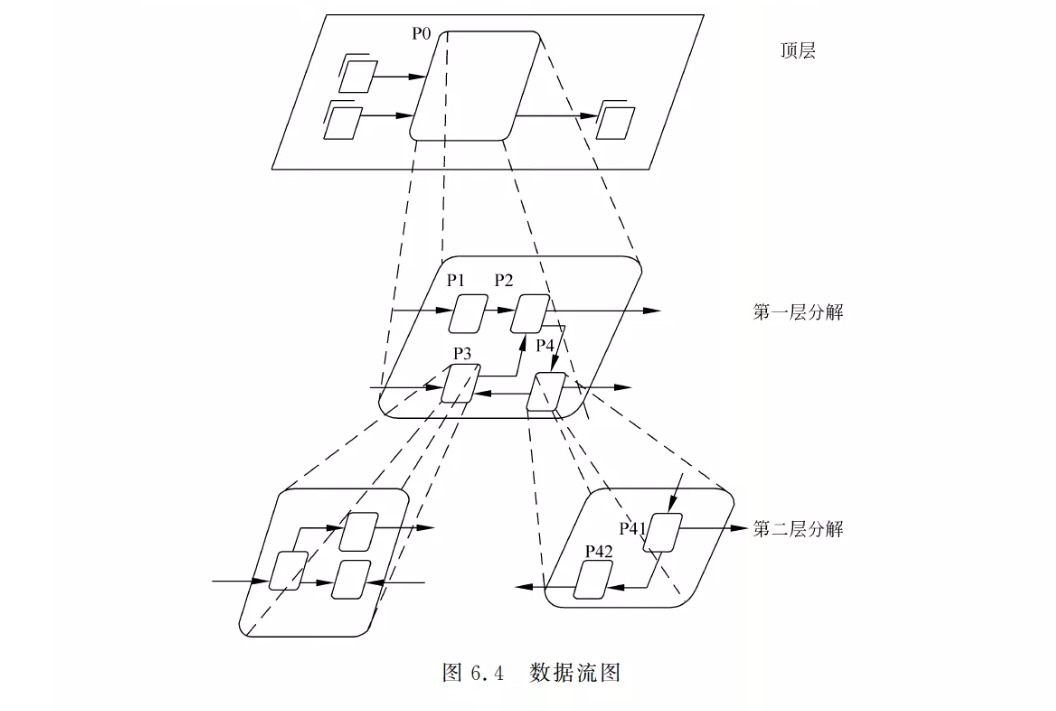
**软件工程导论图表示例**

* 数据流图

 图6.4为数据流图的总体结构，一个系统可以分为多层来描述。

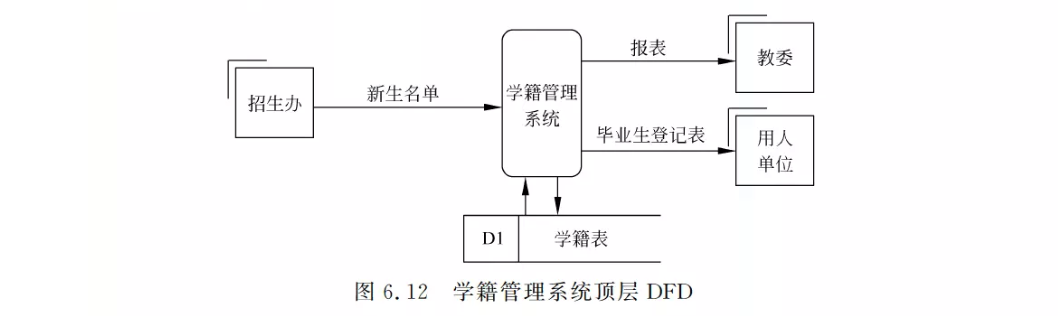
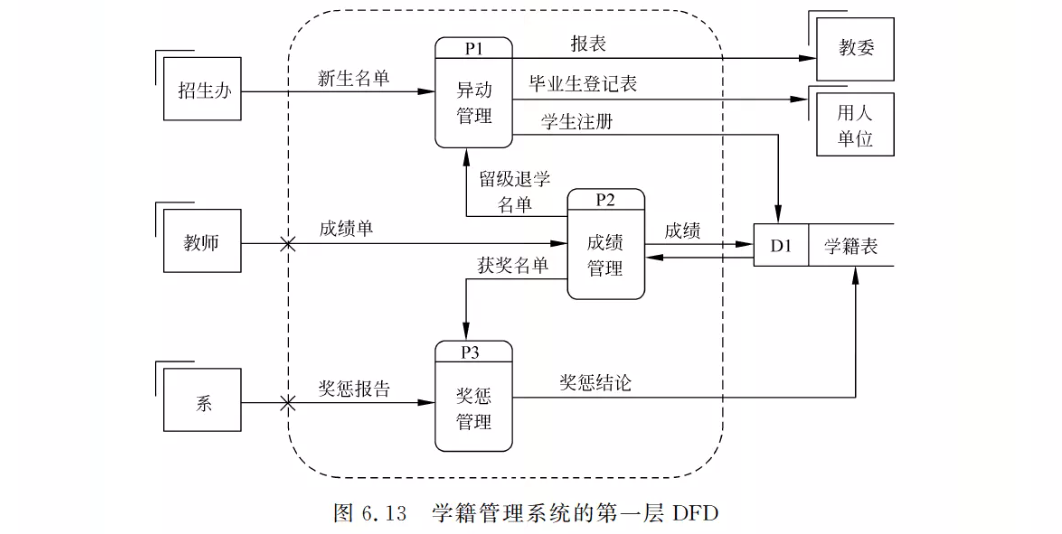


图6.12为学籍管理系统的顶层数据流图

招生办、教委、用人单位为用户角色，”学籍管理系统“是我们需要设计的系统，“D1数据表”指的是数据表

 图6.13为学籍管理系统顶层数据流图的进一步细化，该图为第一层数据流图

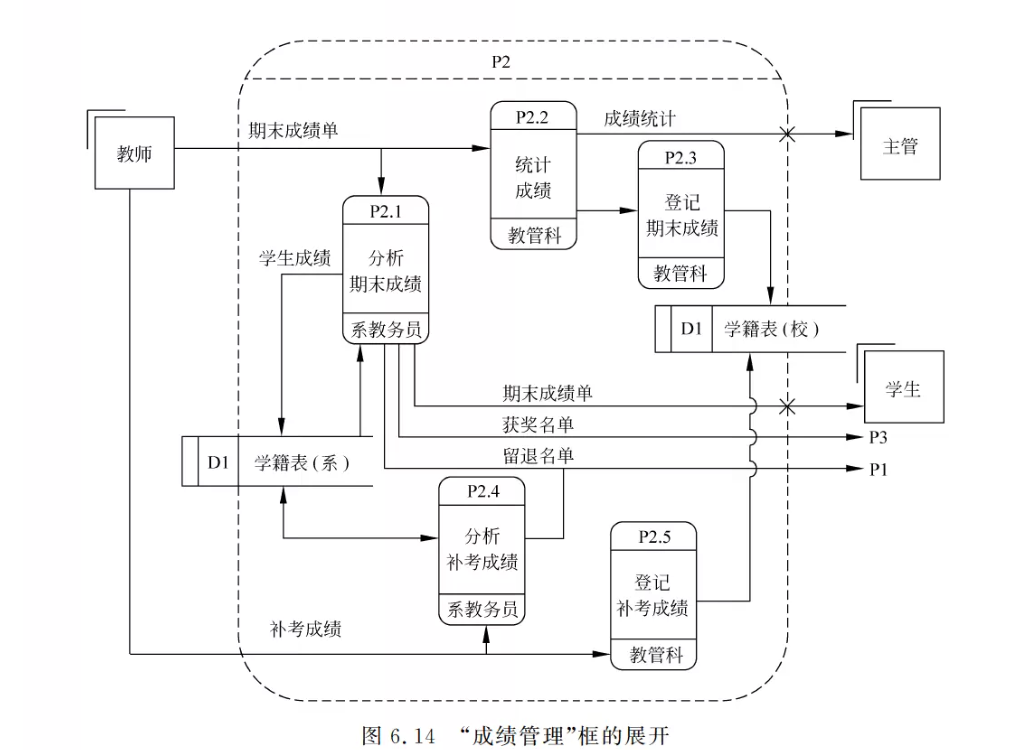


图6.14为第一层数据流图中“成绩模块”展开的数据流图

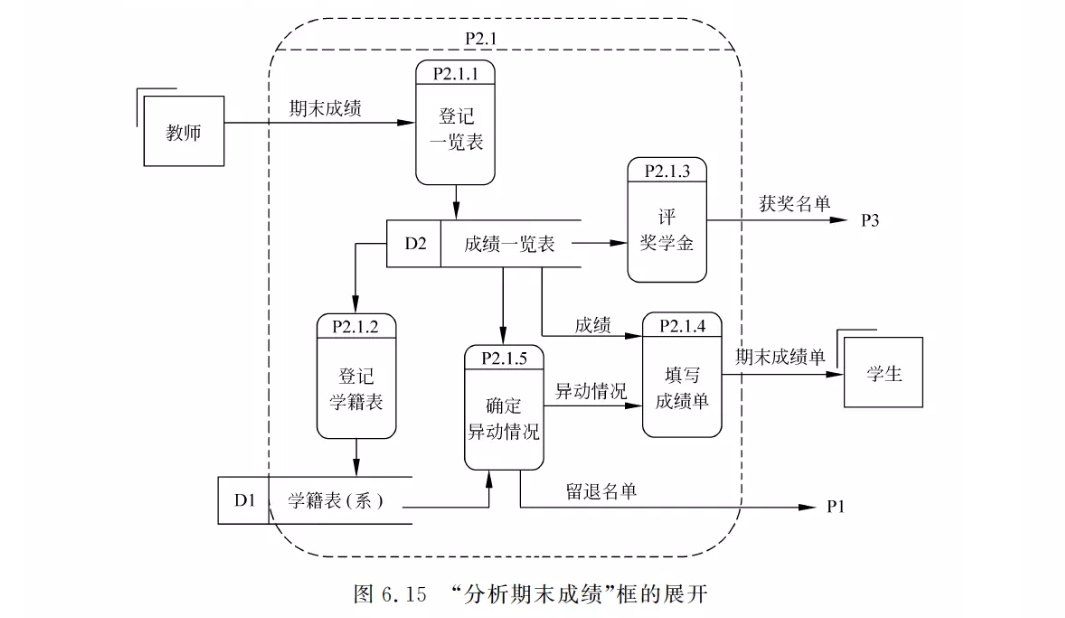


图6.15为第一层数据流图中“分析期末成绩”展开的数据流图

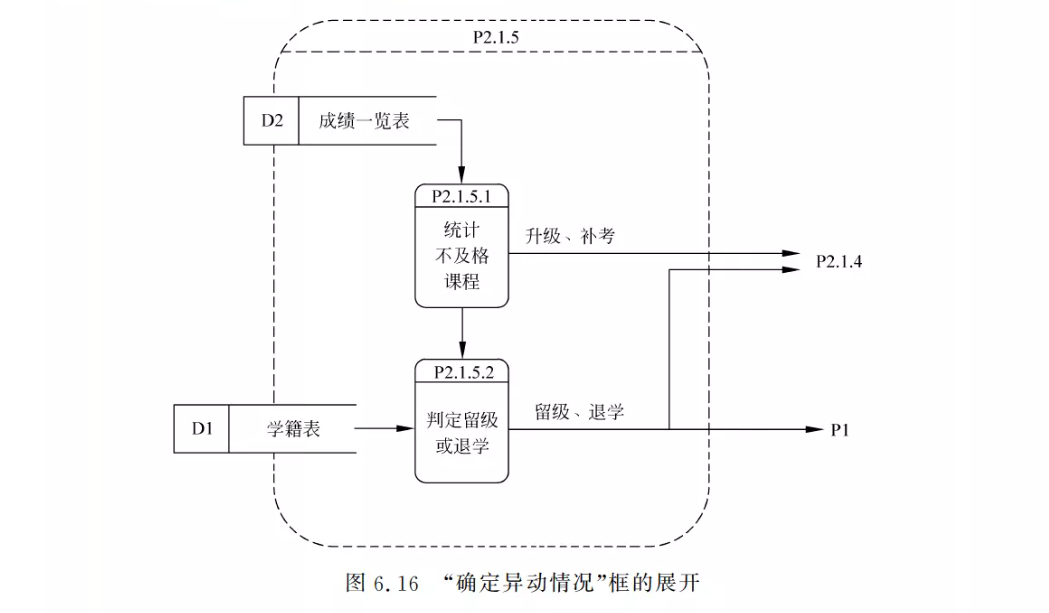
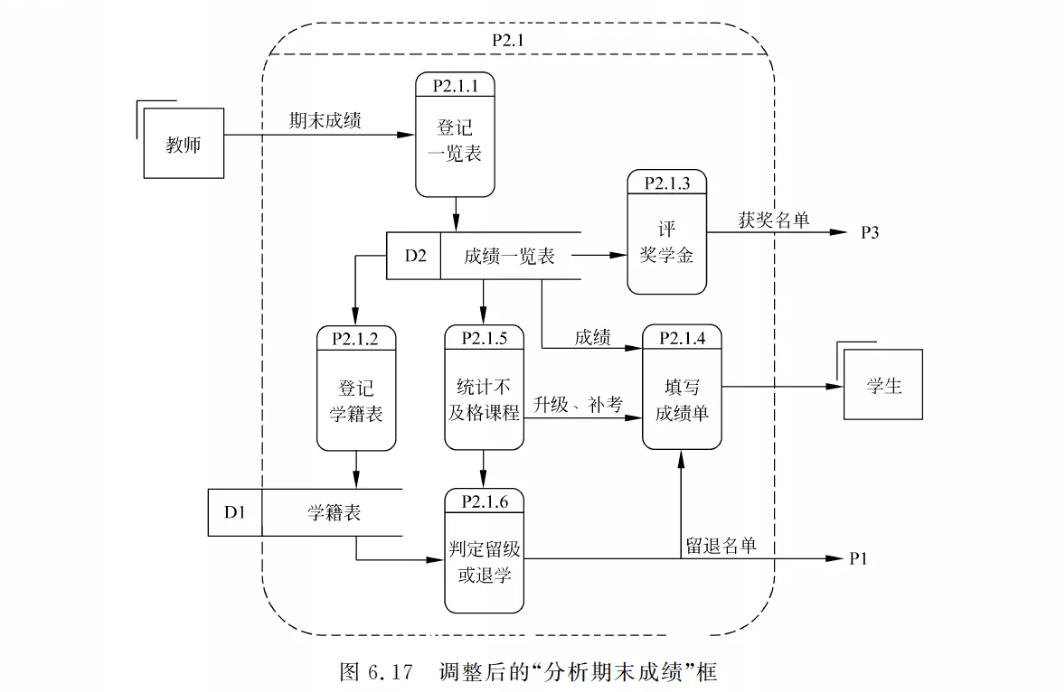


图6.16为第一层数据流图中“确定异动情况”展开的数据流图

 图6.17为第一层数据流图中“分析期末成绩”优化后的数据流图

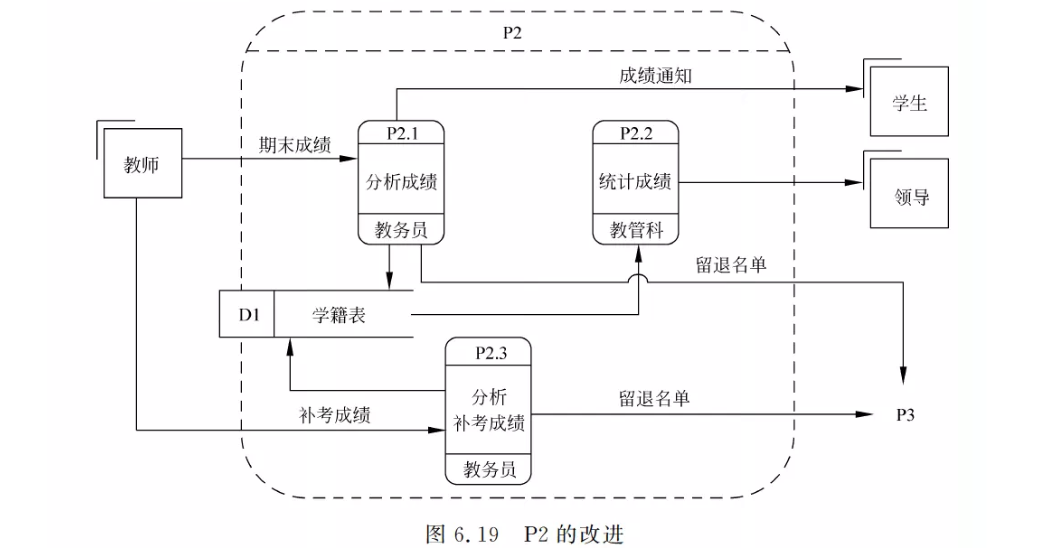


图6.19为P2优化后的数据流程图

* 盒图

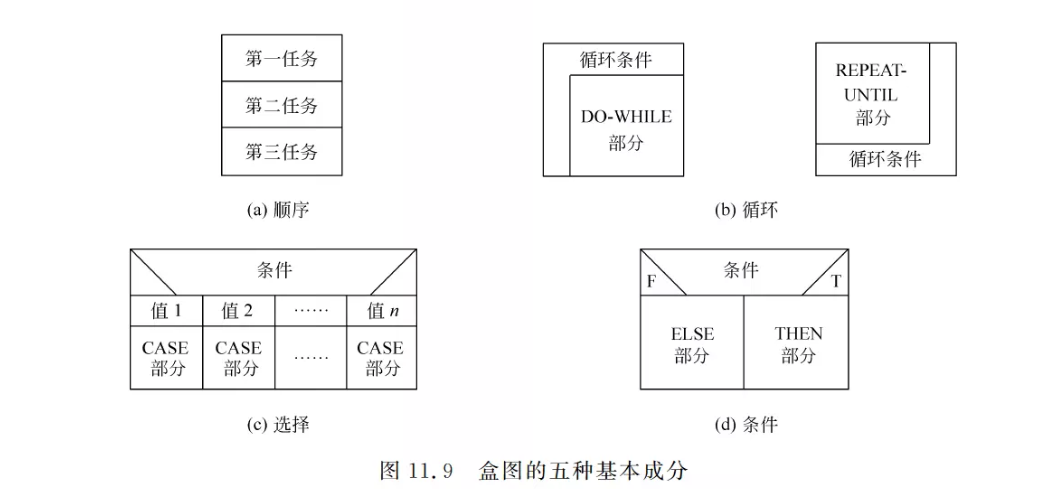
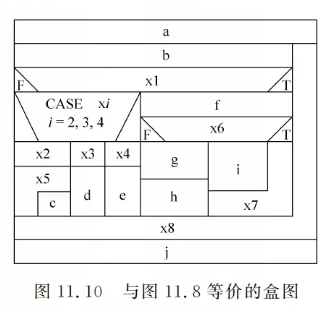
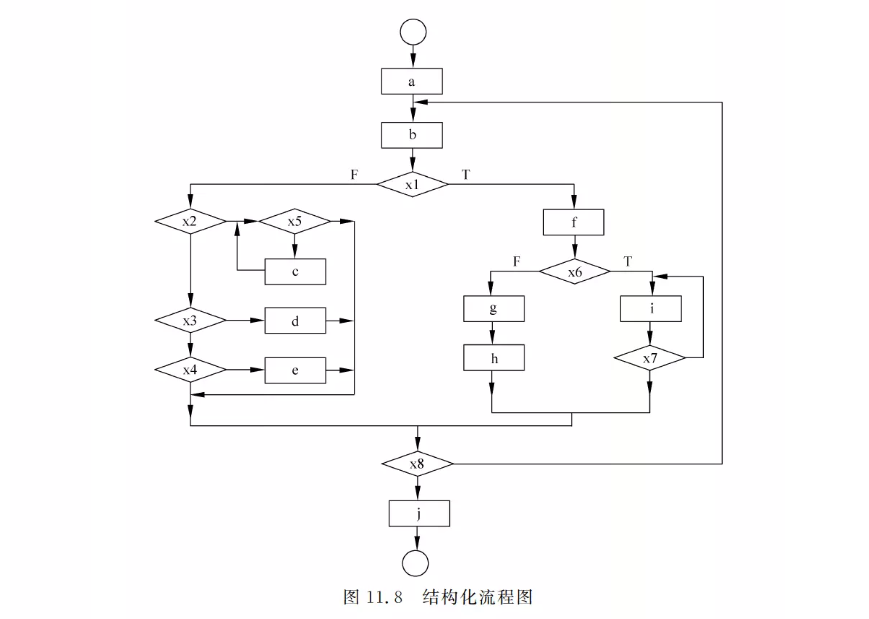
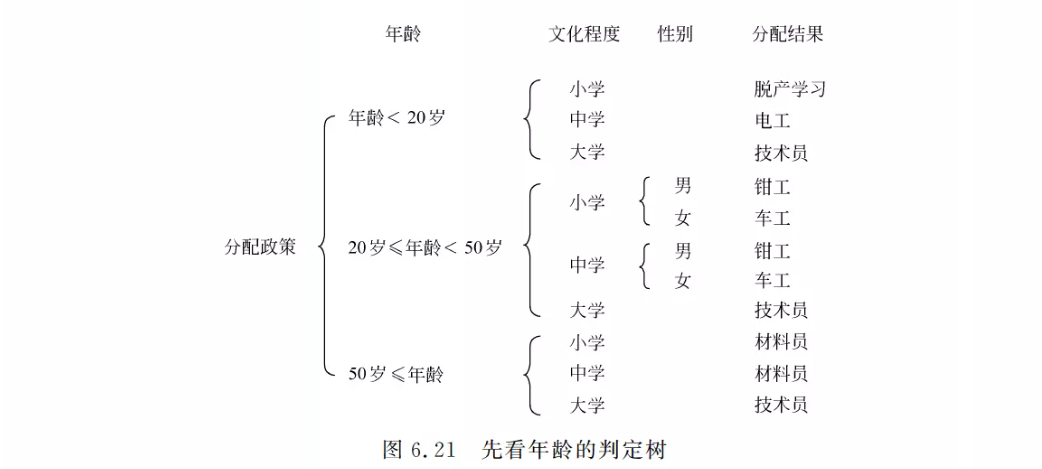
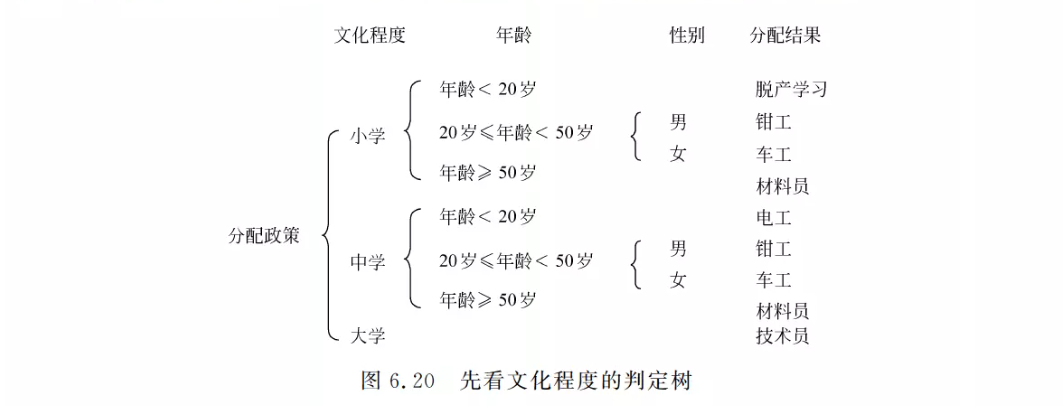


图11.8为一段代码的流程图，与下图的盒图相对应



* 判定树



* 判定表

在一些条件较多、在每个条件下取值也较多的判定问题，可以用判定表进行表示。其优点是能把各种组合情况完整的表示出来，有时还能帮助发现遗漏和矛盾情况。下面是一个关于工厂职工的判定表示例：

某工厂对一部分职工重新分配工作，分配原则:

1. 年龄小于20者，文化程度为小学则为脱产学习；文化程度是中学为电工。
2. 年龄在20—50之间，文化程度为小学或中学者，男性为钳工，女性为车工；文化程度是大学为技术员。
3. 年龄在50岁以上，文化程度是小学或中学者当材料员；文化程度为大学为技术员。

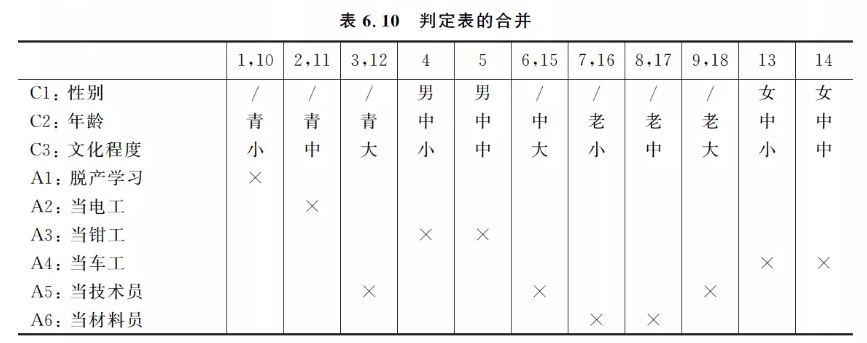
根据以上分析，共3个因素：性别、年龄、文化程度，排列组合共18种情况，而结果一种有六种：脱产学习、电工、钳工、车工、技术员、材料员，称之为六种行动。不同的条件组合，采取不同的行动。由此可得下表：



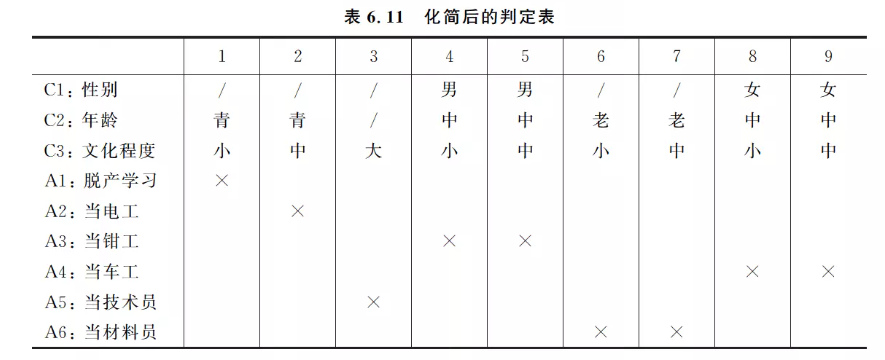
将不同条件组合对应的行动打上标记“×”，可得下表：



上表将所有组合情况都列出来，因此不会有遗漏，但过于繁琐，可以通过下列原则进行合并：在相同行动的情况下，检查它对应的各列条件组合种是否存在无需判断的条件。例如：第一列与第十列，对应的行动都是A1：脱产学习，对应的C2：年龄值取相同，C3：文化程度取值也相同，仅条件C1:性别取值不同。将性别合并后可得：



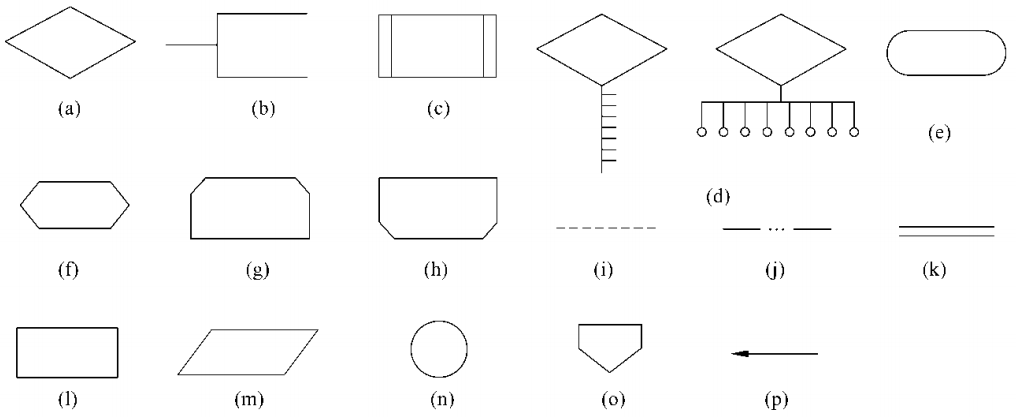
将年龄和学历也进行合并可得最简判定表：



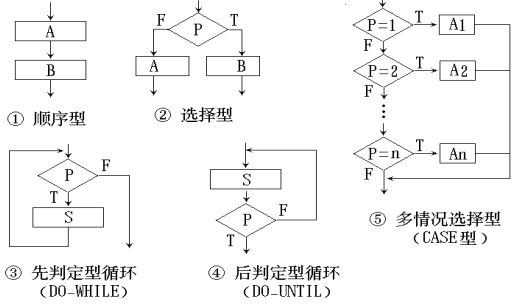
1. 给出判定表的步骤如下所示：
2. 分析决策问题涉及的几个条件；
3. 分析每个条件取值的集合；
4. 列出条件的各种可能组合；
5. 分析决策问题涉及几个可能的行动；
6. 做出有条件组合的判定表；
7. 决定各种条件组合的行动；
8. 按合并规则简化判定表。

* 程序流程图

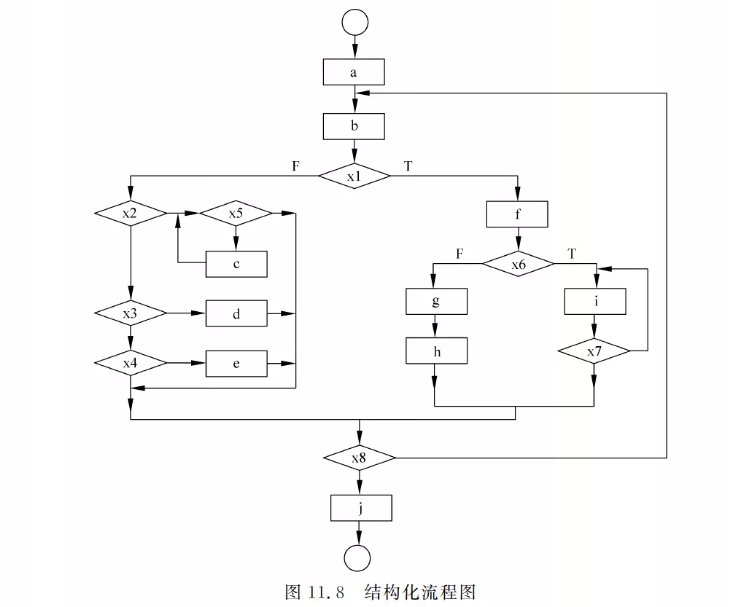
程序流程图即程序框图，是历史最为悠久，流行最广的一种图形表示方法。程序流程图中使用的符号有：(a)选择(分支)； (b)注释； (c)预先定义的处理； (d) 多分支；(e) 开始或停止；(f) 准备；(g)循环上界限；(h) 循环下界限； (i) 虚线； (j) 省略符； (k) 并行方式； (l) 处理； (m) 输入输出； (n) 连接； (o) 换页连接； (p) 控制流



五种基本控制结构



使用上述标准结构反复嵌套而形成的流程图称为结构化流程图如下图所示：



* 界面图

人机交互是指人通过屏幕、键盘等设备与计算机进行信息交换，包括输入与输出，用户界面设计即人机交互设计，其基本原则是为用户操作着想，而不应从设计人员设计方便考虑，其主要包括一下几点：

1. 对话要清楚、简单、用词要符合用户观点和习惯；
2. 对话要使用不同操作水平的用户，便于维护和修改；
3. 错误信息设计要有建议性，用词应当友善，简洁清楚，并尽可能告知使用者产生错误的可能原因；
4. 关键操作要有强调警告，对某些要害操作，无论操作人员是否有误操作，系统应该进一步确认。

图形用户界面的基本元素有下拉菜单、弹出菜单、列表框/组合列表框、单选钮、复选框、文本输入框、命令按钮、滚动条、表格等。在使用这些元素时需要注意以下原则：

（**一致性**）

1. 用户界面的各个画面设计在整体上应保持相同或相似的外观。如按钮和选择项的位置尽量安排在同样的地方，便于用户熟练掌握屏幕上的信息。
2. 用户界面使用的词汇、图示、颜色、选取方式、交流顺序，其意义与效果应前后一致。
3. 允许纯键盘输入方式，输入的移动顺序从左到右、从上往下。

（**可视化**）

1. 正确使用图形的表达能力。纯文字表达效果枯燥，形式单一，而滥用图形表示有时会造成画面混乱、反而使用户不易了解。
2. 由于图形对象占用系统资源较多，处理速度慢，因此在时间响应要求高，而硬件资源档次较低的环境中，不宜采用图像界面。

（**直接映射**）

（**有效反馈**）

同时，根据用例事件流或数据处理过程的描述，也需要对用户界面加以设计，主要包括以下人机交互的必要内容：

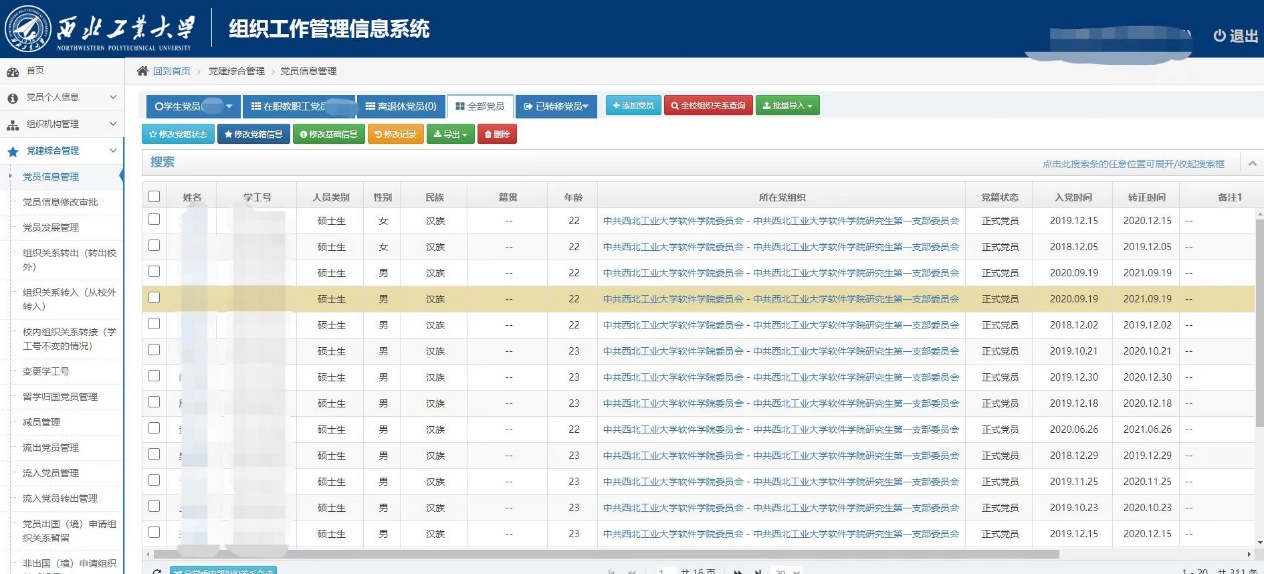
1. 界面窗口的整体布局和界面元素；
2. 需要由系统执行的操作按钮及响应；
3. 需要由用户输入到系统中的数据项。数据项的输入方式和格式要求，以及需要由系统执行的数据校验；
4. 对界面事件系统需要及时做出回应的说明；
5. 需要由系统输出给用户的数据窗口或消息。



日历选择界面



用户登录界面



用户信息管理界面

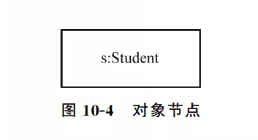
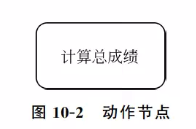
* 活动图

UML 活动图用于业务流程建模或者用例中的场景逻辑模型。例如，用活动图建模财务报销流程，或者建模算法过程。活动图是流程的可视表示，强调流程中活动的顺序和条件。流程就是“先做什么，接着做什么，最后做什么”。业务流程是由不同的人协同完成的一系列活动，活动之间有严格的先后顺序。

活动图元素分为以下几种：

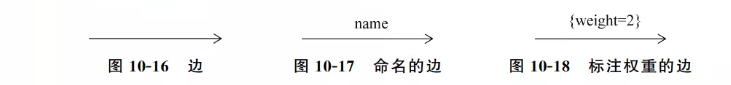
1. 动作节点和对象节点

动作节点用圆角矩形表示，对象节点用直角矩形框表示：



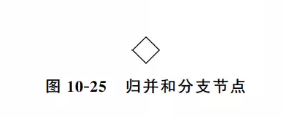
1. 边

活动图中的边的标记使带箭头的实线，边上可以标注名字以及权重，如下所示：



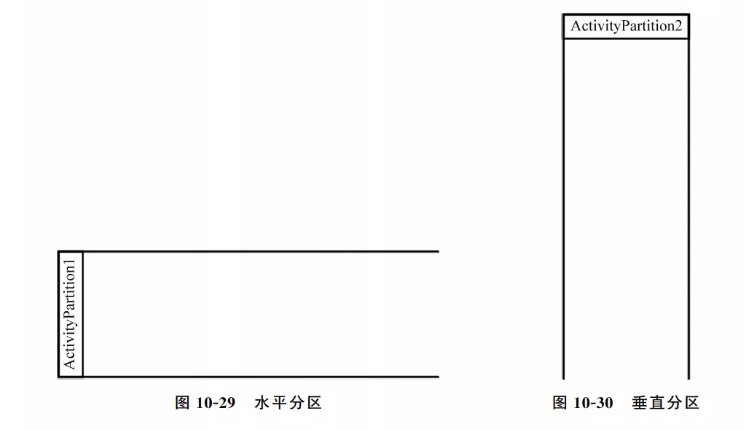
1. 控制节点

控制节点分为开始节点、结束节点、分叉节点、汇合节点、分支节点以及归并节点。

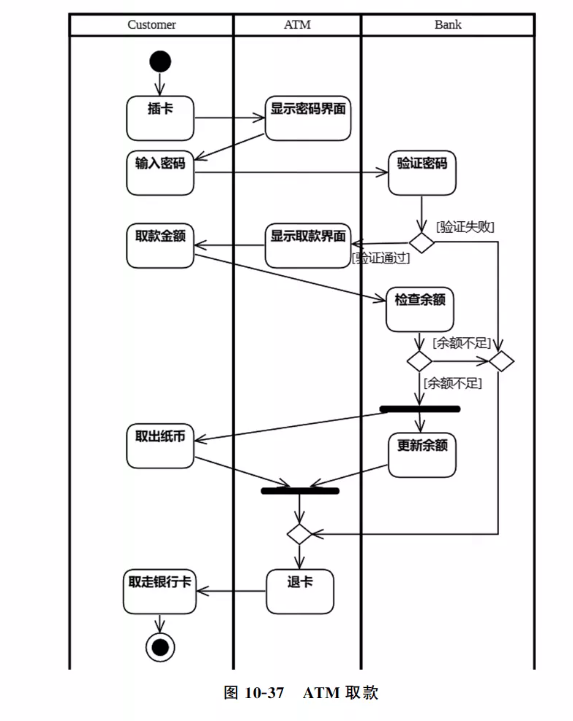


1. 分区（泳道）

分区的记号使两条并行直线，一端被矩形框封闭，分区的名字标记在矩形框内。分区可以是水平或者垂直的。



案例：银行自动柜员机ATM取款的活动图，该流程设计客户（Customer）、ATM和银行计算机系统（BANK）三个活动的执行角色，每个角色的活动分组到一个泳道中：

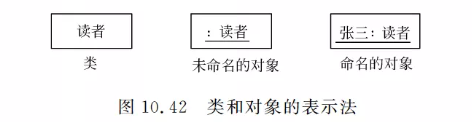


* 时序图

时序图，也叫顺序图，用来描述设计用例实现的多个对象实例以及对象交互时传递的消息，并按照用例的执行步骤为顺序指明对象的交互顺序。顺序图的基本元素有对象、参与者、生命线、激活框、消息和消息路线。

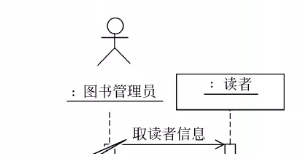
1. 对象

对象即是一个类的实例，其图像符号也与类相同，并带有一条叫做“生命线”的垂直虚线。生命线上的矩形条表明对象在特定时间的生存周期。



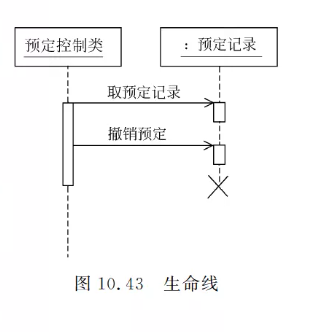
1. 参与者

参与者作为一个交互过程的发起者，通常在顺序图中的第一条生命线加以表示。



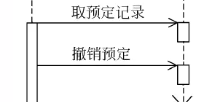
1. 生命线和激活框

对象下方的虚线即为生命线，其代表了对象的生命周期。当对象被销毁，其生命线就会中断。生命线上的矩形框（激活框）表明交互中对象何时起作用，在顺序图中，激活框非必要。



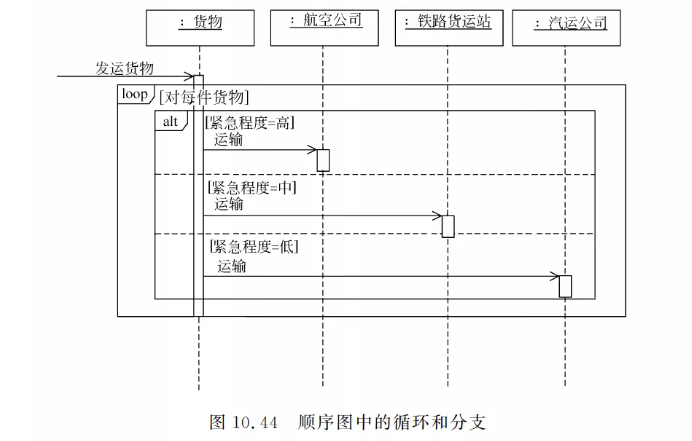
1. 消息

消息是对象之间的通信，消息传递的同时对应活动随之发生。其在顺序图中的表示形式为从一个对象的生命线到另一个对象生命线的水平箭头。消息像一个后可能会回送结果，采用反向的虚线和箭头表示。

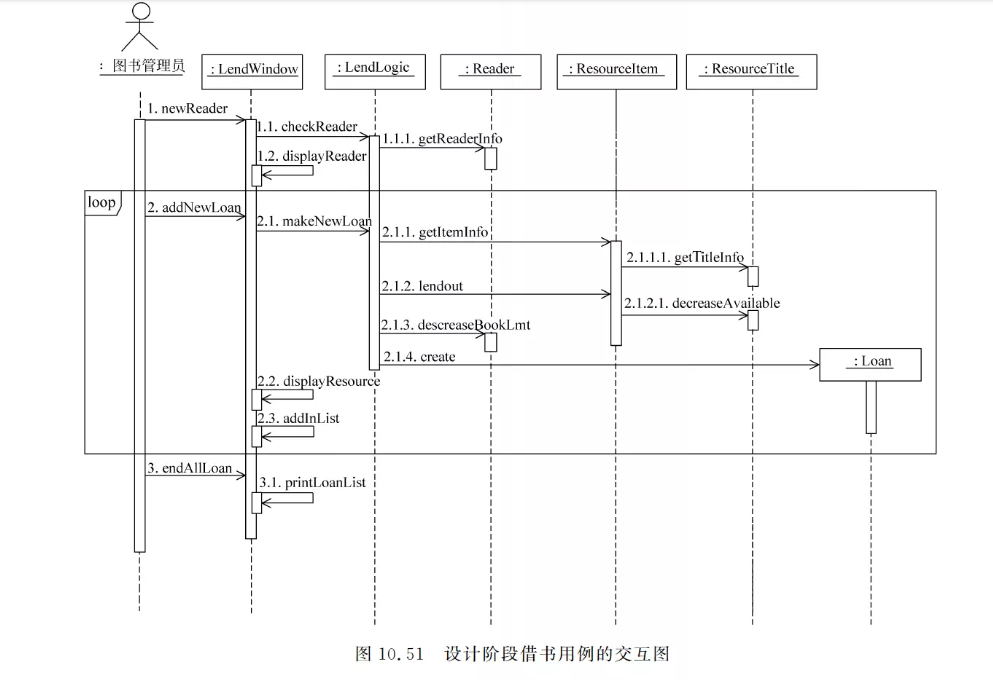


1. 循环和条件

顺序图中的控制流可用交互框架来表示，该框架可以将顺序图中的某个区域框起，并划分为若干片段，并对每个框架加以标识符进行标注，循环操作使用loop标识符，条件操作使用alt标识符：



下面是一个借书顺序图示例：

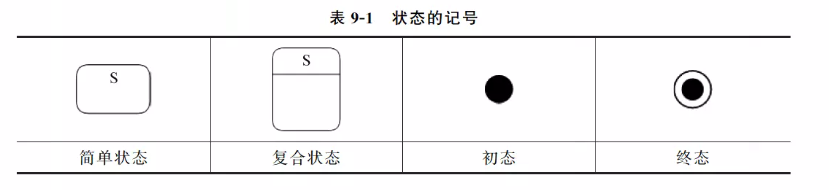


* 状态图

状态图又称状态机图，是状态机的可视表示。

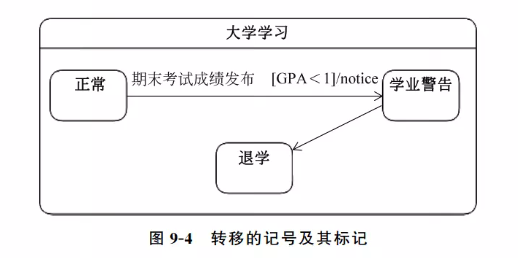
1. 状态

状态由初态、终态和中间状态组成。初态只能有一个；而终态可以是0个、1个活多个，中间状态可以是简单状态或复合状态。



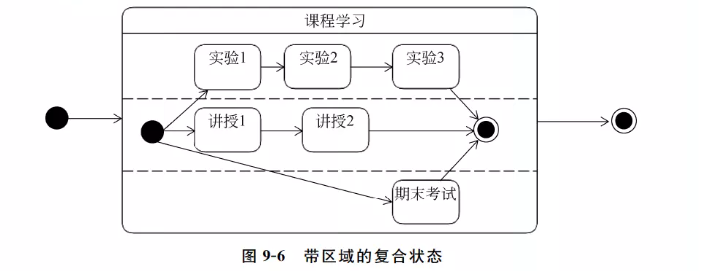
1. 转移

转移用带箭头的实线进行表示，实线上使用文本描述转移。在这样一个转移上可以定义0个或多个触发器用以判定是否转移。

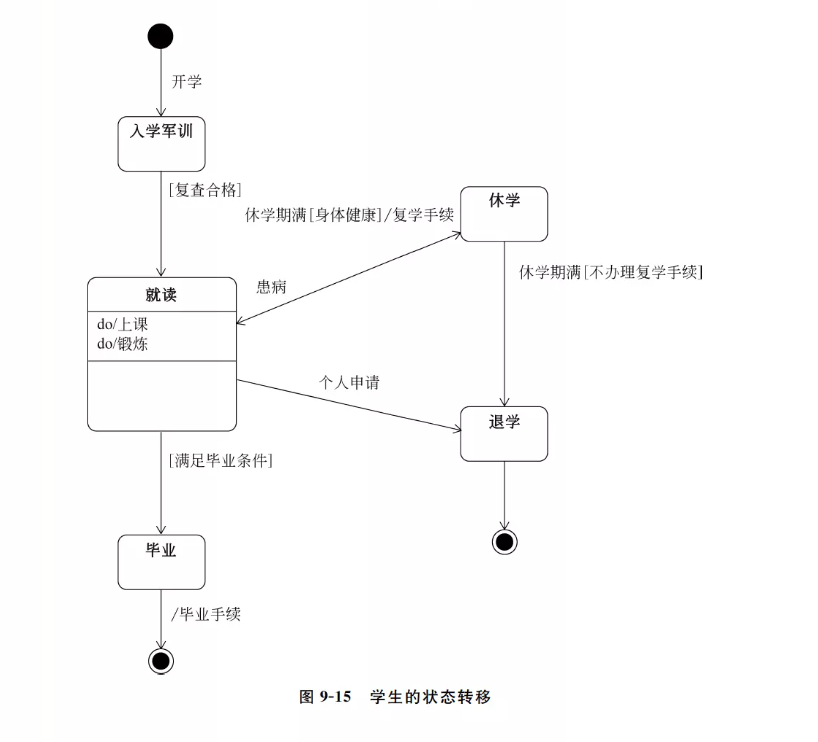


1. 区域

复合状态中至少有一个区域，具有多个区域的符合复合状态使用虚线进行区域划分。



下面是一个关于学生入学、休学、退学的状态图示例：

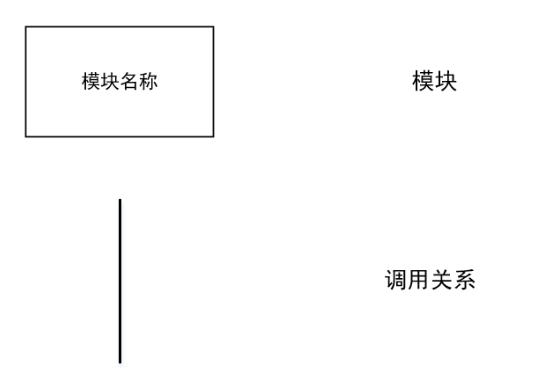


* HIPO图

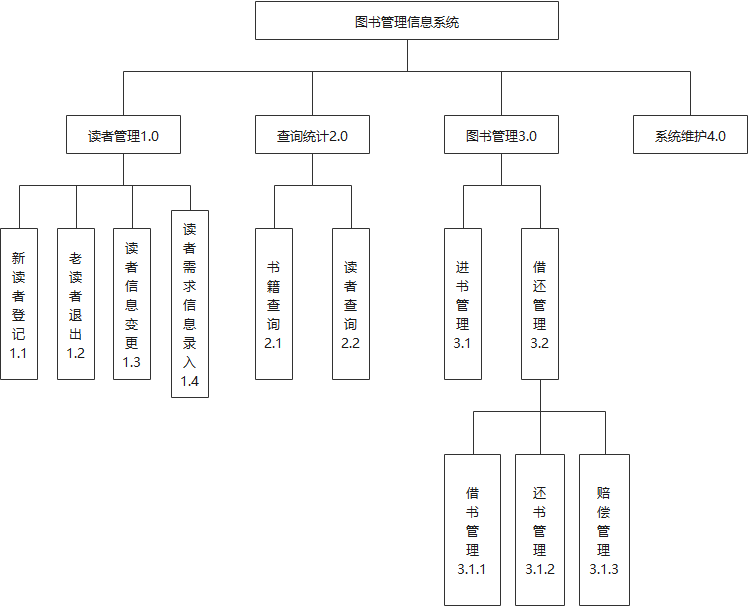
HIPO图是[IBM](https://baike.baidu.com/item/IBM/9190?fromModule=lemma_inlink)公司于70年代中期在层次结构图(structure chart)的基础上推出的一种描述系统结构和模块内部处理功能的工具(技术)。HIPO图由层次结构图和[IPO图](https://baike.baidu.com/item/IPO%E5%9B%BE/9866101?fromModule=lemma_inlink)两部分构成,前者描述了整个系统的设计结构以及各类模块之间的关系,后者描述了某个特定模块内部的处理过程和输入/输出关系。

1. 层次图

层次图又叫H图，用于描绘软件的层次结构，适用于自顶向下设计软件的结构描述。主要包括模块和调用关系两种元素：



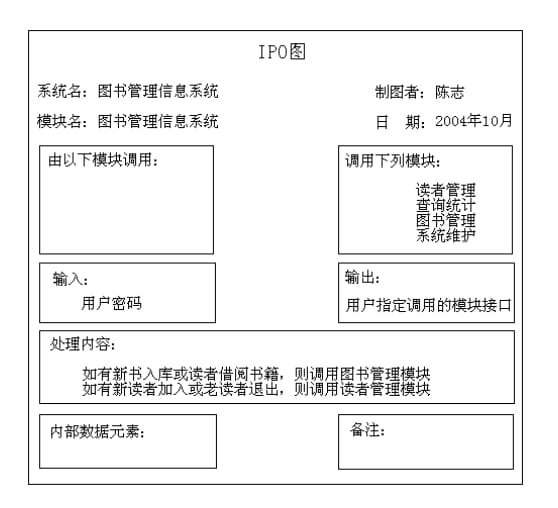
为了能使HIPO图具有可追踪性，在H图(Hierarchy， 层次图)里除了最顶层的方框之外，每个方框都加了编号，一个图书管理系统的H图如下所示：



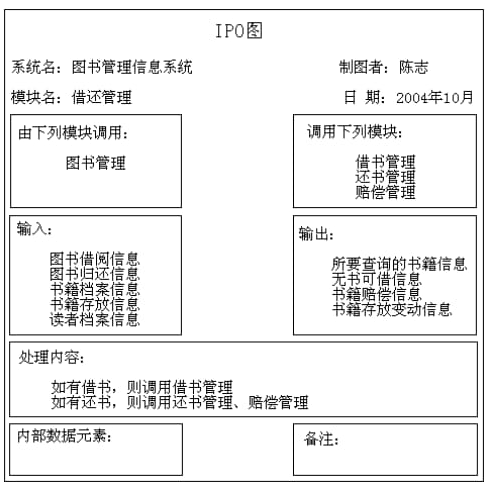
1. IPO图

IPO图是一张图形化的表格，是对H图中的每一个模块的输入与输出关系、处理内容、本模块之间的调用关系的描述。

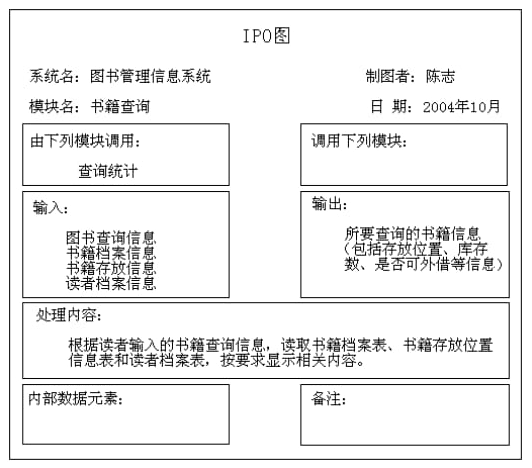
图书管理信息系统模块IPO图：



借还管理模块IPO图：



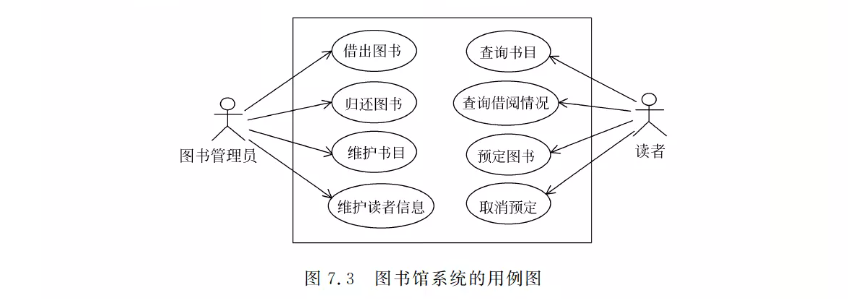
书籍查询IPO图：

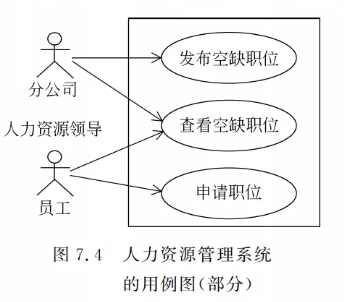


* 用例图：

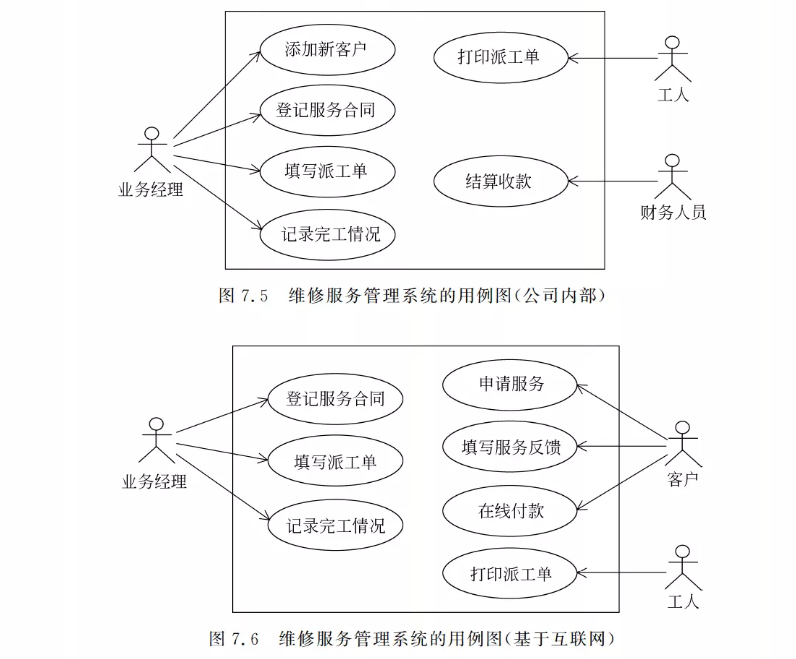
用例就是需求，通过用例名称可以表达系统要完成的工作。

图书管理系统用例图：



人力资源管理系统用例图：

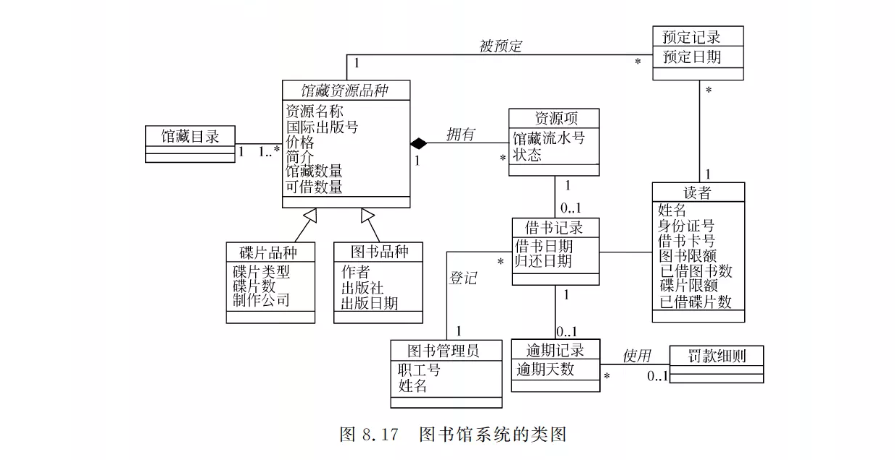
维修服务管理系统用例图：



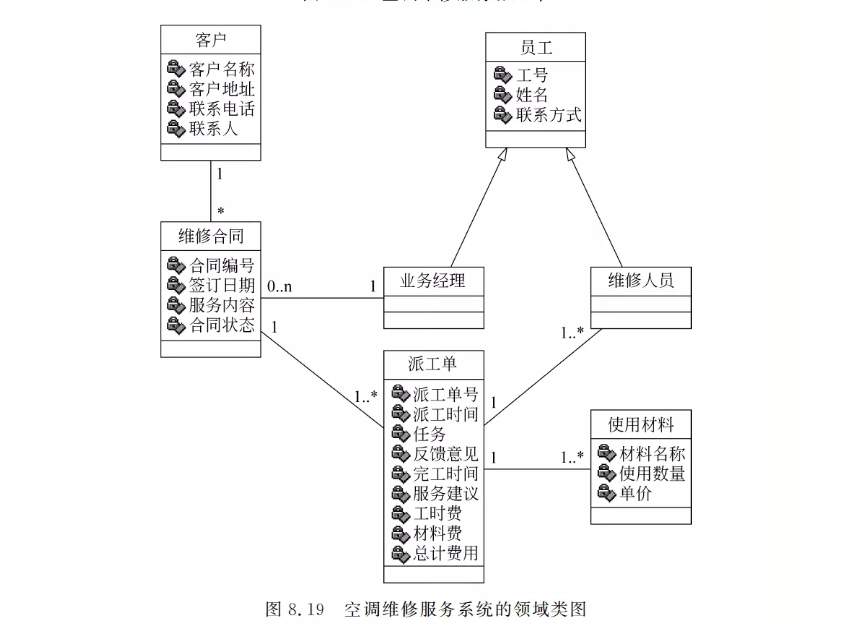
* 类图：

识别出全部概念类后,通过添加属性、添加关联,以及泛化处理后形成领域模型，采用UML类图来表示，它们是构成用例实现的基础条件。这是分析过程中最重要的模型。

图8.17是图书馆系统的完整领域类图：



空调维修服务系统的领域类图：



* 包图：

包图用来描述包及其依赖关系。当有用于表现层的界面包和用于领域层的包时，表现层包中的类因为要使用领域包中的领域类提供的服务,所以表示包对领域包有依赖关系，图10.5是图书馆系统的一个包图例子。

