

# 软件开发中的SA方法和SD方法

区进明  
(广西计算中心)

## 提 要

本文简明扼要地向读者介绍在开发计算机软件过程中实用的方法和技术,如分析阶段的SA(Structured Analysis)方法,设计阶段的SD(Structured Design)方法,并通过具体例子介绍这些方法的应用。

## 一、概 述

SA方法与SD方法是软件工程中常用的主要方法。其中SA方法用于软件开发中的系统分析阶段;SD方法用于设计阶段。这两种方法是互相衔接的,以其得到一个结构良好系统模型。

软件工程学研究的是:“如何应用一些科学理论和工程上的技术来指导软件的开发,从而达到用较少的投资获得高质量软件的最终目标”。

## 二、SA方法(结构化分析)

在软件系统的开发初期,用户和软件人员之间缺乏共同的语言,双方交流时存在着隔阂。用户熟悉本身的业务但不熟悉计算机技术,软件人员则熟悉计算机技术而不了解用户的业务。软件人员考虑的是程序结构、程序语言、数据结构、程序效率等问题,而用户并不能确切地理解这些概念。此外,软件系统规模大、逻辑复杂,用户的要求又经常有所变化,这些因素更增加了交流的困难。开发人员往往急于求成,于是在未完全明确软件系统究竟应该“做什么”的情况下,软件人员就开始进行设计工作,而用户则不清楚软件人员在设计一个怎样的系统。直至系统完成之后,用户才发现它们不符合要求,但这已经太迟了。

在这些教训的基础上,人们认识到:为了设计出满意的系统,首先必须有一段时间集中精力认真分析用户究竟要求系统“做什么”,这个阶段就是分析阶段。

### 2.1 SA方法的描述手段

目前用于分析阶段的技术很多,其中具有代表性的是“结构化分析”即SA方法。SA是结构化分析(Structured Analysis)的简称,它是由美国Yourdon公司提出的。它适合于分析大型的数据处理系统,特别是企事业管理方面的系统。SA方法适用于分析系统的功能:数据和加工两方面。在表达问题时,尽可能采用图形的方式,并采用“分解”和“抽象”这两个基本手段控制系统的复杂性。

对于一个复杂的系统,如何理解和表达它的功能呢?SA方法用了“由顶往下,对系统作逐层分解”的方法。

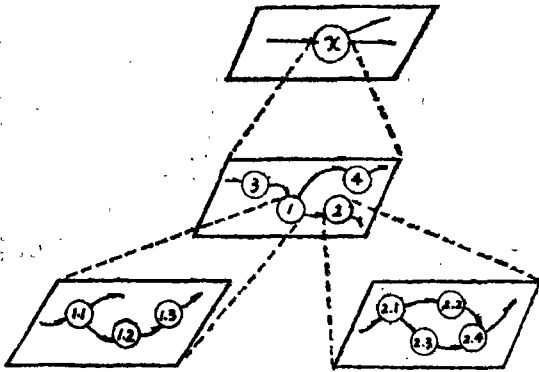


图1

图1中系统x很复杂,为了理解它,我们将它分解成1, 2, 3, 4四个子系统;如果子系统1, 2仍然很复杂,我们再将它们分解成1.1、1.2, ……; 2.1, 2.2, ……等子系统。如此继续下去,直到子系统足够简单,容易清楚地理解和表达为止。

图1中,顶层抽象地描述了整个系统,而底层具体地画出了系统的每一个细部,中间层则是从抽象到具体的逐步过渡。在对系统作了合理的分解以后,就可分别理解系统的每一个细部,然后为每一个细部写下说明,将所有这些说明组织起来,就获得了整个系统的说明书。系统说明书是分析阶段的最终产物,也可以说是用户与开发人员之间技术上的共同约定。开发人员将以它为基础进行系统的设计,用户与软件管理人员在软件系统投入运行时可以它作为验收系统的标准。当然在整个开发期间开发人员可根据具体情况不断地修改完善系统说明书。

用SA方法编写的系统说明书由三部分组成:

- 一套分层的数据流程图。
- 一本词典。
- 其他补充材料。

数据流程图说明系统由哪些部分组成,以及各部分之间的联系,它是说明书中最主要的部分。而词典则为数据流程图中出现的每一个成分提供详细的说明,再同补充材料结合起来就完整而精确地描述了系统的逻辑模型。

### 2.2 SA方法的基本步骤

目前大多数计算机系统都是用来代替一个当前已经存在的人工数据处理系统,对于这类系统的分析过程可分四步进行:

- 1) 理解当前的现实环境,获得当前系统的具体模型;
- 2) 从当前系统的具体模型抽象出当前系统的逻辑模型;
- 4) 分析未来系统与当前系统逻辑上的差别,设立目标系统的逻辑模型;
- 4) 为未来系统的逻辑模型作补充。

### 2.3 数据流程图

SA方法采用“分解”的手段来理解一个复杂的系统。数据流程图就是描述“分解”的工具。

数据流程图从数据加工的角度描述企事业的业务活动,它描述了一个组织分成哪几个部分,以及这些部分之间的联系,下面看一个培训中心管理系统的例子。

培训中心是一个功能很复杂的系统。它是这样处理日常业务的,学员发来的电报、文件、电话,经收集、分类后,分几种情况处理;①如果是报名的,则将报名数据送给负责报名的职员,他们要查阅课程文件,检查是否额满,然后在学生文件、课程文件上登记,并开出报名单交财务人员,财务人员再开发票经复审后通知学员;②如果是付款的,则由财务人员在帐目文件上登记,再经复审后给学员一个通知(收据);③如果是查询的,则交查询部门查阅课程文件后给出答复;④如果是要注销原来选修的课程,则由注销人员在课程、学生、帐目文件上作相应修改,经复审后通知学员;⑤对一些要求不清楚的函电,培训中心拒绝处理。图2是描述这个管理系统的流程图。

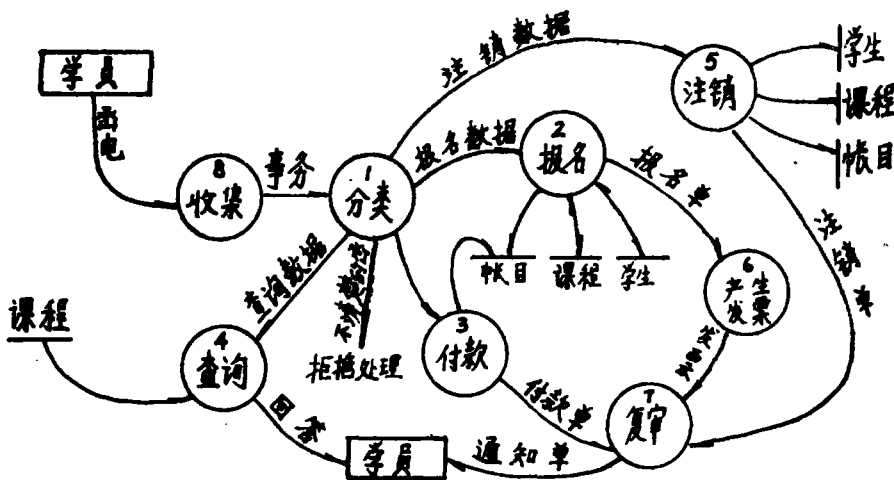


图2

图2 告诉我们:系统可以分解成收集、分类、报名,……等八个部分,这些部分之间有图2所示的联系。要理解整个系统,只要分别理解这八个部分就可以了。由于每个部分比起整个系统来小多了,所以问题将大大简化。

图2 中有四种基本成分:

- 数据流(用箭头表示)。
- 加工(用圆圈表示)。

- 文件（用直线段表示）。
- 数据流的源点或终点（用方框表示）各个成分都有一个互不相同的名字，以便标识。

### 三、SD方法（结构化设计）

SD是结构化设计（Structured Design）的简称，它是由美国IBM公司的W. Stevens、G. Myers和Z. Constantine等人提出的，目前在软件系统的结构设计中SD方法使用最广。这个方法适用于任何软件系统的总体设计，它可以同分析阶段的SA方法前后衔接起来使用。

在系统的总体设计中，SD方法考虑的是如何建立一个结构良好的程序系统，它研究了模块（在SD方法中所指的模块是能够用一个名字调用的一段程序）分解的影响，并在此基础上提出了评价模块结构质量的两个具体标准——块间联系和块内联系，还给出了从表达用户要求的数据流程图导出模块结构的规则。

#### 3.1 模块结构的质量标准

用SD方法设计的程序系统，模块具单一的功能，模块之间是相对独立的，衡量模块结构相对独立性的标准是块间联系和块内联系。块间联系是对模块独立性的直接衡量，块间联系越小就意味着模块的独立性越高。块内联系是指一个模块内部各成分（语句或语句段）之间的联系，块内联系大了，则模块的相对独立性势必会提高，如图3所示。

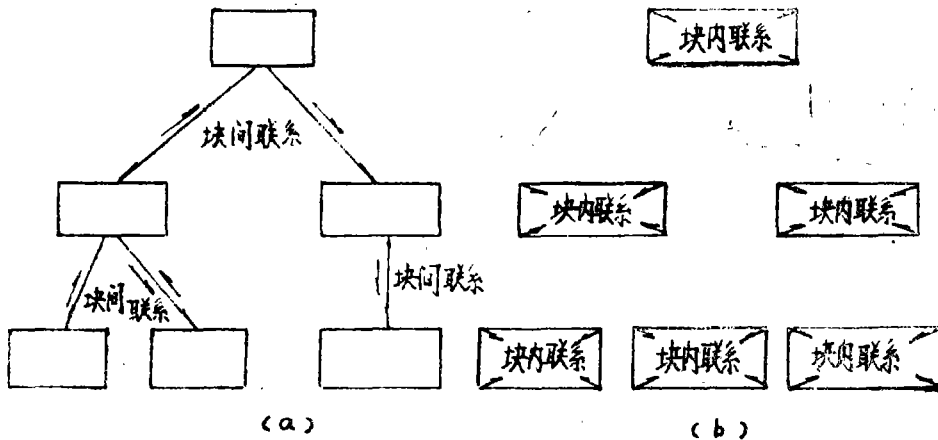


图3

SD方法的目标是使块间联系最小，块内联系最大，以便使每个模块可以独立地被理解、编写、测试、排错和修改，这就使复杂的开发工作得以简化。此外，模块的相对独立性也能有效地防止错误在模块之间扩散蔓延，因而提高了系统的可靠性。所以我们可以说SD方法的长处就是来自于模块之间的相对独立性，它提高了系统的质量（易理解性，易维护性、可靠性等）也减少了研制所需的人工。

结构图3（a）中的主要成分有：

- 模块——它用方框表示，方框中可写模块的名字，一个模块的名字应适当地反映这个模块的功能，这就在某种程度上反映了块内联系。
- 调用——从一个模块指向另一模块的箭头表示前一模块中含有对后一模块的调用。

• 数据——调用箭头边上的小箭头表示调用时从一个模块传送给另一模块的数据，小箭头也指出了传送的方向。

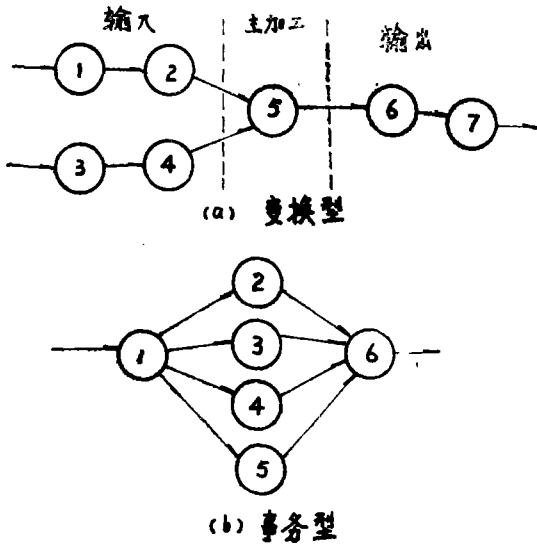


图4

表达的用户要求的程序结构。

下面仅通过一个例子讨论“变换分析”技术，以说明SD方法的具体应用。

例如一个企业管理系统，它很复杂，在分析阶段我们应用SA方法得到了一个表达用户要求的分层的数据流程图，其中生产统计子系统生产日报表部分如图5所示。

使用“变换分析”技术可以从图5导出标准形式的程序结构(图6)，其过程可分成以下三步：

- 找出主加工、逻辑输入和逻辑输出。
- 设计模块结构的顶层和第一层。
- 设计中，下层模块。

我们将所研究的这部分也称为一个系统。变换型结构的主加工是系统的中心工作，主加工的输入数据流称为系统的“逻辑输入”，主加工的输出数据流称为系统的“逻辑输出”，相对地，系统输入端的数据流称为“物理输入”，系统输出端的数据流称为“物理输出”。从输入设备获得的物理输入一般要经过编辑、格式转换、合理性检查等一系列辅助性加工后变成纯粹的“逻辑输入”传送给主加工，同样，主加工产生的纯粹的“逻辑输出”要经过格式转换、组成物理块，缓冲处理等辅助性加工后成为物理输出最后从系统送出。

下面分别讨论变换过程的三步。

1. 找出系统的主加工、逻辑输入和逻辑输出。

通常，几股数据流的汇合处往往是系统的主加工。如果一时不能确定主加工在哪里，则

3.2 从数据流程图导出初始结构图  
在分析阶段用SA方法获得了用数据流程图等描述的系统说明书。

数据流程图一般有两种典型的结构，变换型结构和事务型结构。如画4所示；变换型结构是一种线性的结构，它可以明显地分成输入、主加工和输出三部分。事务型结构中，某一个加工将它的输入分离成一串平行的数据流，然后选择执行后面的某一个加工。

这两种典型的结构可分别通过“变换分析”和“事务分析”技术导出标准形式的程序结构图，这些技术都是先设计模块结构顶端的主模块，然后“由顶向下逐步细化”，最后得到一个满足数据流程图所

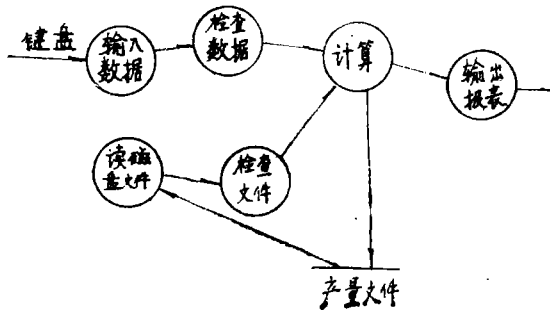


图5

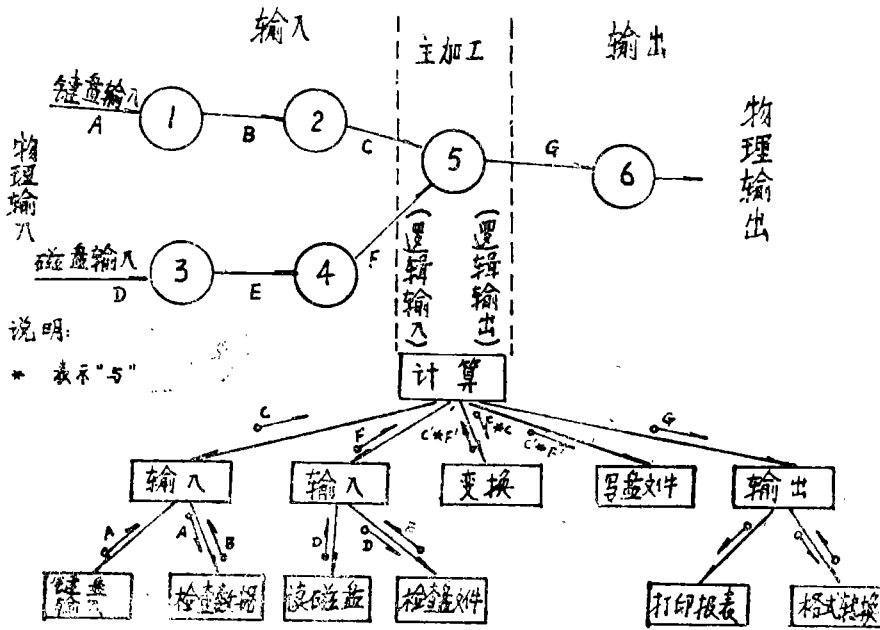


图6

可以用下面的方法决定哪些数据流是逻辑输入和逻辑输出；从物理输入端开始，一步步向系统中间移动，直至达到这样一个数据流：它已不能再被看作为系统的输入，则其前一个数据流就是系统的逻辑输入；同样，从物理输出端开始，一步步向系统中间移动，也可以找出逻辑输出。

由于各人看法不同，找出的主加工可能也不同，但一般不会相差很远。

### 2. 设计模块结构的顶层和第一层

由顶向下设计的关键是找出“顶”在哪里，决定了系统的主加工就是决定了程序结构的“顶”的位置，我们先设计一个主模块，并将它画在与主加工相应的位置上如图6所示，主模块的功能是完成整个程序要做的主要工作。

程序结构的“顶”设计好之后，下面的结构就可按输入、变换、输出等分支来处理，我们这样设计结构的第一层，为每一个逻辑输入设计一个输入模块，它的功能是向主模块提供数据。为每一个逻辑输出设计一个输出模块，它的功能是将主模块提供的数据输出。为主加工设计一个变换模块，对于日报表这个具体例子，数据经主加工后需将当天的生产数据记录在磁盘上，所以多一个“写磁盘文件”块。此时应注意第一层模块同主模块之间的传送的数据要同数据流程图相对应（见图6）。

这样就得到了结构图的上层，这里主模块控制并协调输入、变换、输出等模块的工作。一般说来，主要根据一些逻辑（条件或循环）来控制对这些模块的调用。

### 3. 设计中、下层模块

这一步是由顶向下，逐步细化地为每一个模块设计它的下属。

输入模块的功能是向它调用模块提供数据，所以它本身必定要有一个数据来源，因此输入模块可由两部分组成，一部分是接受输入数据——输入模块，另一部分将这些数据变换成其调用模块所需要的数据——变换模块，（图6）。同理，输出模块的功能是将其调用模块提供的数据输出，所以它也应该由两部分组成，一部分是将调用模块提供的数据变换成输出的形

式——变换模块，一部分是输出模块（图6）。上述设计过程可以由顶向下递归地进行，直至达到系统的输入端或输出端。

为模块设计下层没有一定的规则可遵循，此时需研究数据流程图中相应加工的组成情况，如图7中，主加工又由子加工X、Y、Z组成，所以第一层模块“K”可以有三个下层模块分别与子加工X、Y、Z相对应。

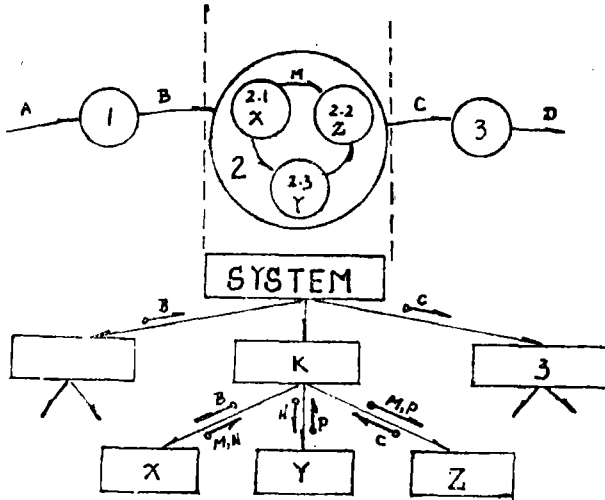


图7

上述调用模块与被调模块间传送的数据应同数据流程图相对应。每设计出一个新的模块应注意给它起一个适当的名字，以反映出这个模块的功能，

运用上述变换分析技术，我们可以较容易地获得与数据流程图相对应的初始结构图，即与问题结构相对应的程序结构，我们还可注意到，这种初始结构符合变换程序的标准形式，所以质量是比较好的。

### 参 考 文 献

- [1] 武汉大学软件工程研究所编：软件工程方法，1984；
- [2] 潘锦平编：软件开发技术，上海科学技术文献出版社，1985；