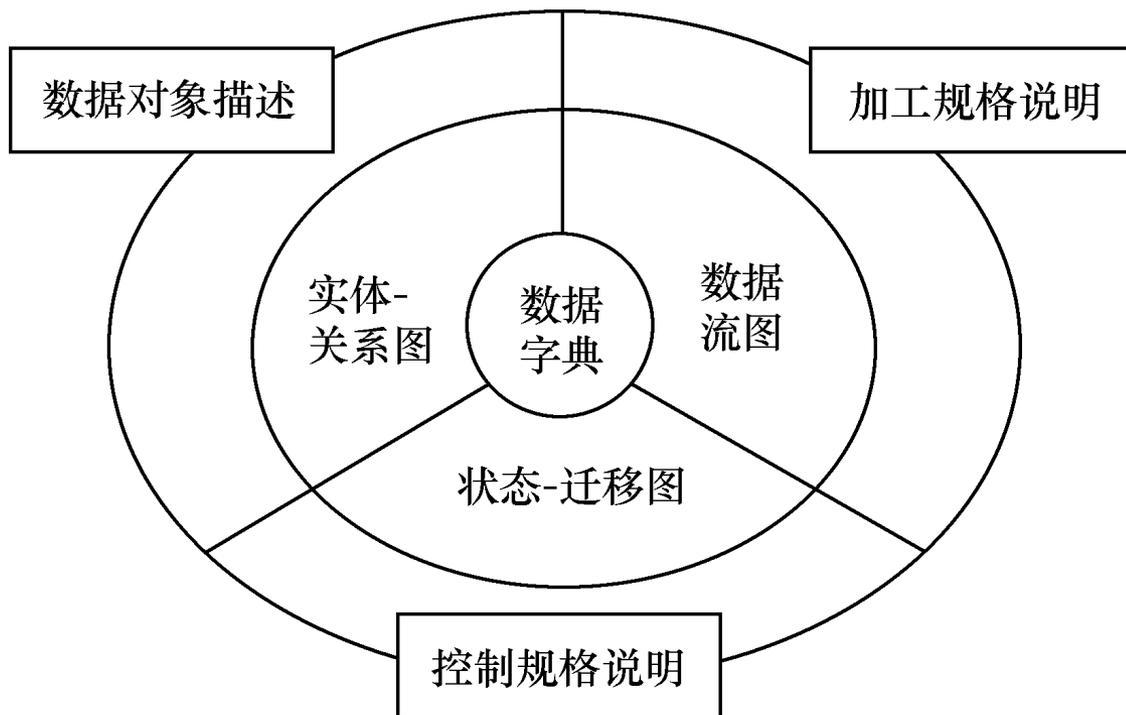
3.2 结构化分析方法

- **结构化分析方法**
 - **传统的分析建模方法称为结构化分析 (structured analysis, SA) 方法。**
 - **最有代表性的是一种面向数据流进行需求分析的方法，最初于20世纪70年代由D.Ross提出，后来又经过扩充，形成了今天的结构化分析方法的框架。**

3.2 结构化分析方法

- 结构化分析模型

结构化分析方法是一种建模技术，它建立的分析模型如图所示。



3.2.1 功能建模

- 概念

功能建模的思想就是用抽象模型的概念，按照软件内部数据传递、变换的关系，自顶向下逐层分解，直到找到满足功能要求的所有可实现的软件为止。功能模型用数据流图来描述。

3.2.1 功能建模

• 数据流图的基本图形符号



或

加工。对输入数据进行变换以产生输出数据，其中要注明加工的名字。



或

外部实体，即数据输入源（Source）或数据输出汇点（Sink）。其中要注明数据源或数据汇点的名字。



或

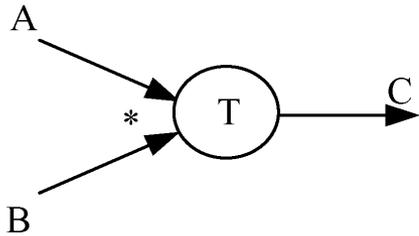
数据存储。要用名词或名词性短语为数据存储命名。



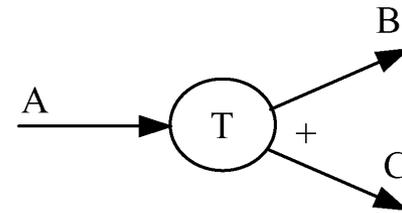
数据流。描述被加工数据及传递方向。箭头旁边要注明数据流的名字，可用名词或名词性短语命名。

3.2.1 功能建模

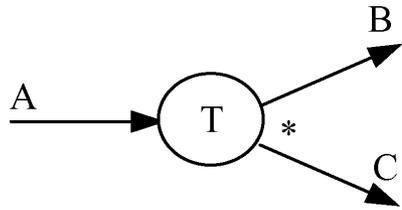
• 多个数据流之间的关系



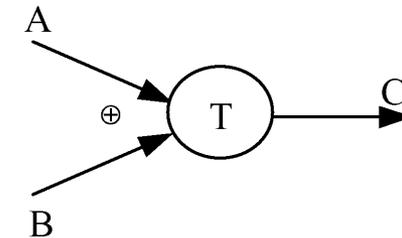
数据 A 和 B 同时输入
才能变换成 C



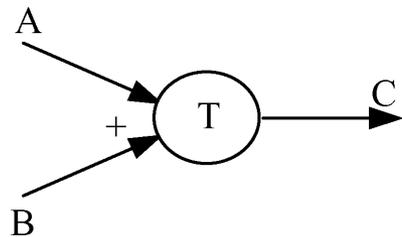
数据 A 变换成 B 或 C，
或 B 和 C，其中，“+”
号可以省略



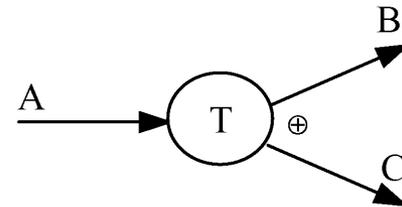
数据 A 变换成 B 和 C



数据 A 或数据 B（但不
能 A、B 同时）输入时变
换成 C



数据 A 或 B，或 A 和 B
同时输入变换成 C，其中，
“+”号可以省略



数据 A 变换成 B 或 C，
但不能变换成 B 和 C

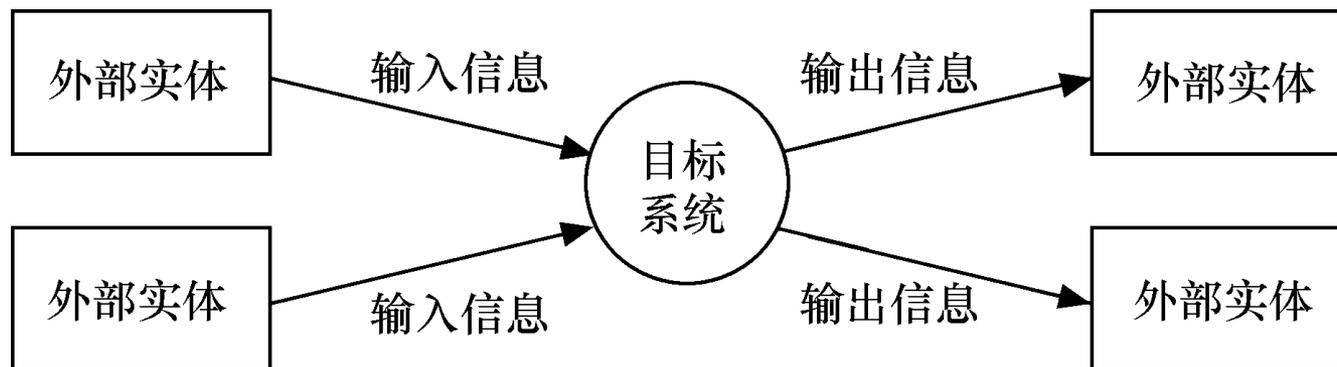
3.2.1 功能建模

- 环境图

- 环境图（context diagram）也称为**顶层数据流图**（或**0层数据流图**），它仅包括一个数据处理过程，也就是要开发的目标系统。
- 环境图的作用是确定系统在其环境中的位置，通过确定系统的输入和输出与外部实体的关系确定其边界。

3.2.1 功能建模

- 典型的环境图



3.2.1 功能建模

- 招生系统需求描述
 - 学校首先公布招生条件，考生根据自己的条件报名，之后系统进行资格审查，并给出资格审查信息；
 - 对于资格审查合格的考生可以参加答卷，系统根据学校提供的试题及答案进行自动判卷，并给出分数及答题信息，供考生查询；
 - 最后系统根据学校的录取分数线进行录取，并将录取信息发送给考生。

3.2.1 功能建模

- 招生系统的环境图

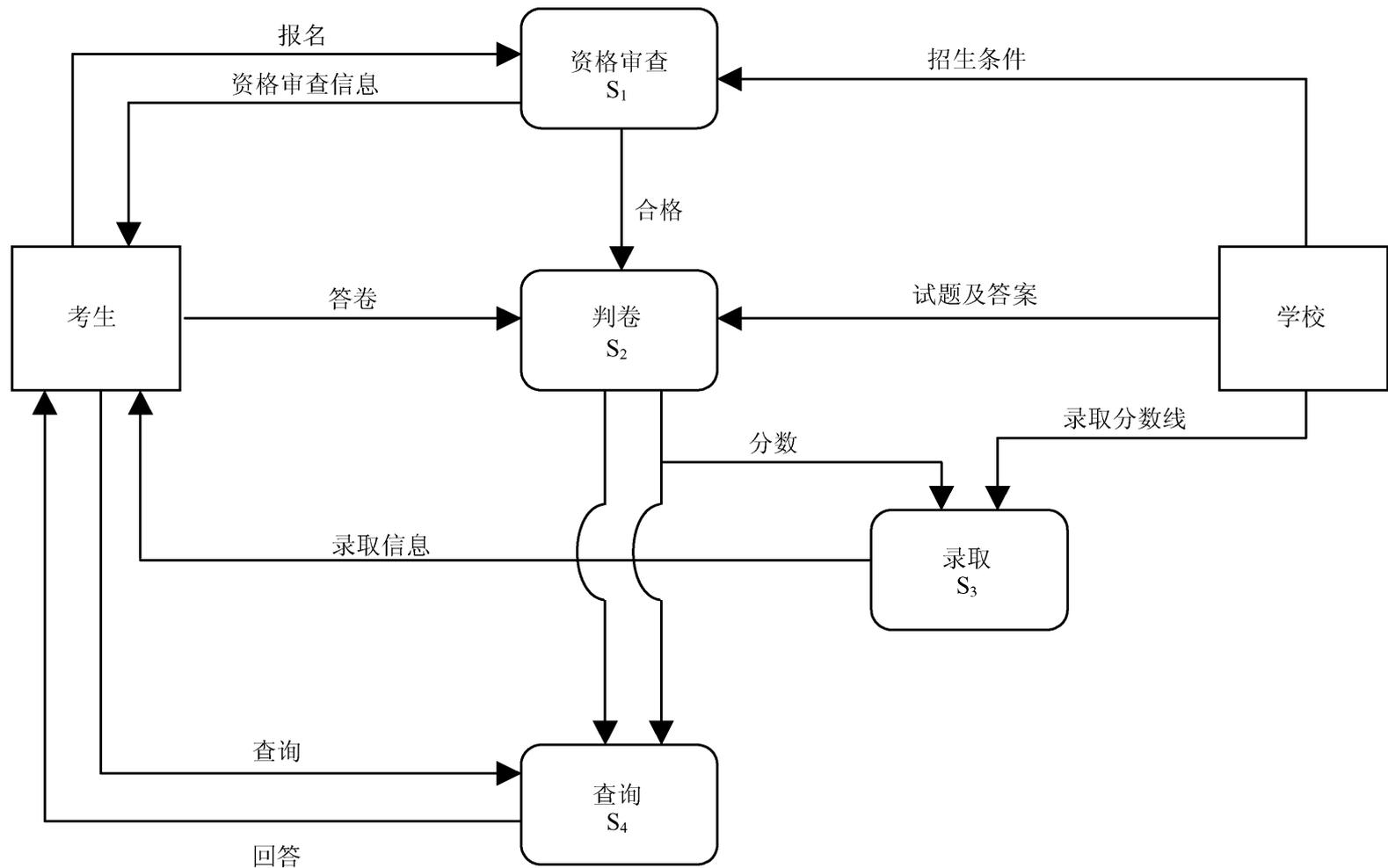


3.2.1 功能建模

- **数据流图的分层**
 - **对于稍微复杂一些的实际问题，在数据流图上常常出现十几个甚至几十个加工，这样的数据流图看起来不直观，不易理解，分层的数据流图能很好地解决这一问题。**
 - **按照系统的层次结构进行逐步分解，并以分层的数据流图反映这种结构关系，能清楚地表达和容易理解整个系统。**

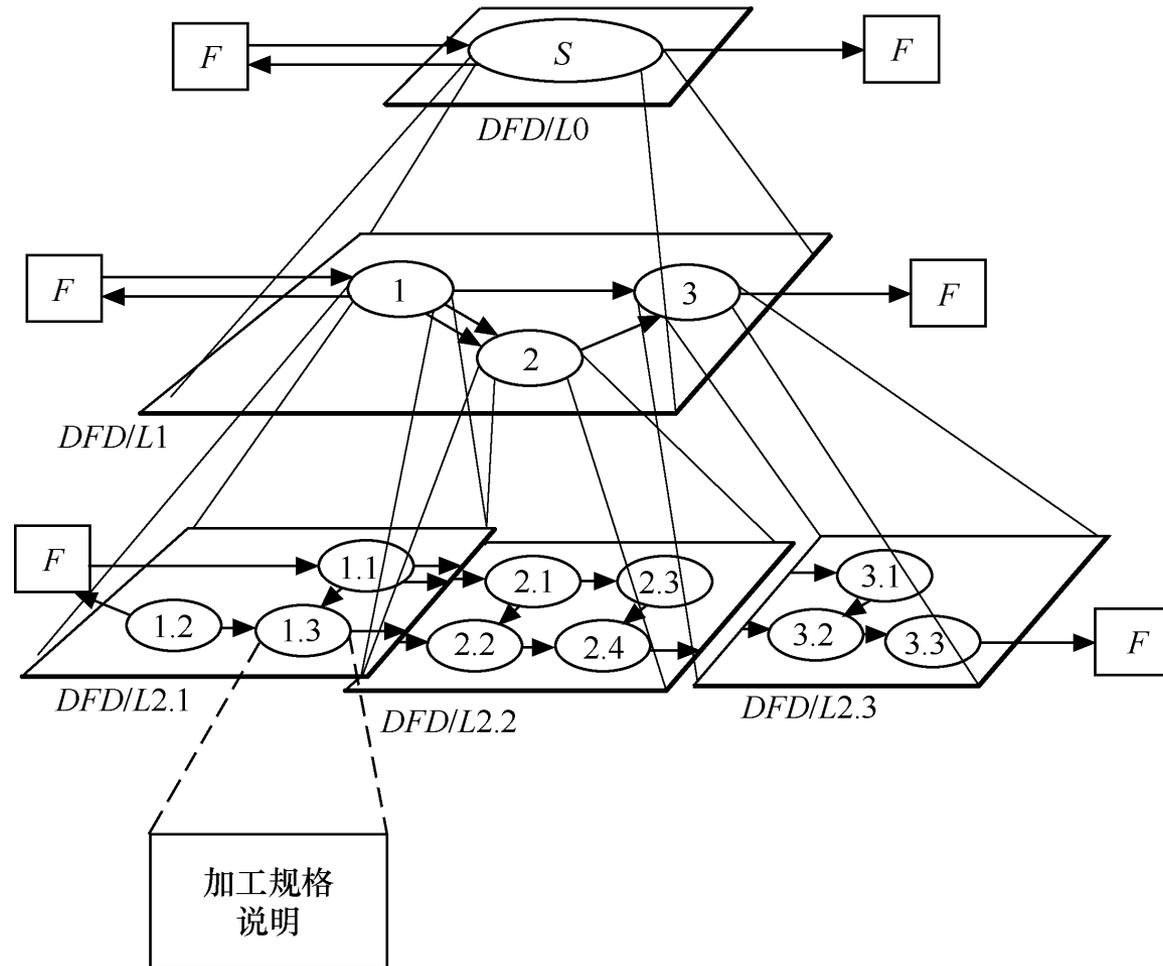
3.2.1 功能建模

• 招生系统的分层数据流图



3.2.1 功能建模

- 数据流图的分层示意图



3.2.1 功能建模

• 实例研究

银行储蓄系统的业务流程：

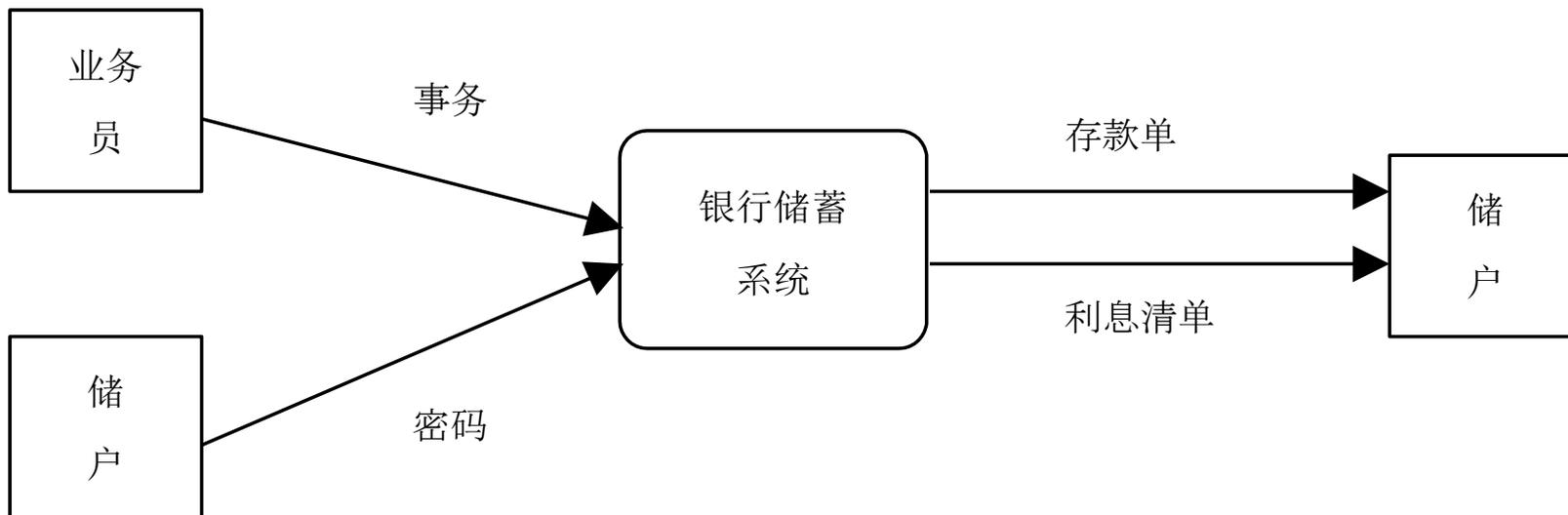
- **储户填写的存款单或取款单由业务员键入系统；**
- **如果是存款则系统记录存款人姓名、住址（或电话号码）、身份证号码、存款类型、存款日期、到期日期、利率、密码（可选）等信息，并印出存单给储户；**
- **如果是取款而且开户时留有密码，则系统首先核对储户密码，若密码正确或存款时未留密码，则系统计算利息并印出利息清单给储户。**
- **要求画出分层的数据流图，并细化到2层数据流图。**

3.2.1 功能建模

- (1) 识别外部实体及输入输出数据流。
 - **外部实体**：储户、业务员。
 - **输入数据**：如果需要储户输入密码，储户才直接与系统进行交互。储户填写的存款或取款信息通过业务员键入系统，可以将存款及取款信息抽象为事务。
 - **输出数据**：存款单，利息清单。

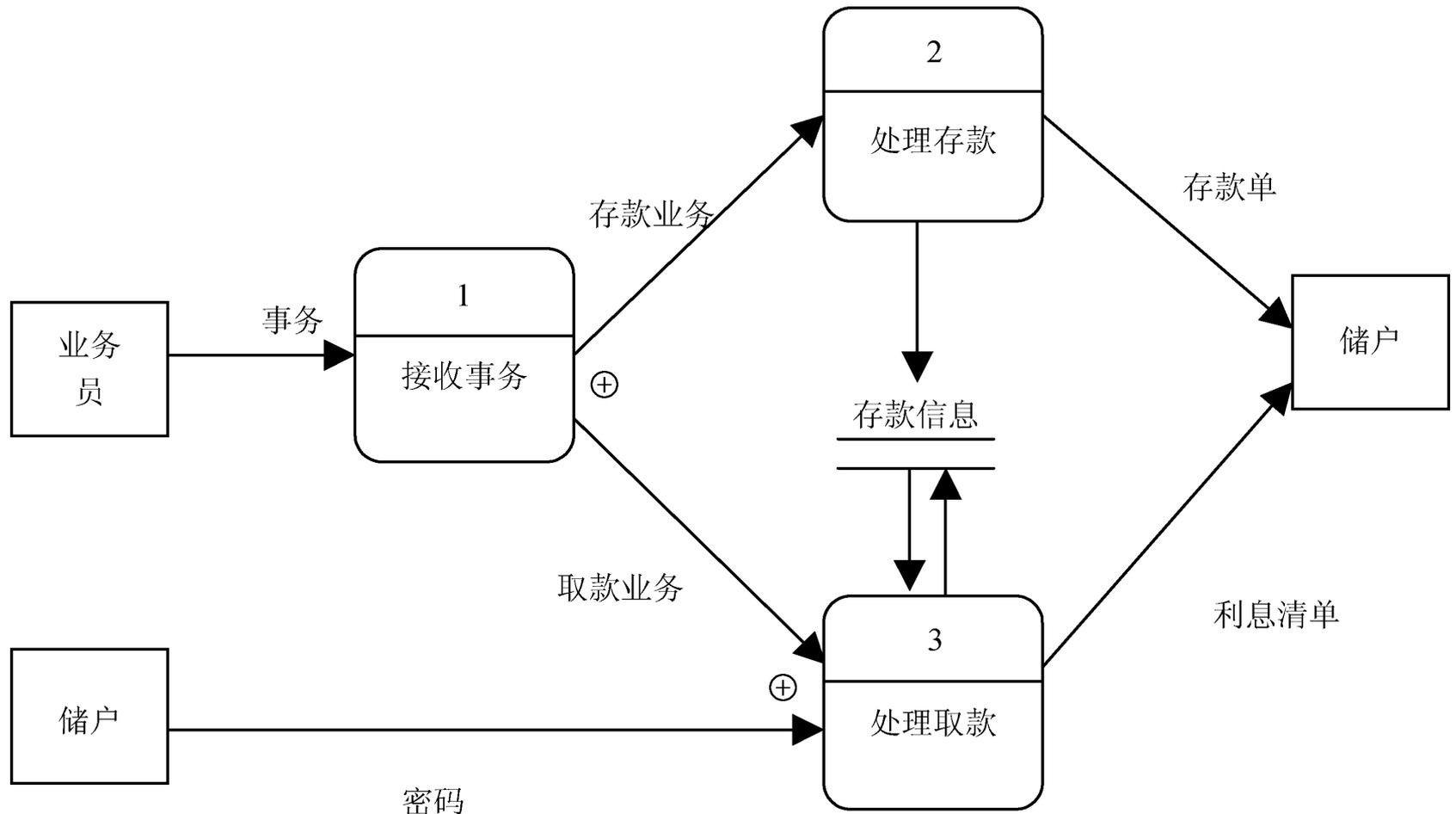
3.2.1 功能建模

- (2) 画出环境图（顶层数据流图）



3.2.1 功能建模

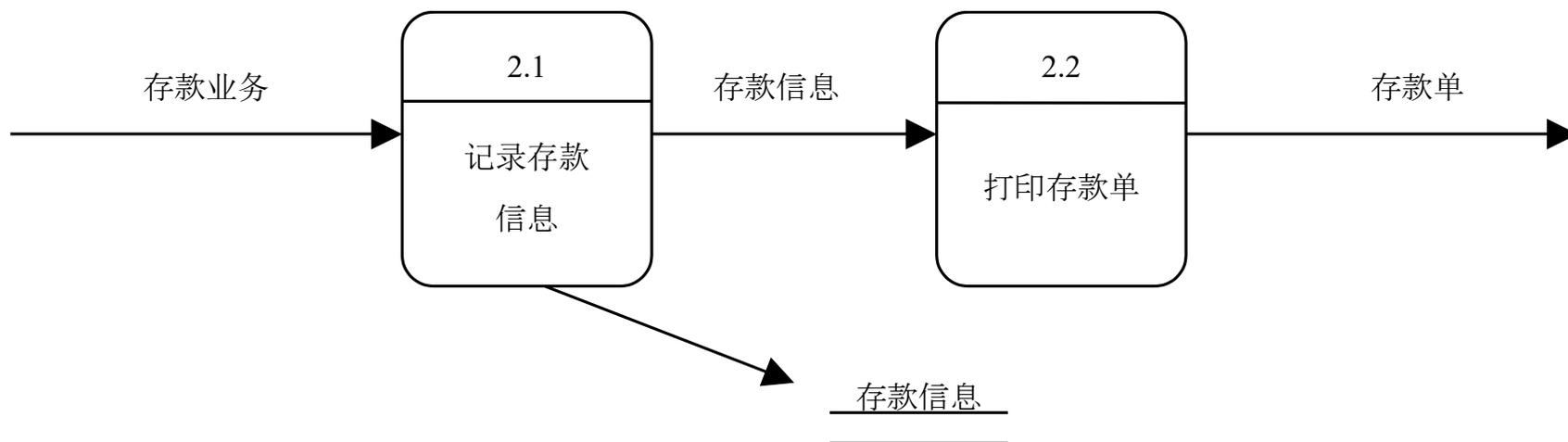
- (3) 画出一层数据流图



3.2.1 功能建模

- (4) 画出二层数据流图

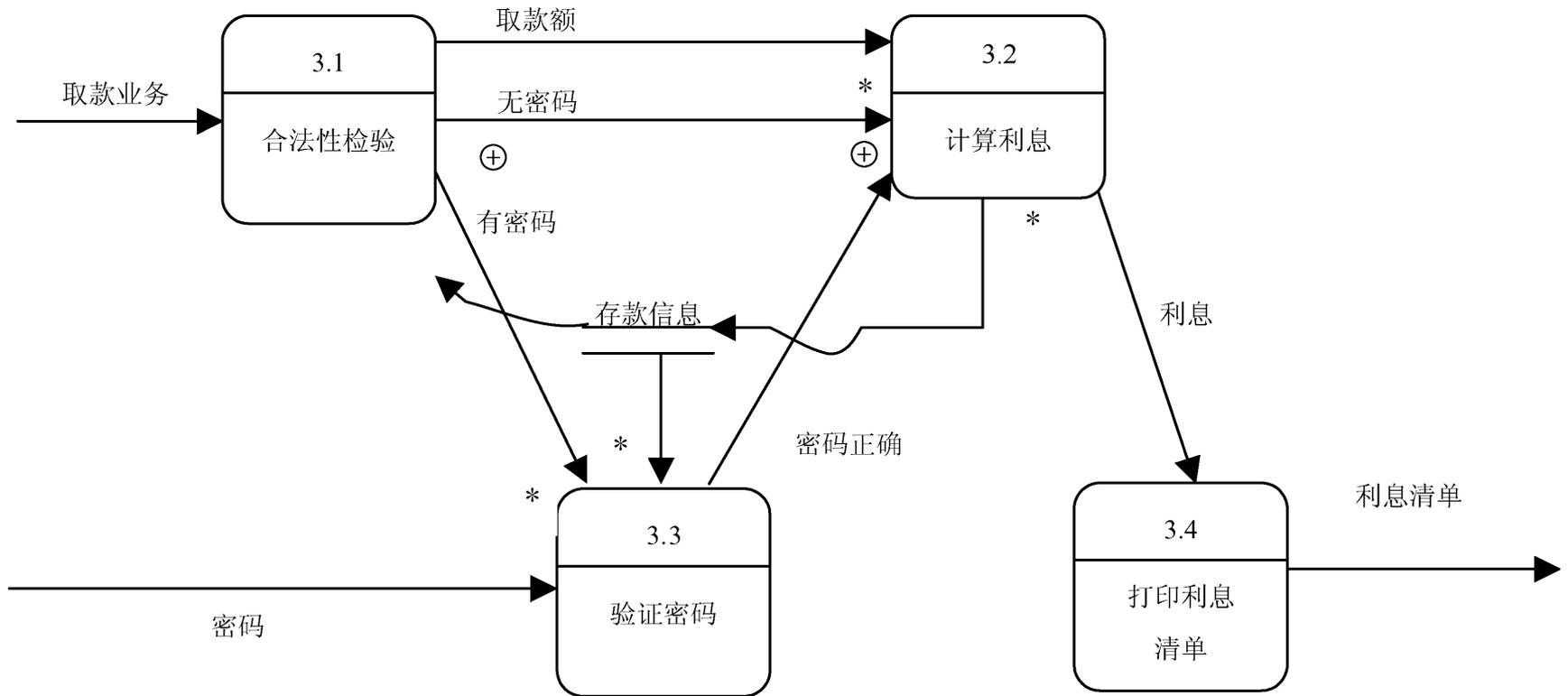
对一层图中的“处理存款”及“处理取款”进行进一步分解，得到二层数据流图，即**处理存款的数据流图**和**处理取款的数据流图**。



处理存款的数据流图

3.2.1 功能建模

- (4) 画出二层数据流图



处理取款的数据流图

3.2.2 数据建模

- 在结构化分析方法中，使用**实体—关系建模技术**来建立数据模型。
- 这种技术是在较高的抽象层次（概念层）上对数据库结构进行建模的流行技术。
- 实体—关系模型表示为可视化的**实体—关系图**（entity-relationship diagram, ERD），也称为ER图。
- ER图中仅包含3种相互关联的元素：**数据对象**（实体）、描述数据对象的**属性**及数据对象彼此间相互连接的**关系**。

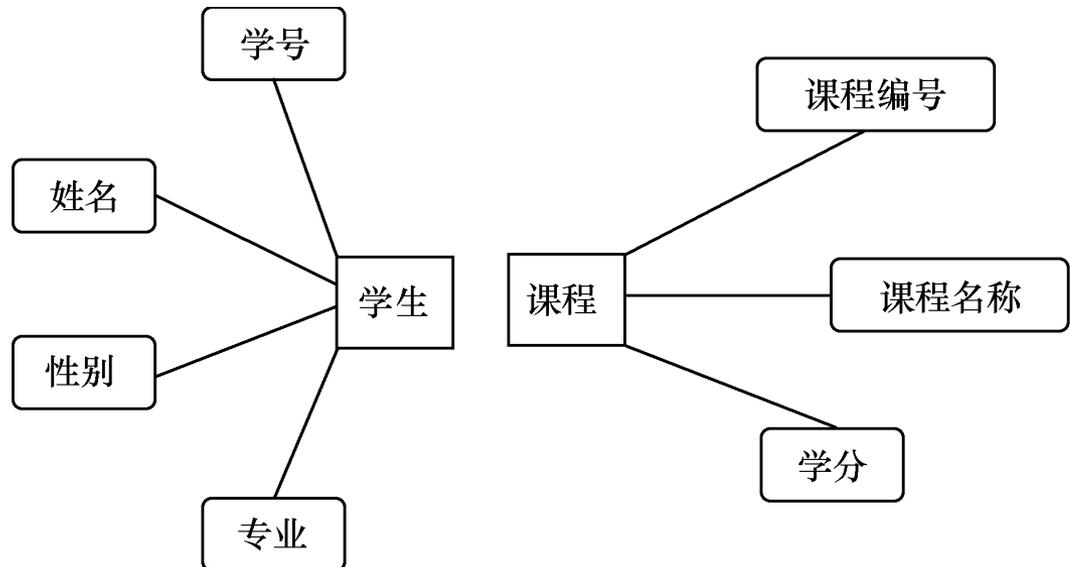
3.2.2 数据建模

- **数据对象**
 - **数据对象是目标系统所需要的复合信息的表示，所谓复合信息是具有若干不同属性的信息。在ER图中用矩形表示数据对象。**
 - **在实际问题中，数据对象（实体）可以是外部实体、事物、角色、行为或事件、组织单位、地点或结构等。**

3.2.2 数据建模

• 属性

- 属性定义数据对象的特征，如数据对象**学生**的学号、姓名、性别、专业等，**课程**的课程编号、课程名称、学分等。
- 在ER图中用椭圆或圆角矩形表示属性，并用无向边将属性与相关的数据对象连接在一起。



3.2.2 数据建模

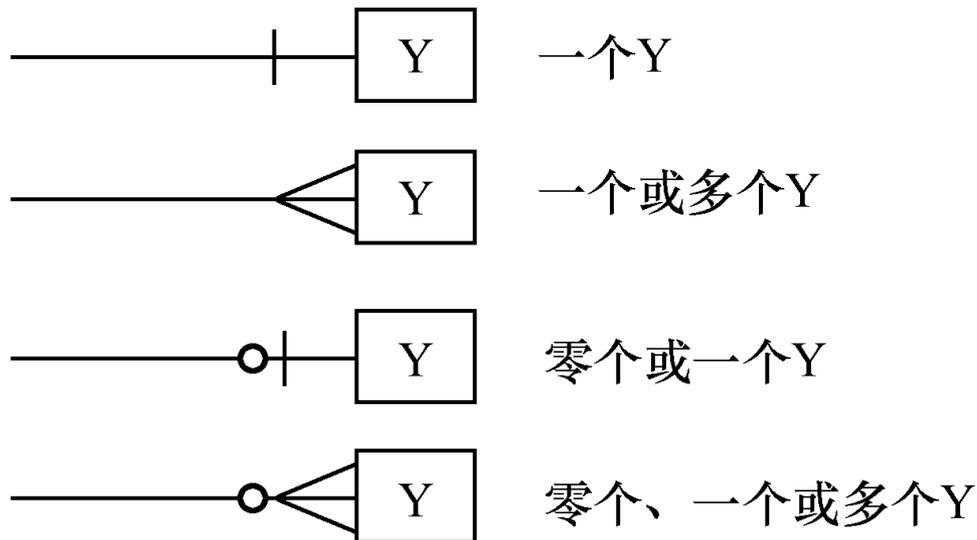
- 关系

- 不同数据对象的实例之间是有关联关系的，在ER图上用**无向边**表示。
- 在无向边的两端应标识出关联实例的数量，也称为关联的**重数**。
- 从关联重数的角度可以将关联分为 3 种。
 - (1) **一对一 (1:1) 关联**
 - (2) **一对多 (1:m) 关联**
 - (3) **多对多 (m:n) 关联**
- 实例关联还有“必须”和“可选”之分。

3.2.2 数据建模

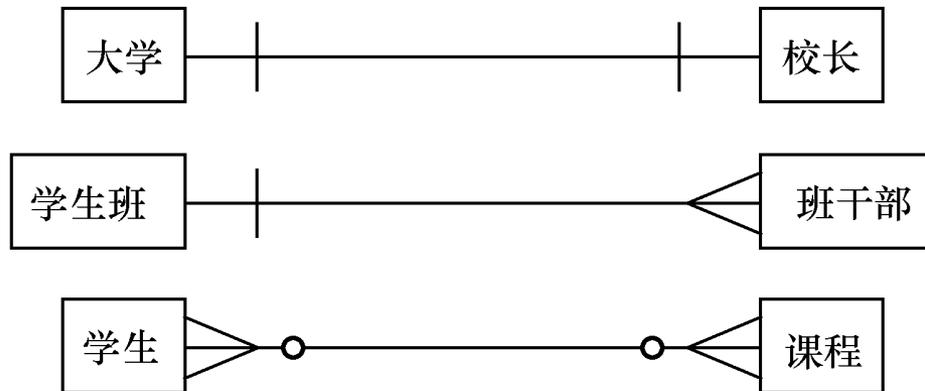
• 关联数量的表示

- ▶ 在ER图中用圆圈表示所关联的实例是可选的，隐含表示“0”，没有出现圆圈就意味着是必须的。出现在连线上的短竖线可以看成是“1”。



3.2.2 数据建模

- 关联关系举例



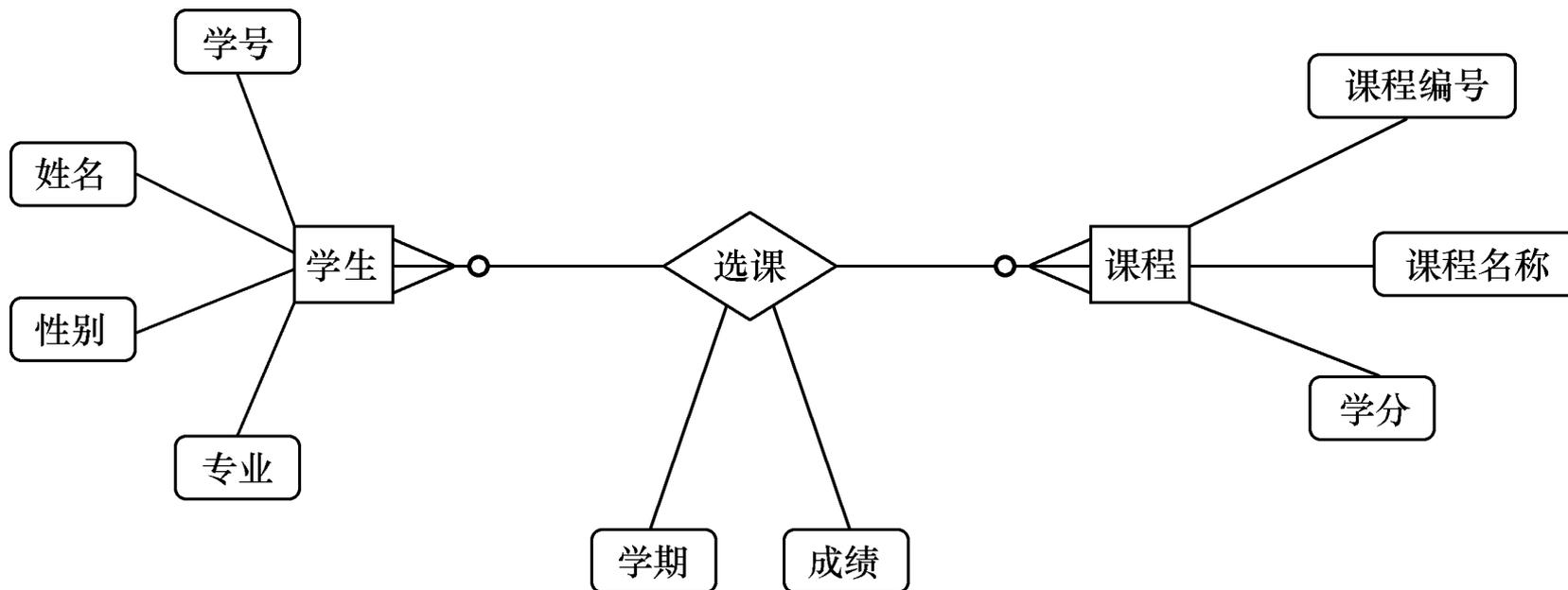
3.2.2 数据建模

- **关系的属性**

- **关系本身也可能有属性**，这在多对多的关系中尤其常见，如学生和课程之间的关系可起名为“选课”，其**属性**应该有**学期**、**成绩**等。
- **关系属性的表示**：在表示关系的无向边上再加一个菱形框，并在菱形框中标明关系的名字，关系的属性同样用椭圆形或圆角矩形表示，并用无向边将关系与其属性连接起来。

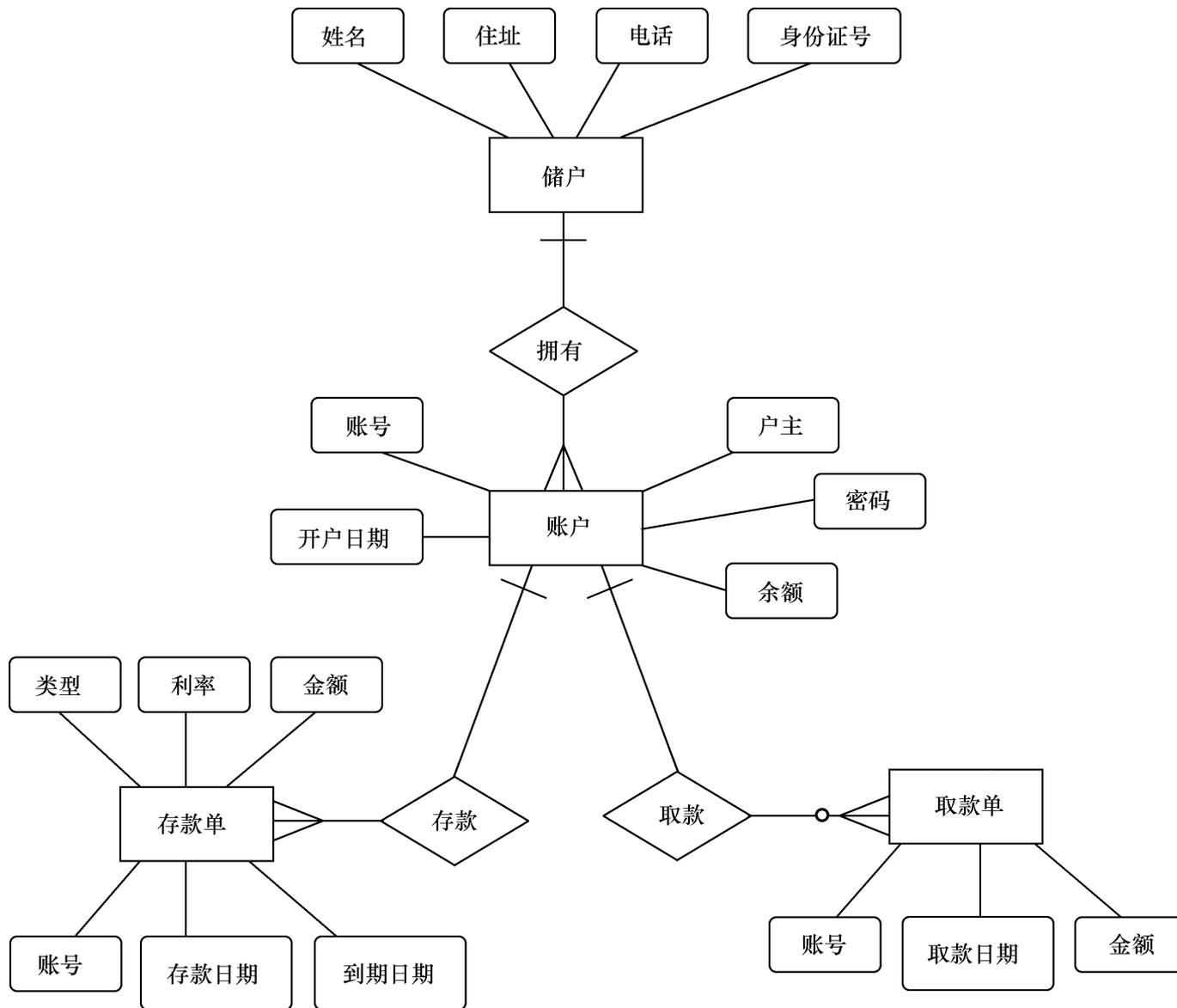
3.2.2 数据建模

• 关系的属性



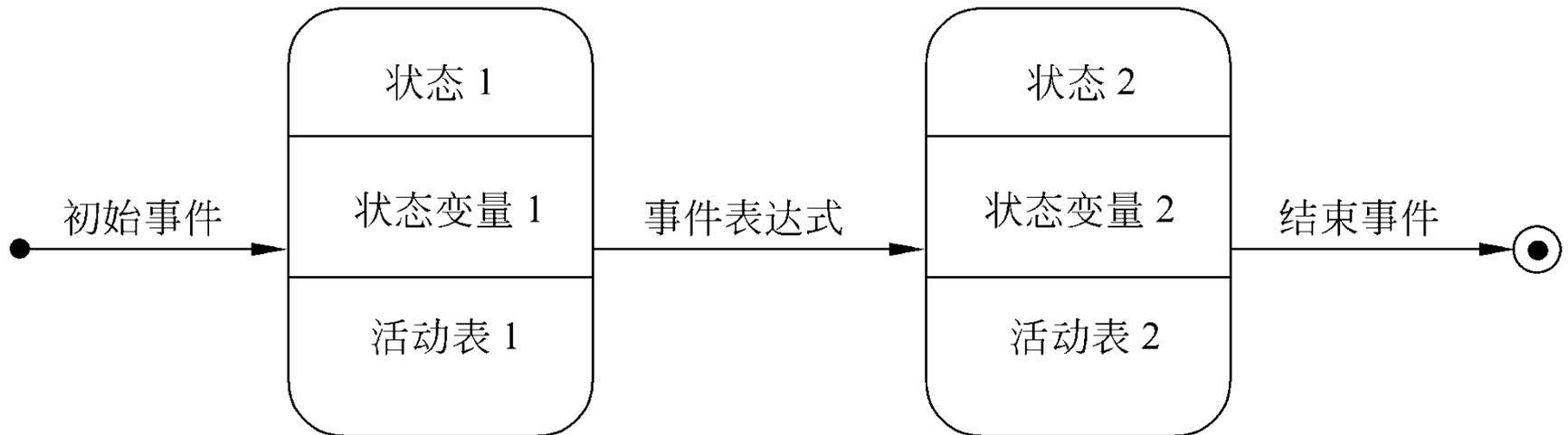
3.2.2 数据建模

- 银行储蓄系统的ER图



3.2.3 行为建模

- **状态转换图（简称状态图）通过描绘系统的状态及引起系统状态转换的事件，来表示系统的行为。状态图中使用的主要符号如图所示。**



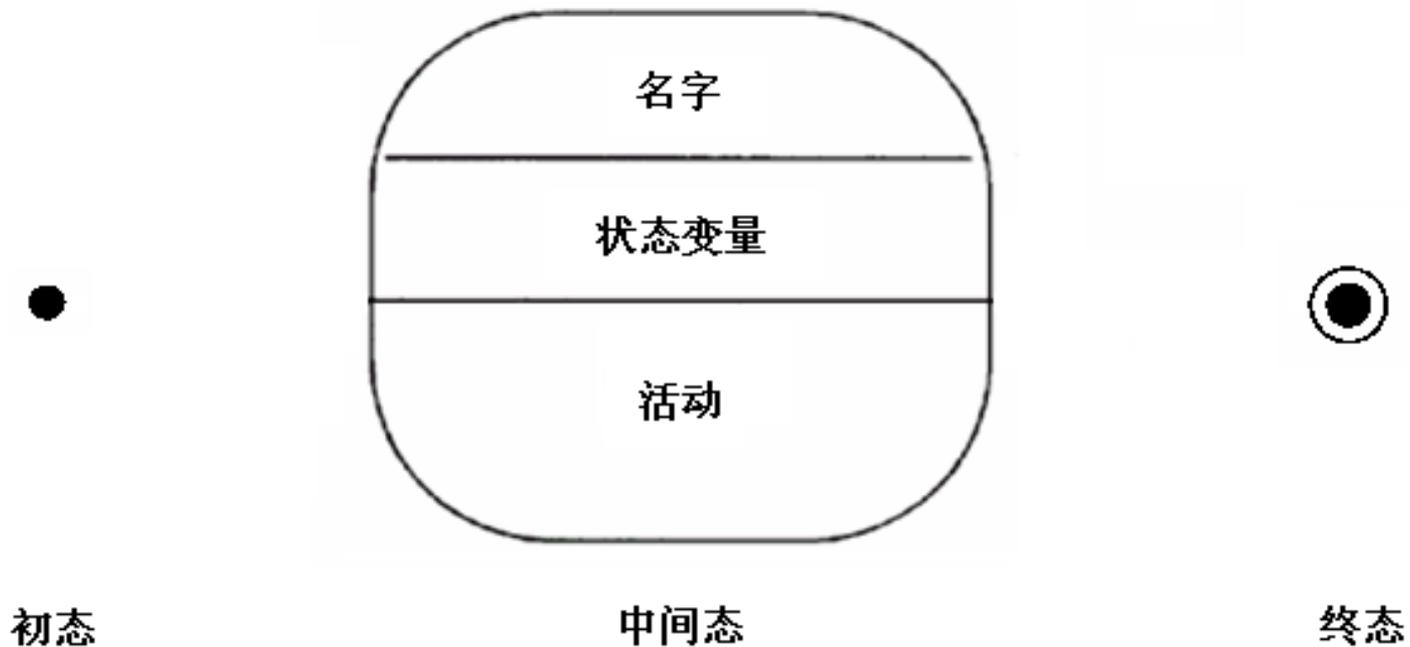
3.2.3 行为建模

- **状态**

- **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，状态规定了系统对事件的响应方式。
- 状态可能有：**初态**（初始状态）、**终态**（最终状态）和**中间态**。
- 在一张状态图中只能有一个初态，而终态则可以有多，也可以没有。

3.2.3 行为建模

- **状态的表示：**初态用**实心圆**表示，终态用**牛眼**图形表示，中间态用**圆角矩形**表示。



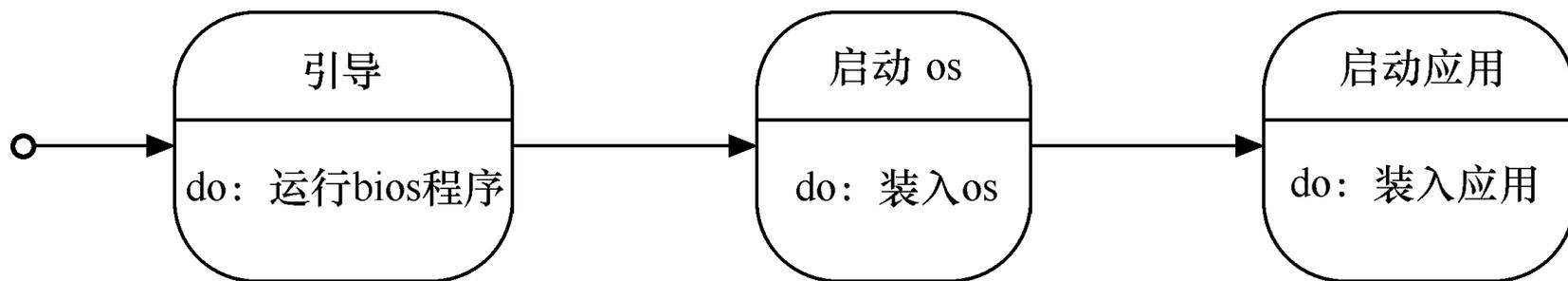
3.2.3 行为建模

- **状态转换**
 - **状态图中两个状态之间带箭头的连线称为状态转换。**
 - **状态的变迁通常是由事件触发的，在这种情况下应在表示状态转换的箭头线上标出触发转换的事件表达式。**
 - **如果在箭头线上未标明事件，则表示在源状态的内部活动执行完之后自动触发转换。**

3.2.3 行为建模

- 状态转换

- 下图为计算机应用程序的启动过程，在这个过程中没有外部事件触发，每个状态下的活动完成时，状态发生转换。



3.2.3 行为建模

- **事件**

- 事件是在某个特定时刻发生的事情，它是对引起系统做动作或从一个状态转换到另一个状态的外部事件的抽象。事件表达式的语法如下：

事件说明 (守卫条件) /动作表达式

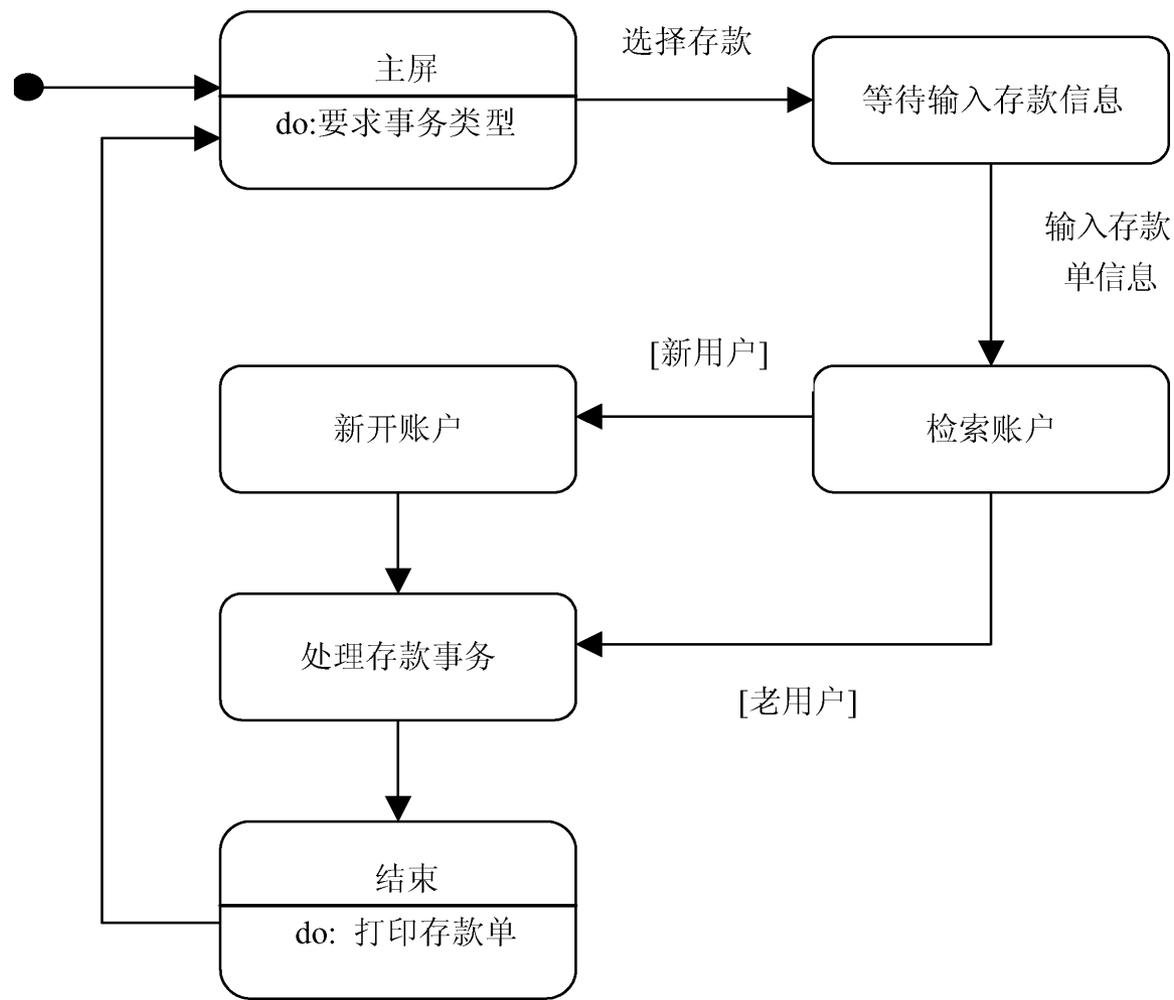
- (1) **事件说明**的语法如下：

事件名 (参数表)

- (2) **守卫条件**是一个布尔表达式。如果同时使用守卫条件和事件说明，则当且仅当事件发生且布尔表达式成立时，状态转换才发生。如果只有守卫条件没有事件说明，则只要守卫条件为真，状态转换就发生。
- (3) **动作表达式**是一个过程表达式，当状态转换开始时执行该表达式。

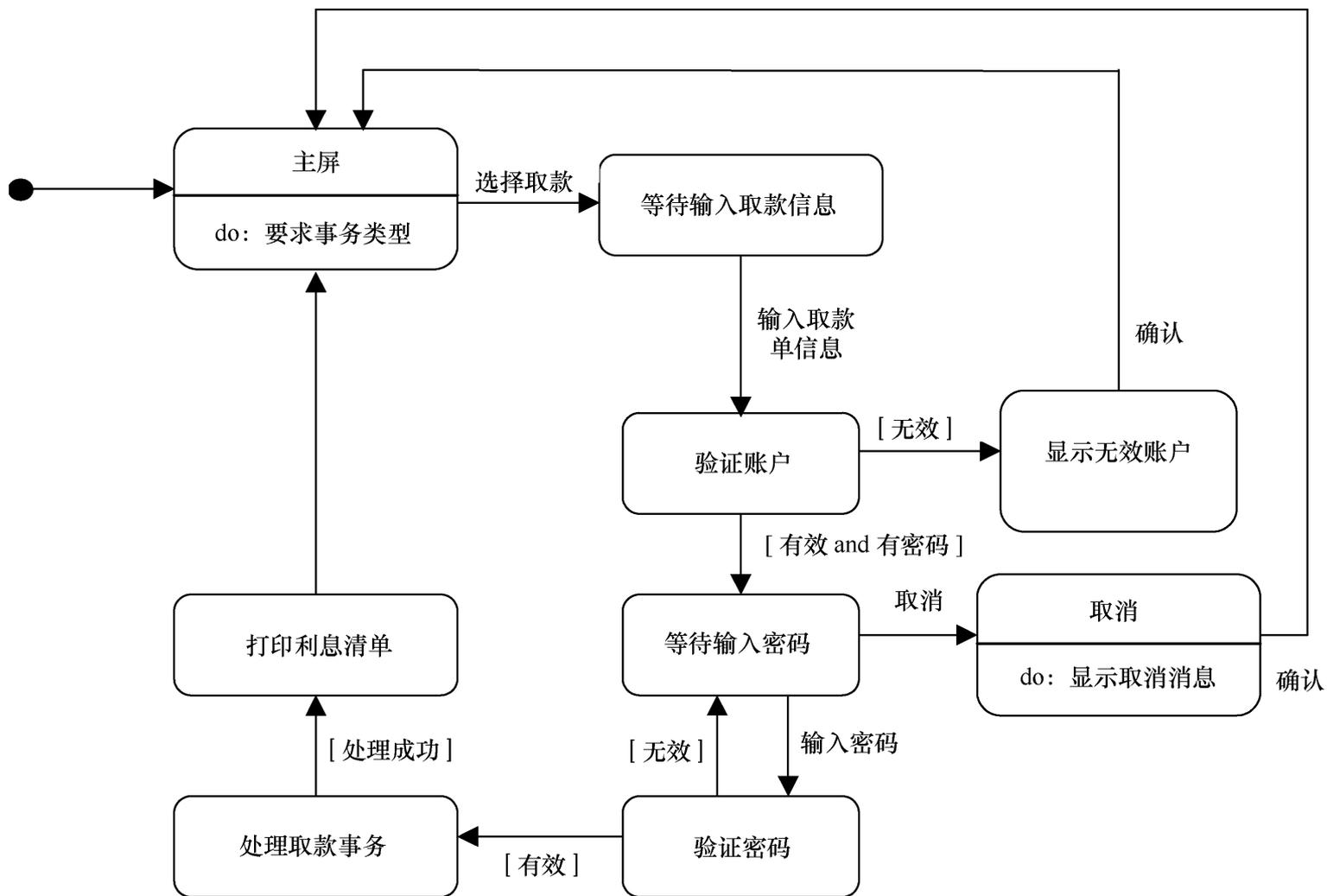
3.2.3 行为建模

• 存款过程的状态图(考虑新开户)



3.2.3 行为建模

取款过程的状态图



3.2.4 数据字典

- **数据字典以词条方式定义在数据模型、功能模型和行为模型中出现的数据对象及控制信息的特性，给出它们的准确定义，包括数据流、加工、数据文件、数据元素，以及数据源点、数据汇点等。**
- **数据字典成为把3种分析模型黏合在一起的“黏合剂”，是分析模型的核心”。**

3.2.4 数据字典

- **词条描述**
 - **对于在数据流图中每一个被命名的图形元素均加以定义；**
 - **其内容包括图形元素的名字，图形元素的别名或编号，图形元素类别（如加工、数据流、数据文件、数据元素、数据源点或数据汇点等）、描述、定义、位置等。**

3.2.4 数据字典

• 数据流词条

➤ 数据流是数据结构在系统内传播的路径，数据流词条应包括以下几项内容。

- ① 数据流名：要求与数据流图中该图形元素的名字一致。
- ② 简述：简要介绍它产生的原因和结果。
- ③ 组成：数据流的数据结构。
- ④ 来源：数据流来自哪个加工或作为哪个数据源的外部实体。
- ⑤ 去向：数据流流向哪个加工或作为哪个数据汇点的外部实体。
- ⑥ 流通量：单位时间数据的流通量。
- ⑦ 峰值：流通量的极限值。

3.2.4 数据字典

• 数据元素词条

- 数据流图中的每个数据结构都是由数据元素构成的，数据元素是数据处理中最小的、不可再分的单位，它直接反映事物的某一特征。
 - ① 类型：数据元素分为数字型与文字型。数字型又分为离散值和连续值，文字的类型用编码类型和长度区分。
 - ② 取值范围：离散值的取值或是枚举的（如3, 17, 21），或是介于上下界的一组数（如2..100）；连续值一般是有取值范围的实数集（如0.0..100.0）。对于文字型，文字的取值需加以定义。
 - ③ 相关的数据元素及数据结构。

3.2.4 数据字典

• 数据存储文件词条

➤ 数据存储文件是数据保存的地方。一个数据存储文件词条应有以下几项内容。

- ① 文件名：要求与数据流图中该图形元素的名字一致。
- ② 简述：简要介绍存放的是什么数据。
- ③ 组成：文件的数据结构。
- ④ 输入：从哪些加工获取数据。
- ⑤ 输出：由哪些加工使用数据。
- ⑥ 存取方式：分为顺序、直接、关键码等不同存取方式。
- ⑦ 存取频率：单位时间的存取次数。

3.2.4 数据字典

• 加工词条

➤ 加工可以使用诸如判定表、判定树、结构化语言等形式表达，主要描述如下。

- ① 加工名：要求与数据流图中该图形元素的名字一致。
- ② 编号：用以反映该加工的层次和父子关系。
- ③ 简述：加工逻辑及功能简述。
- ④ 输入：加工的输入数据流。
- ⑤ 输出：加工的输出数据流。
- ⑥ 加工逻辑：简述加工程序和加工顺序。

3.2.4 数据字典

• 数据源点及数据汇点词条

➤ 对于一个数据处理系统来说，数据源点和数据汇点应比较少。

- ① 名称：要求与数据流图中该外部实体的名字一致。
- ② 简述：简要描述是什么外部实体。
- ③ 有关数据流：该实体与系统交互时涉及哪些数据流。
- ④ 数目：该实体与系统交互的次数。

3.2.4 数据字典

- **数据结构描述**

- **在数据字典的编制中，分析员最常用的描述数据结构的方式有定义式、Warnier图等。**
- (1) **定义式。在数据流图中，数据流和数据文件都具有一定的数据结构，因此，必须以一种清晰、准确、无二义性的方式来描述数据结构。**
- (2) **Warnier图。Warnier图是表示数据结构的另一种图形工具，它用树形结构来描绘数据结构。**

3.2.4 数据字典

• 定义式中的符号

数据结构定义式中的符号

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, $x=[a, b]$, $x=[ab]$, 表示 x 由 a 或由 b 组成
[... ...]	或	
{...}	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 x 中至少出现 3 次 a , 至多出现 8 次 a
(...)	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 为取值为 a 的数据元素
...	连接符	例如, $x=1..9$, 表示 x 可取 1 到 9 之中的任一值
..		

3.2.4 数据字典

• 存折的定义格式

存折 = 户名 + 所号 + 账号 + 开户日 + 性质 + (印密) +
1{存取行}50

所号 = "001".. "999"

户名 = 2{字母}24

账号 = "000000000001".. "999999999999"

开户日 = 年 + 月 + 日

性质 = "1".. "6"

印密 = ("0" | "000001".. "999999")

存取行 = 日期 + (摘要) + 支出 + 存入 + 余额 + 操作 + 复核

日期 = 年 + 月 + 日

年 = "0001".. "9999"

月 = "01".. "12"

日 = "01".. "31"

3.2.4 数据字典

• 存折的定义格式

摘要 = 1{字母}4

支出 = 金额

存入 = 金额

余额 = 金额

金额 = “0000000.01”..“99999999.99”

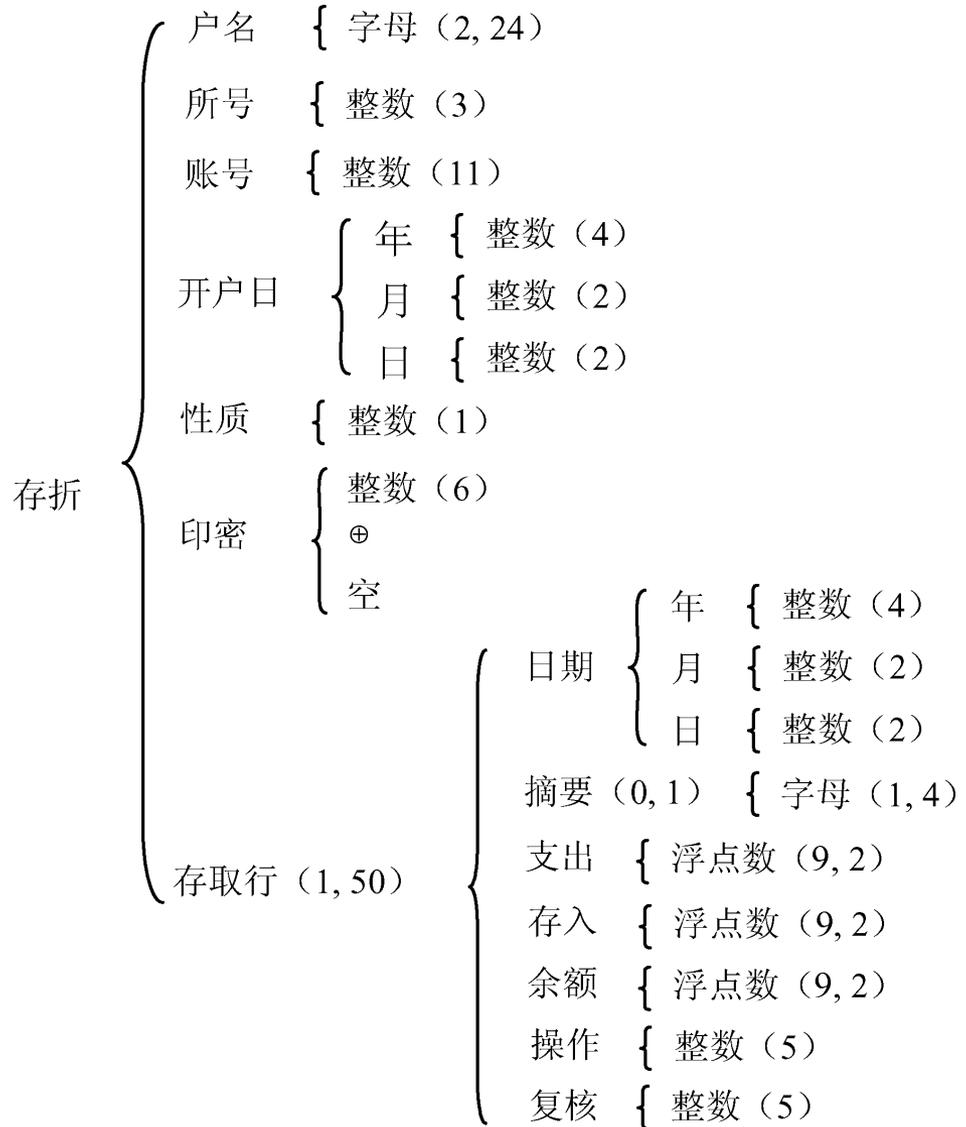
操作 = “00001”..“99999”

复核 = “00001”..“99999”

字母 = [“a”..“z”|“A”..“Z”]

3.2.4 数据字典

• Warnier图 举例：存折



3.2.5 加工规格说明

- 在对数据流图的分解中，位于层次树最低层的加工也称为**基本加工**或**原子加工**，对于每一个基本加工都需要进一步说明，这称为**加工规格说明**。
- 在编写基本加工的规格说明时，主要目的是要表达“做什么”，而不是“怎样做”。

3.2.5 加工规格说明

- **加工规格说明应满足如下的要求：**

- (1) 对数据流图的每一个基本加工，必须有一个加工规格说明。**
- (2) 加工规格说明必须描述基本加工如何把输入数据流变换为输出数据流的加工规则。**
- (3) 加工规格说明必须描述实现加工的策略而不是实现加工的细节。**
- (4) 加工规格说明中包含的信息应是充足的，完备的，有用的，没有重复的多余信息。**

3.2.5 加工规格说明

• 决策表

决策表由4个部分组成：

- 左上部分是**条件荏**，在此区域列出了各种可能的单个条件；
- 左下部分是**动作荏**，在此区域列出了可能采取的单个动作；
- 右上部分是**条件项**，在此区域列出了针对各种条件的每一组条件取值的组合；
- 右下部分是**动作项**，这些动作项与条件项紧密相关，它指出了在条件项的各组取值的组合情况下应采取的动作。

3.2.5 加工规格说明

• 决策表举例

商店业务处理系统中“检查订货单”的决策表。

		1	2	3	4
条件	订货单金额	>5000 元	>5000 元	≤5000 元	≤5000 元
	偿还欠款情况	>60 天	≤60 天	>60 天	≤60 天
动作	在偿还欠款前 不予批准	√			
	发出批准书		√	√	√
	发出发货单		√	√	√
	发催款通知书			√	

3.2.5 加工规格说明

• 决策表的改进

如果表中有两条或更多的处理规则具有相同的动作，并且其条件项之间存在着某种关系，就可设法将它们合并。

改进的“检查订货单”的决策表

		1	2	3
条件	订货单金额	>5000 元	—	≤5000 元
	偿还欠款情况	>60 天	≤60 天	>60 天
动作	在偿还欠款前不予批准	√		
	发出批准书		√	√
	发出发货单		√	√
	发催款通知书			√

3.2.5 加工规格说明

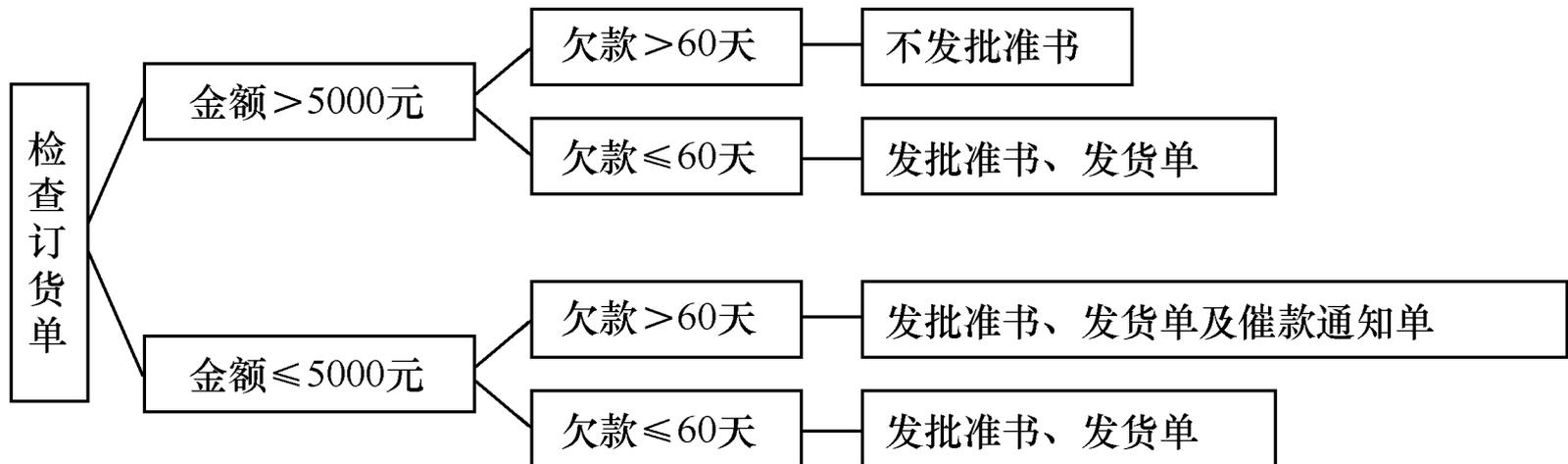
• 建立决策表的步骤

- (1) 列出与一个具体过程（或模块）有关的所有处理。
- (2) 列出过程执行期间的所有条件（或所有判断）。
- (3) 将特定条件取值组合与特定的处理相匹配，消去不可能发生的条件取值组合。
- (4) 将右部每一纵列规定为一个处理规则，即对于某一条件取值组合将有什么动作。

3.2.5 加工规格说明

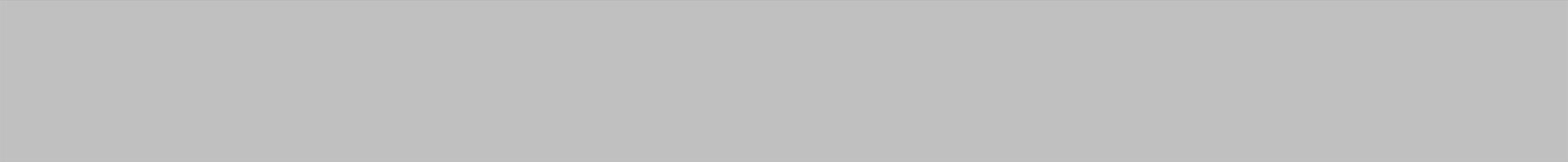
• 决策树

- 决策树 (decision tree) 也是用来表达加工逻辑的一种工具, 有时候它比决策表更直观。
- 检查订货单的决策树



3.3 系统需求规格说明

- **需求分析阶段的重要任务之一是根据分析的结果编写需求规格说明，经过严格评审并得到用户确认之后，作为这个阶段的最终成果。**
- **按照国家标准GB/T 8567—2006《计算机软件文档编制规范》，涉及需求规格说明的文档有“软件需求规格说明（SRS）”、“数据需求说明（DRD）”等。**



That's All!