

讲师介绍

邹月平 51CTO学院微职位讲师

全国计算机技术与软件专业技术资格考试辅导用书编委会委员，电子工业出版社多次重印的书籍《信息系统项目管理师历年真题解析（第3版）》副主编，《系统集成项目管理工程师历年真题解析（第3版）》副主编，《软件设计历年真题解析》副主编、《软件设计历年真题解析》副主编、《系统分析师历年真题解析》等书籍，主要讲授软考历年真题解析、计算机技术知识、项目管理知识等。

51CTO学院

系统设计

51CTO学院

- 系统设计是系统分析的延伸与拓展。系统分析阶段解决“做什么”的问题，而系统设计阶段解决“怎么做”的问题。同时，它也是系统实施的基础，为系统实施工作做好铺垫。合理的系统设计方案既可以保证系统的质量，也可以提高开发效率，确保系统实施工作的顺利进行。
- 系统设计阶段又称为物理设计阶段，它是信息系统开发过程中一个非常重要的阶段。其任务是根据系统规格说明书中规定的功能要求，考虑实际条件，具体设计实现逻辑模型的技术方案，也就是设计新系统的物理模型，为下一阶段的系统实施工作奠定基础。

- 系统设计的主要内容包括**概要设计**和**详细设计**。概要设计又称为系统总体结构设计，它是系统开发过程中很关键的一步，其主要任务是将系统的功能需求分配给软件模块，确定每个模块的功能和调用关系，形成软件的模块结构图，即系统结构图。在概要设计中，将系统开发的总任务分解成许多个基本的、具体的任务，为每个具体任务选择适当的技术手段和处理方法的过程称为详细设计。根据任务的不同，详细设计又可分为多种，例如，网络设计、代码设计、输入/输出设计、处理流程设计、数据存储设计、用户界面设计、安全性和可靠性设计等。

- **1. 网络设计**
- **网络设计的主要任务是，根据系统的要求选择网络结构，按照系统结构的划分，安排网络 and 设备的分布，然后根据物理位置考虑网络布线和设备的部署，还要根据实际业务的要求划定各网络节点的权限、级别和管理方式等，选择相应的系统软件和管理软件。**
- **2. 代码设计**
- **代码是用数字或字符来表示各种客观实体。在系统开发过程中，进行代码设计的主要目的是确保代码的唯一化、规范化和系统化。在进行代码设计时，首先需要考虑系统的编码问题，编码问题的关键在于分类，有了一个科学的分类方式，系统要建立编码规范就相对较为容易。**

- **3. 输入/输出设计**
- **输入设计的目的是确保向系统输入的数据的完整性、正确性和一致性，其主要内容包括确定输入数据的内容、输入方式设计、输入格式设计和检验方式的设计；输出设计的目的是确保系统输出数据的完整性、正确性和一致性，其主要内容包括确定输出的内容、选择输出设备与介质，以及确定输出格式等。**
-

- **4. 处理流程设计**

- **处理流程设计是系统详细设计的重要组成部分，它的主要目的是确定各个系统模块的内部结构，即内部执行过程，包括局部数据组织和控制流，以及每个具体加工过程和实施细节。**

- **5. 数据存储设计**

- **数据存储设计主要是根据数据处理要求、处理方式、存储的信息量大小、数据使用的频率和所能提供的设备条件等，选择数据存储的方式、存储介质、数据组织方式和记录格式，并估算数据的容量。一个好的数据存储设计应该充分体现系统的业务流程，充分满足组织的各级管理要求。同时，还应该使得后续的系统开发工作方便、快捷，系统开销小，且易于管理和维护。**

- **输入设计需要遵循以下原则：**
- **（1）输入数据最少原则。**在满足需求的前提下尽量提供较少的数据输入，数据的输入量越少，出错的几率越低，花费的时间也越少。
- **（2）简单性原则。**输入过程应尽量简单，如性别、出生日期等数据设计为选择项，一方面方便用户的使用，节省输入时间，同时可以降低出错的可能性。
- **（3）尽早验证原则。**对输入数据的检验尽量接近数据的输入点，及时发现输入中存在的错误，以便能够尽早进行改正。
- **（4）少转换原则。**输入数据尽量采用原始的数据格式，避免在数据转换过程中发生错误。

- **6. 用户界面设计**
- **界面是系统与用户交互的最直接的层面，界面的好坏决定用户对系统的第一印象，而设计优良的界面能够引导用户自己完成相应的操作，起到向导的作用。**
- **良好的用户界面设计需要遵循如下一些基本原则：**
- **（1）置于用户控制之下。在定义人机交互方式时，不强迫用户采用不是必须的或者不情愿的方式来进行操作，允许交互的中断和撤销。当用户操作技能等级提高时，可以实现流水化的交互方式，允许用户定制交互方式，以便使用户界面与内部技术细节隔离，允许用户和出现在屏幕上的对象直接进行交互。**

- **（2）减轻用户的记忆负担。尽量减轻对用户记忆的要求，创建有意义的缺省设置，定义一些符合用户直觉的访问途径，适当定义一些快捷方式，界面的视觉布局应该尽量与真实世界保持一致，并能够以不断扩展的方式呈现信息。用户可以快速学习并使用系统，提供尽量“傻瓜式”的操作界面，方便用户使用。界面中各个元素的名称应该易懂，用词准确，避免模棱两可的字眼，能够做到“望文知意”，理想的情况是用户不用查阅帮助，就能知道该界面元素的功能，并正确地进行相关操作。**

•

- (3) 保持界面一致性。用户应以一致的方式提供或获取信息，所有可视信息的组织需要按照统一的设计标准，在系列化的应用软件中需要保持一致性，用户已经很熟悉的一些界面交互模型不到万不得已时，不要随意进行修改。需要确保用户界面操作和使用的一致性，例如，所有窗口按钮的位置要一致、提示信息和界面元素的命名要一致、界面颜色和风格要一致等。用户界面的一致性可以使用户能够统一地对待系统的各个不同的功能界面，以及系列化的系统，从而降低培训和支持成本。
- 以上三条原则由著名用户界面设计专家Theo Mandel博士所创造，通常称之为人机交互的“**黄金三原则**”。另外，在设计用户界面时，还需要保证界面的合理性和独特性，有效进行组合，注重美观与协调；恰到好处地提供快捷方式，注意资源协调等。

- **7. 安全性和可靠性设计**
- **安全性和可靠性设计的目的是确保系统的安全性和可靠性，对系统的运行环境和数据处理进行有效的控制，保证系统安全、有效地运行。其主要内容包括系统运行环境安全性分析和控制，例如，对管理结构的组织、硬件和系统软件、自然环境等方面的分析与必要的监督和控制等；还包括对数据处理的控制，例如，输入内容和输入方式的控制、错误程序与异常处理等。**

典型真题

- 界面是系统与用户交互的最直接的层面。Theo Mandel博士在界面设计中，提出了著名的人机交互“黄金三原则”，包括保持界面一致、减轻用户的记忆负担和（ ）。
 - A.遵循用户认知理解
 - B.降低用户培训成本
 - C. 置于用户控制之下
 - D.注意资源协调方式

典型真题

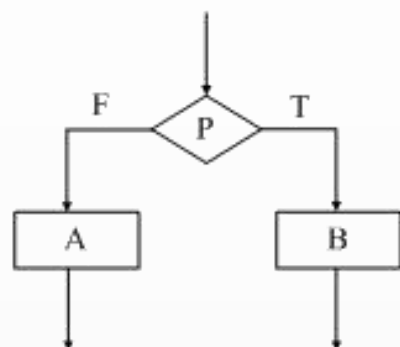
- 试题分析
- 人机交互“黄金三原则”包括：置于用户控制之下、减少用户的记忆负担、保持界面的一致性。
- 试题答案：C

流程设计

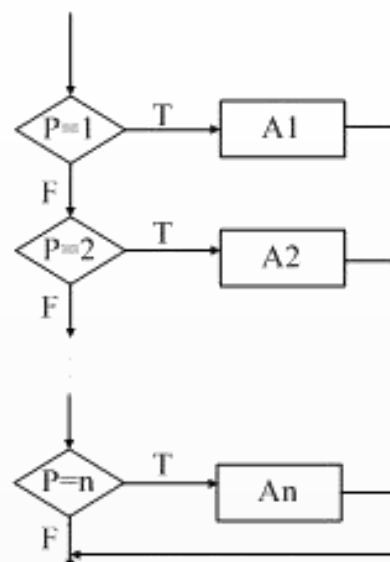
- 在处理流程设计过程中，为了更清晰地表达过程规则说明，陆续出现了一些用于表示处理流程的工具，这些工具包括三类，分别是图形工具、表格工具和语言工具。其中常见的图形工具包括程序流程图、IPO图、盒图、问题分析图、判定树，表格工具包括判定表，语言工具包括过程设计语言等。
- 1. 程序流程图



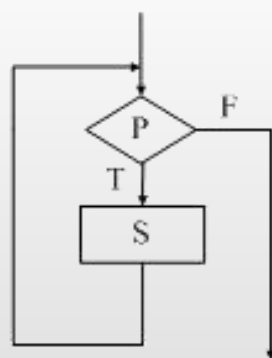
顺序型



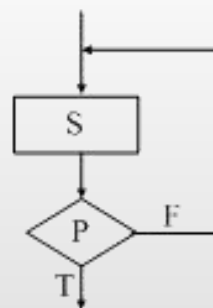
选择型



多分支选择型



当型循环
(DO-WHILE)



直到型循环
(DO-UNTIL)

- 2. IPO图
- IPO图是由IBM公司发起并逐步完善的一种流程描述工具。系统分析阶段产生的数据流图经转换和优化后形成的系统模块结构图的过程中将产生大量的模块，分析与设计人员应为每个模块写一份说明，即可用IPO图来对每个模块进行表述，IPO图用来描述每个模块的输入、输出和数据加工，其导致结构如图所示。

IPO图

系统名称: *****系统

设计人: *****

模块名称: 主控

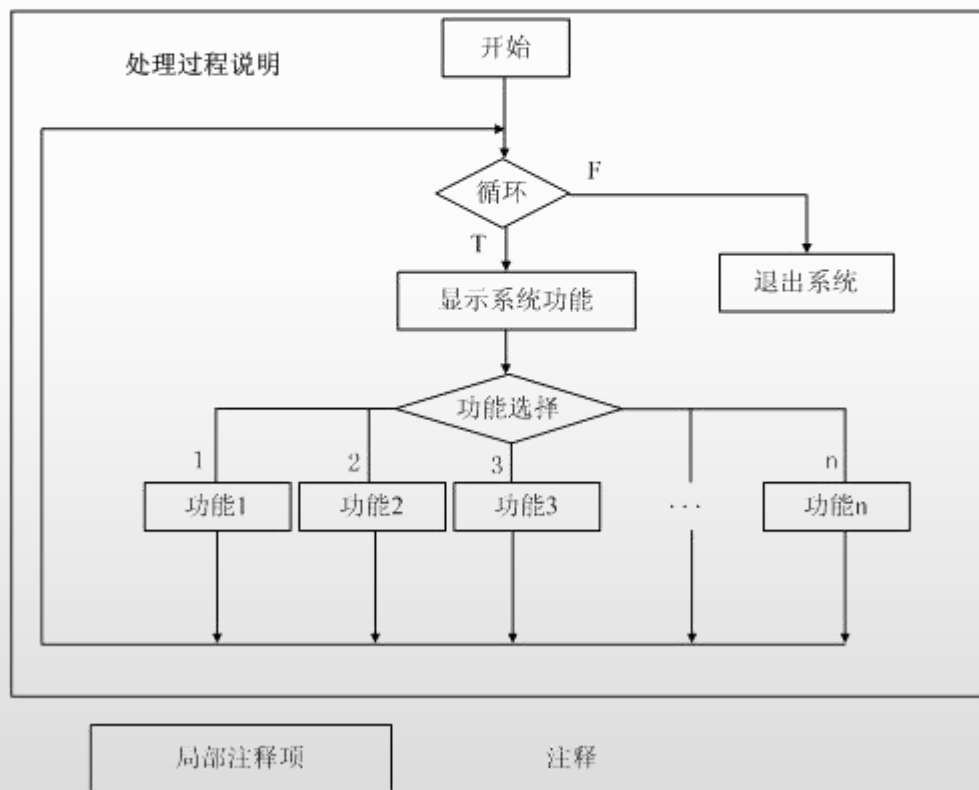
日期: ****年**月

上层调用模块

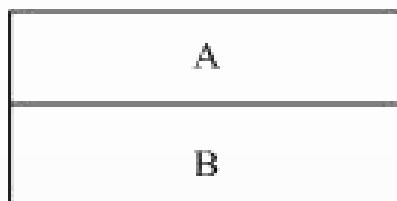
可调用的下层模块

输入

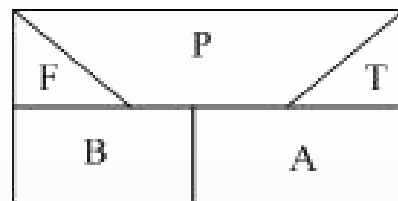
输出



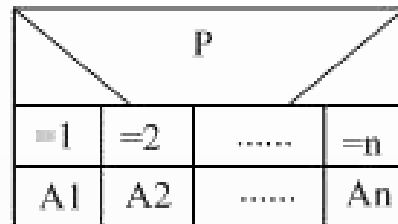
- 3. N-S图
- 为避免流程图在描述程序逻辑时的随意性与灵活性，美国学者 I.Nassi 和 B.Shneiderman 在 1973 年提出了用方框代替传统的 PFD，通常把这种图称为 N-S 图或盒图，与 PFD 类似，在 N-S 图中也包括五种控制结构，分别是顺序型、选择型、WHILE 循环型（当型循环）、UNTIL 循环型（直到型循环）和多分支选择型，任何一个 N-S 图都是这五种基本控制结构相互组合与嵌套的结果。



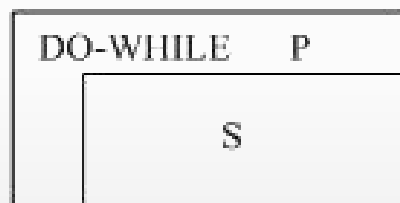
顺序型



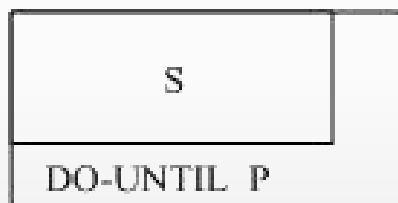
选择型



多分支选择型

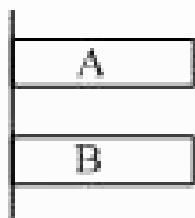


当型循环
(DO-WHILE)

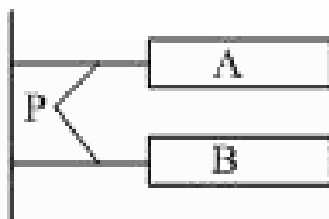


直到型循环
(DO-UNTIL)

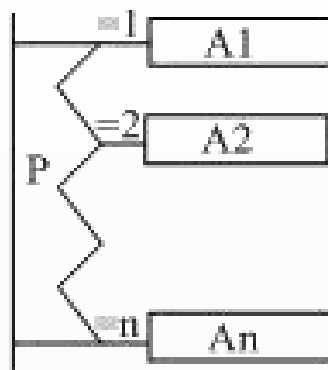
- 4. 问题分析图
- 问题分析图 (*Problem Analysis Diagram* , *PAD*) 是继PFD和N-S图之后, 又一种描述详细设计的工具, 它由日立公司于1979年提出, 也是一种支持结构化程序设计的图形工具。*PAD*也包含五种基本控制结构, 并允许递归使用。



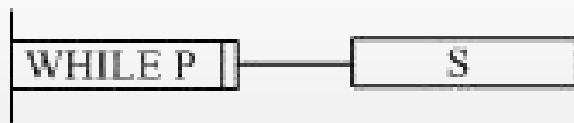
顺序型



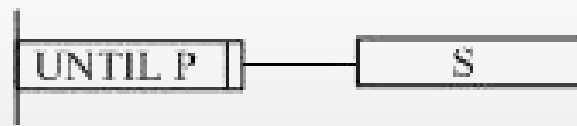
选择型



多分支选择型



当型循环
(DO-WHILE)



直到型循环
(DO-UNTIL)

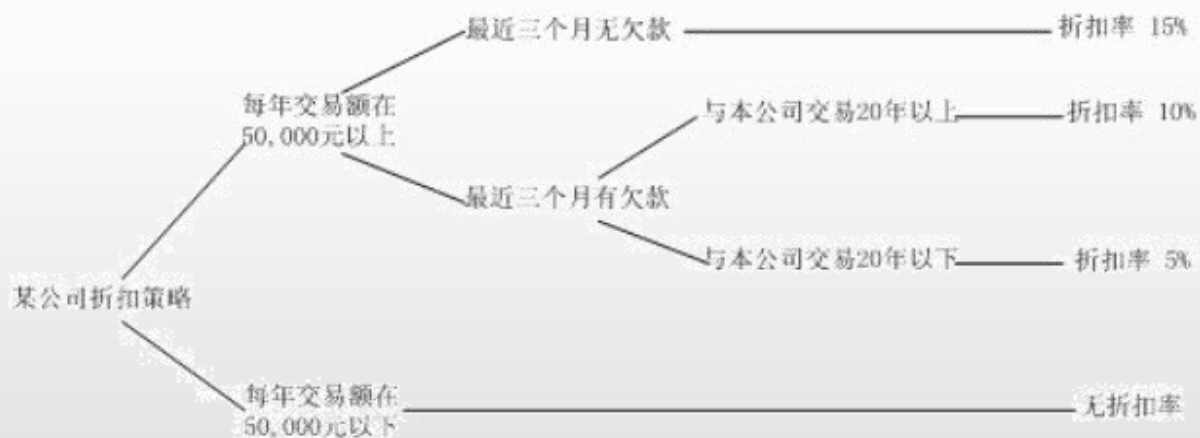
- 5. 过程设计语言
- 过程设计语言 (*Process Design Language* , *PDL*) 也称为结构化语言或伪代码 (*pseudo code*) , 它是一种混合语言 , 采用自然语言的词汇和结构化程序设计语言的语法 , 用于描述处理过程怎么做 , 类似于编程语言。

- 6. 判定表
- 对于具有多个互相联系的条件和可能产生多种结果的问题，用结构化语言描述则显得不够直观和紧凑，这时可以用以清楚、简明为特征的判定表（*decision table*）来描述。

不同条件组合 条件和行动	1	2	3	4	5	6	7	8
C1: 每年交易额在 50,000 以上	T	T	T	T	F	F	F	F
C2: 最近三个月无欠款	T	T	F	F	T	T	F	F
C3: 与本公司交易 20 年及以上	T	F	T	F	T	F	T	F
A1: 折扣率 15%	Y	Y						
A2: 折扣率 10%			Y					
A3: 折扣率 5%				Y				
A4: 无折扣率					Y	Y	Y	Y

- 7. 判定树

- 判定树 (*decision tree*) 也是用来表示逻辑判断问题的一种常用的图形工具，它用树来表达不同条件下的不同处理流程，比语言、表格的方式更为直观。判定树的左侧（称为树根）为加工名，中间是各种条件，所有的行动都列于最右侧。



结构化设计

- 结构化设计 (*Structured Design* , *SD*) 是一种面向数据流的方法，它以SRS和SA阶段所产生的数据流图和数据字典等文档为基础，是一个自顶向下、逐步求精和模块化的过程。*SD*方法的基本思想是将软件设计成由相对独立且具有单一功能的模块组成的结构，分为概要设计和详细设计两个阶段，其中概要设计的主要任务是确定软件系统的结构，对系统进行模块划分，确定每个模块的功能、接口和模块之间的调用关系；详细设计的主要任务是为每个模块设计实现的细节。

- 人们在解决复杂问题时使用的一个很重要的原则，就是将它分解成多个小问题分别处理，在处理过程中，需要根据系统总体要求，协调各业务部门的关系。在SD中，这种功能分解就是将系统划分为模块，**模块是组成系统的基本单位**，它的特点是可以自由组合、分解和变换，系统中任何一个处理功能都可以看成一个模块。

- 一个模块应具备以下**四个要素**：
- (1)输入和输出。模块的输入来源和输出去向都是同一个调用者，即一个模块从调用者那儿取得输入，进行加工后再把输出返回调用者。
- (2)处理功能。指模块把输入转换成输出所做的工作。
- (3)内部数据。指仅供该模块本身引用的数据。
- (4)程序代码。指用来实现模块功能的程序。前两个要素是模块的外部特性，即反映了模块的外貌；后两个要素是模块的内部特性。在结构化设计中，主要考虑的是模块的外部特性，其内部特性只做必要了解，具体的实现将在系统实施阶段完成。
-

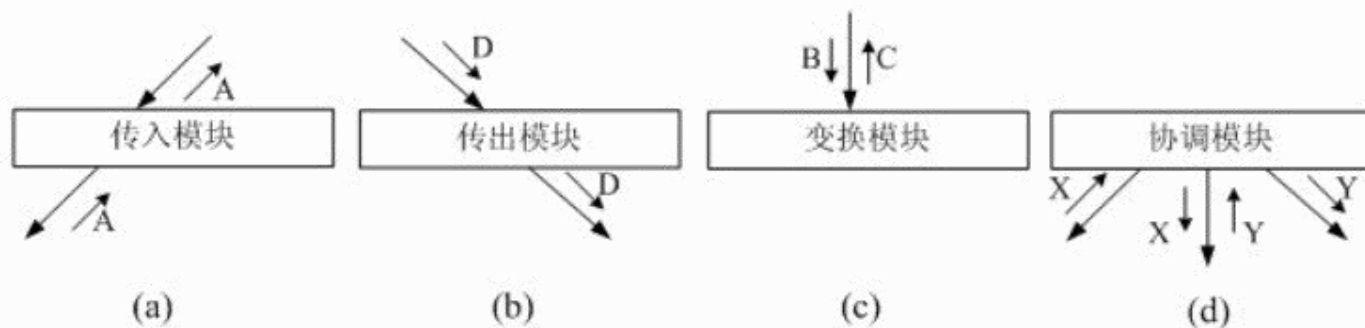
- 在SD方法中，系统由多个逻辑上相对独立的模块组成，在模块划分时需要遵循如下原则：
- （1）模块的大小要适中。系统分解时需要考虑模块的规模，过大的模块可能导致系统分解不充分，其内部可能包括不同类型的功能，需要进一步划分，尽量使得各个模块的功能单一；过小的模块将导致系统的复杂度增加，模块之间的调用过于频繁，反而降低了模块的独立性。一般来说，一个模块的大小使其实现代码在1~2页纸之内，或者其实现代码行数在50~200行之间，这种规模的模块易于实现和维护。

•

- **(2) 模块的扇入和扇出要合理。一个模块的扇出是指该模块直接调用的下级模块的个数；扇出大表示模块的复杂度高，需要控制和协调过多的下级模块。扇出过大一般是因为缺乏中间层次，应该适当增加中间层次的控制模块；扇出太小时可以把下级模块进一步分解成若干个子功能模块，或者合并到它的上级模块中去。一个模块的扇入是指直接调用该模块的上级模块的个数；扇入大表示模块的复用程度高。设计良好的软件结构通常顶层扇出比较大，中间扇出较少，底层模块则有大扇入。一般来说，系统的平均扇入和扇出系数为3或4，不应该超过7，否则会增大出错的概率。**

- **(3) 深度和宽度适当。深度表示软件结构中模块的层数，如果层数过多，则应考虑是否有些模块设计过于简单，看能否适当合并。宽度是软件结构中同一个层次上的模块总数的最大值，一般说来，宽度越大系统越复杂，对宽度影响最大的因素是模块的扇出。在系统设计时，需要权衡系统的深度和宽度，尽量降低系统的复杂性，减少实施过程的难度，提高开发和维护的效率。**

- 一般而言，在SC中存在**四种类型的模块**。



- (1) 传入模块。传入模块从下属模块中获取数据，经过某些处理，再将其传送给上级模块。
- (2) 传出模块。传出模块从上级模块中获取数据，进行某些处理，再将其传送给下属模块。

- **（3）变换模块。**变换模块也称为加工模块，它从上级模块获取数据，进行特定的处理，然后转换成其他形式，再传送回上级模块，大多数计算模块（原子模块）都属于这一类。
- **（4）协调模块。**协调模块是对所有下属模块进行协调和管理的模块。在系统的I/O部分或数据加工部分可以找到这样的模块，在一个好的SC中，协调模块应在较高层出现。
- **在实际系统中，有些模块属于上述某一种类型，也有一些模块是上述几种类型的组合。**

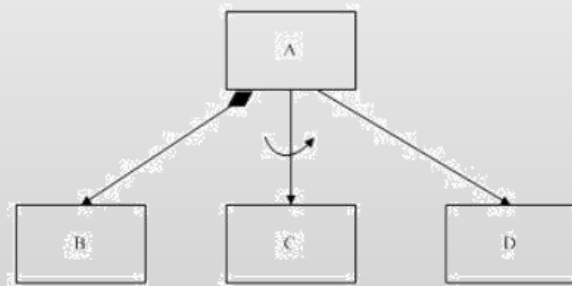
- **系统结构图** (Structure Chart , SC) 又称为模块结构图 , 它是软件概要设计阶段的工具 , 反映系统的功能实现和模块之间的联系与通信 , 包括各模块之间的层次结构 , 即反映了系统的总体结构。
- SC包括模块、模块之间的调用关系、模块之间的通信和辅助控制符号等四个部分。
- (1) 模块。在SC中 , 模块用矩形框表示 , 框中标注模块的名字 , 对于已定义或者已开发的模块 , 可以用双纵边矩形框表示。



- (2) 模块之间的调用关系。绘制方法是两个模块一上一下布局，以箭头相连，上面的模块是调用模块，箭头指向的模块是被调用的模块。“在线选课”模块调用“检索课程”模块，通常，箭头表示的连线可以用直线代替。

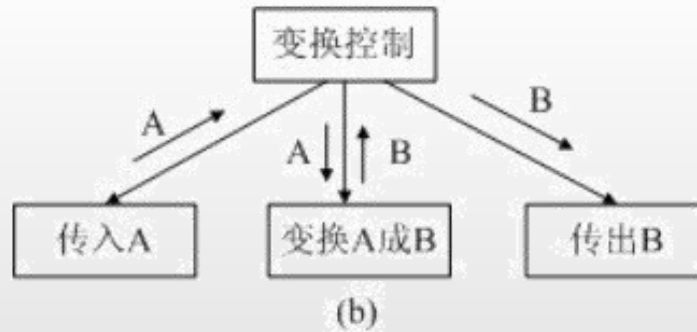
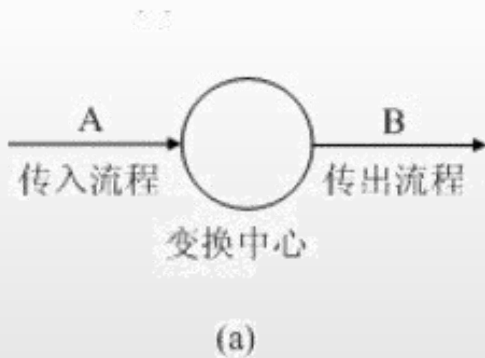


- (3) 模块之间的通信。模块间的通信以表示调用关系的长箭头旁边的短箭头表示，短箭头的方向和名字分别表示调用模块和被调用模块之间信息的传递方向和内容。例如，“在线选课”模块将信息“课程名、学期”传给“检索课程”模块，经加工处理后，“检索课程”模块将信息“课程号、课程名、已选修人数、教师姓名”等再回传给“在线选课”模块。
- (4) 辅助控制符号。当模块A有条件地调用模块B时，在箭头的起点标以菱形，模块A反复调用模块C时，在调用关系的连线上增加一个环状的箭头，如图13-10所示。在SC中，条件调用时所依赖的判断条件和循环调用时所依赖的控制条件通常都无须注明。

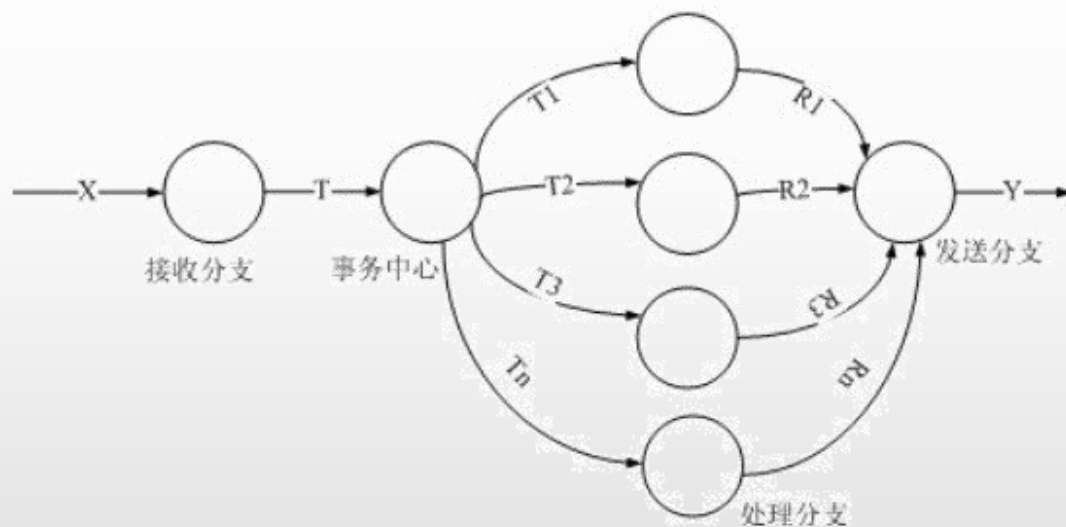


- **变换型SC**

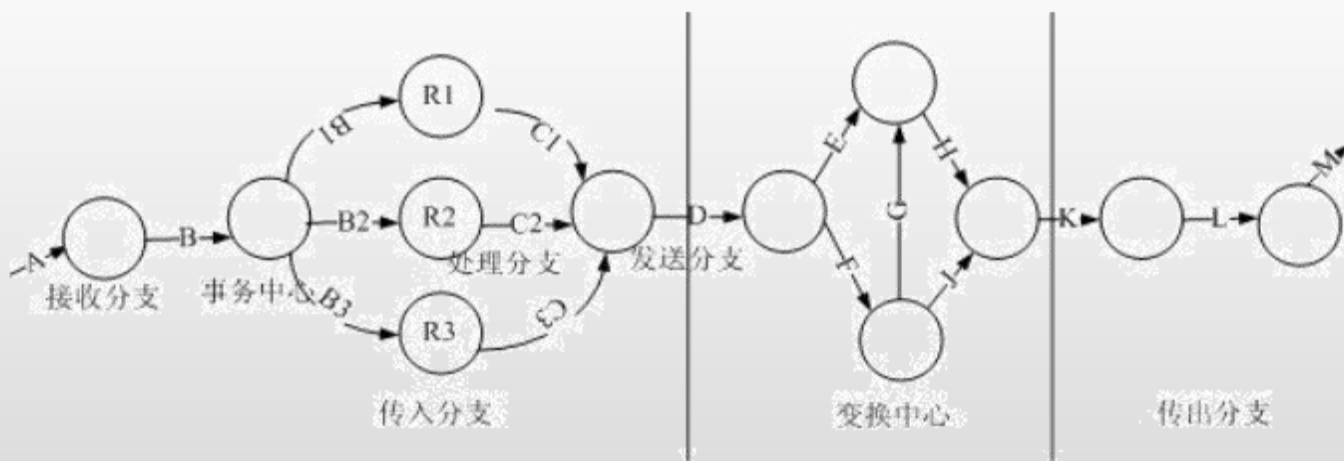
- 信息沿着输入通道进入系统，然后通过变换中心（也称为主加工）处理，再沿着输出通道离开系统，具有这一特性的信息流称为变换流。具有变换流型的SC可明显地分成输入、变换（主加工）和输出三大部分，它的功能是将输入的数据经过加工后输出。



- **事务型SC**
- 信息沿着输入通道到达一个事务中心，事务中心根据输入信息（即事务）的类型在若干个动作序列（称为活动流）中选择一个来执行，这种信息流称为事务流。



- **混合型SC**
- 在规模较大的系统中，其DFD往往是变换型和事务型的混合结构。此时，可把变换分析和事务分析应用在同一DFD的不同部分。
- 可以以变换分析为主，事务分析为辅进行设计。先找出主处理，设计出结构图的上层，然后根据DFD各部分的结构特点，适当选用变换分析或事务分析就可得出SC的某个初始化方案。



技术成就梦想