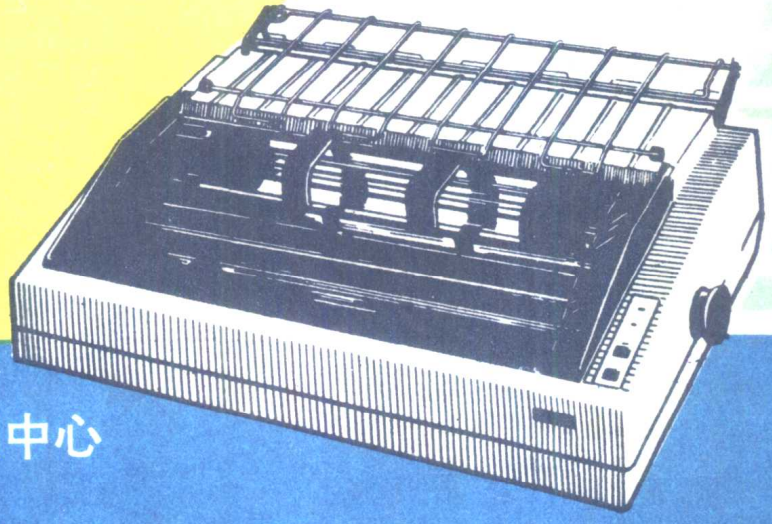
面向 dBASE & FOXBASE 数据库自动化系统（DAS）原理与实现

瓮正科　王新英　著



北京科海培训中心

面向 dBASE & FoxBASE 数据库
自动化系统(DAS)原理与实现

瓮正科　王新英　　著

中科院北京科海培训中心
1991.6

目　录

附录

第一章 数据库自动化引论

软件分为系统软件，应用软件和智能软件三大类。应用软件是耗费资金和人力最多的一类软件。支持应用软件开发的方法学是软件工程。多年来，人们为了寻找高效有力的开发方法，进行着不懈的努力和奋斗，但是，至今为止还没有找到一个最有效的方法。本章试图提出解决 MIS 软件开发与维护的新途径—数据库自动化，并在以后章节中给出具体的实现方法和程序。

1.1 数据库设计自动化

数据库应用系统的开发和一般软件的开发，有着共同点，也有着明显的区别。所以，数据库自动化与程序设计自动化之间必然也存在一定的区别，前者不仅要研究程序设计自动化问题，还要研究数据库本身自动化问题。另一方面，过去，人们只重视数据库本身的开发，而忽略数据库应用程序的开发。这都是片面的，我们认为，数据库自动化分为数据库设计自动化、应用程序设计自动化和数据库自适应三大部分。

定义：数据库设计自动化是指：从需求分析开始，到最后得到应用系统的整个开发过程，采用自动化设计方法，由计算机参与中间大部分开发活动，减少软件人员的大量、复杂的重复劳动，从而有效地提高数据库系统开发速度，缩短开发周期；同时，系统的维护也必须使计算机参与，使得维护实现自动化。目前在这方面最主要的工作是 CASE 技术，利用这种技术开发的软件产品，能为应用系统分析人员、设计人员和开发人员提供一个完整的开发环境。数据库 CASE 技术主要包括四个内容，即 CASE * Method、CASE * Dictionary、CASE * Designer 和 CASE * Generator，下面来具体论述。

(1) CASE * Method

CASE * Method 实质上是数据库设计方法学的一个方法库，它是一个软件系统，用它来解决数据库的设计问题。目前数据库设计方法有 E－R 方法、最小覆盖方法、结构化方法、实体分析方法等等，采用那种方法仍然是需要探讨的。在这个方法库中，是将各种方法都放进去，还是突出一种方法，都是需要进行研究的。美国 ORACLE 公司研制的 CASE * Method 提供多种模型构造技术。我们提出的数据库结构化设计方法很适合于目前大多数人的设计习惯，但要把它作成软件包尚需要进一步开发。

总之，CASE * Method 是数据库设计自动化的核心，它贯穿整个数据库开发生命周期，指导应用项目的成功开发过程。

(2) CASE * Dictionary

字典是一个可移植的、计算机化的多用户数据词典系统。它用于存储、组织和定义系统生命周期的各个阶段所收集的全部信息。一般说来字典应该具备两个基本特点：

①一致性　它包括使用一致性和修改一致性，使用一致性指不论在程序中、数据库逻辑结构中，还是在文档中，对同一个字段信息的内容应该是一致的，要做到这一点，必须使信息出自同一处。修改一致性指对字典的内容修改必须保证只有一个地方可以修改，一旦被修改，其它任何使用该修改信息的地方，都应该自动被修改，或者其它需要修改的地方被提示。一致性是字典系统的最重要的指标。

②完整性　对数据库文件结构的描述必须完整，因为数据库的生成将根据字典的内容进行生成。另外若干应用程序的生成也依赖于字典中的信息。所以，如果信息不完整，应用系统的开发就不可能完整。

关于字典的功能，长期以来，人们把字典都看成仅仅是分析、设计的辅助工具，是一种文档，事实上，不应该仅仅作为文档，而应该作为程序的一个部分，例如，作为菜单的内容等等。

(3) CASE * Designer

CASE * Designer 是为数据库设计人员和系统分析人员提供的一个高级的、高度图形化的用户接口软件。这种软件的主要目的是供开发人员能够方便、灵活地建立各种图形化模型如 E－R 模型等，同时该软件也是开发人员和用户之间的一个桥梁，加强两者之间的沟通和协作，能使开发者对应用系统的一些错误认识及时得到纠正，使构造出来的模型能够准确地反映其系统需求，进而为以后的各个阶段打好基础。

这种软件的需要性完全决定 CASE * Method，因为如果模型不是图形形式，要这个设计师意义不大。

(4) CASE * Generator

CASE * Generator 是数据库自动化中最重要的内容，它的任务是根据字典信息和设计报告书自动生成出一个应用系统，当然包括数据库和应用程序两部分，前者包括数据库文件名，文件结构，一些库的物理结构，甚至一部分数据要能够生成出来，在作者研制的数据库自动化系统(简称 DAS，下同)中，这些功能得到实现。另一部分就是应用程序生成，这在程序设计自动化部分详细论述。

综上所述，数据库本身的设计自动化最主要是方法库的设计，字典和生成器一种辅助工具，设计师决定与方法库的关系。另外必须引起足够重视的是自适应调节器，它的主要任务是收集用户不断提出的新要求，设计成自适应报告书，供生成器再生成，这样就构成了数据库结构数据部分的自动化。

1.2 数据库程序设计自动化

所谓程序设计自动化是指通过一个给定的描述，经过一个程序变换成所希望的代码，从而省去人工大量的烦琐、不易、低效和重复劳动。当前，基于程序自动设计系统主要采用四种方法即过程法、演绎法、变换法和检查法，其中变换法是最受欢迎的方法，论述如下。

定义：所谓变换法是指自动程序设计系统的输入是一种甚高级语言编写的程序，通过施用一系列变换把这种输入转变成低级实现。实际上就是由一个程序转至成另一个功能等价的程序。

从变换法的定义可以了解到，根据该方法研制自动化系统由三部分组成，(1)一个模式(或样板)，(2)一组逻辑适用性条件，(3)一个动作过程。当找到一个模式实例时，就检查逻辑适用性条件，确定是否需要施用变换，如果需要，则求值动作以计算新的一段代码，用以代替由模式匹配的代码。这种系统的典型结构如图 1.2.1 所示。

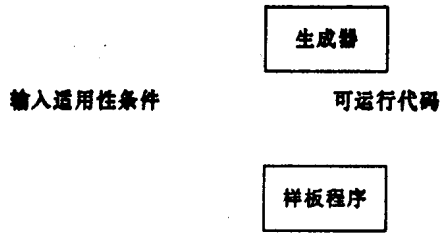


图 1.2.1 程序生成器的典型结构

这个典型结构给人们对程序生成一个直观的认识，但是，程序变换系统有时可能不需要样板程序，有时不一定每次都生成一个程序，只作一些代替运行。一般说来程序变换系统有三种基本变换形式，既算法变换型、代替变换型和运行变换型，分述如下：

1. 算法变换型 对有一些程序的生成，有时不需要样板程序，由变换算法直接输出可运行的程序。

例如，有如下关系：

关系模型 ：学生情况表

文件名称 ：STUDENT. DBF

字段信息字典

中文说明 chinese	字段名称 field—name	类型 field—type	长度 field—len	小数点数 field—dec
学　　号	SNO	C	6	0
姓　　名	SNAME	C	8	0
性　　别	SSEX	C	2	0
出身日期	SDATE	D	8	0

求学生情况表的输入格式程序。通过 DAS 中的“屏幕生成”功能，经过对话，便生成如下程序：

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAM NAME, STUDEN02. FMT
* 生成日期:06/18/90
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
@ 3,20 SAY ' 学生情况表 '
@ 5, 2 SAY ' 学 号 ' GET SNO PICTURE ' XXXXXX'
@ 7, 2 SAY ' 姓 名 ' GET SNAME PICTURE ' XXXXXXXX'
@ 9, 2 SAY ' 性别 ' GET SSEX PICTURE ' XX'
@ 11, 2 SAY ' 出身日期 ' GET SDATE
RETURN
```

生成以上程序的程序在以后章节中介绍。算法变换型其变换器本身是一个算法包，算法的描述均以生成目标的方式被设计，所以，运行生成程序之后，输入适当的条件就可以生成出所需要的程序。

2. 运行变换型。这种方法的特点是没有生成的程序，这似乎不可思议，固然生成系统不

生成程序，事实上这是一个很有效的方法。这种方法的主要想法是生成器作为应用系统的一部分，在运行过程中，根据输入的适用性条件，选中样板程序，生成器将适用性条件代替样板程序中待匹配内容，然后直接运行，运行之后，样板程序归位，供下一次变换。这种方法的最大优点是使得所生成的应用系统程序量达到最少。例如有如下一段程序：

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAM NAME:iasXX.prg
* NOTE:检索
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
......
*   打开文件
SELECT 1
USE &XX1
STORE .T. TO TT
DO WHILE TT
   LIST &XX2 FOR &XX3
   STORE "N" TO LP
   @1,2 SAY" 还查询吗？[Y/N]" GET lp
   READ
   IF LP<>"Y"
     STORE .F. TO LP
  ENDIF
ENDDO
RETURN
```

这是一个通用检索程序，其中三变量 XX1,XX2,XX3 是未知的，当对那个数据库文件操作时，就添入相应的内容。运行变换型的程序的主要任务就是当运行该程序时，如何把这三个变量换掉。例如在 SELECT 1 之前加一句 DO IASWJ,IASWJ 程序是专门用来取文件名，并将文件名赋给 XX1 变量。在 LIST 命令之前，调用一个子程序，完成取字段名任务，并赋给 XX2；再调用一个子程序，完成条件表达式的编辑，并将表达式赋给 XX3。这种每次在运行过程中，完成表达式的替换，就是运行变换型生成方法。事实上若干 MIS 通用辅助开发工具都是根据这个思想开发的。

3. 代替变换型。这种方法是最典型的变换方法，国内若干 dBASE 生成系统都采用这种方法，它根据输入条件，选中某个样板程序，生成出所需要的程序，然后构成一个应用系统。具体做法是对每个库文件，将诸如索引检索程序通过表达式变换后，变成专用程序，这样和人们开发的应用系统一样。下面介绍具体步骤。

(1)样板程序设计　样板程序是程序生成系统的基础，样板程序的质量决定了所生成的应用系统的质量，所以对样板程序必须千锤百炼。下面通过一个具体实例来说明样板程序的设计。

例：表 1 是一个学生情况表，要求设计一个编辑该文件的样板程序，其选择菜单要求具有两个选择项。

表 1　　学生情况表　　(STUD.DBF)

学　号	姓　名	年　龄	性　别
SNO	SNAME	OLD	SEX
C (6,0)	C (8,0)	C(2,0)	C(2,0)

900101	欧阳奋进	20	男
900102	王 望	21	男
900103	肖丽丽	20	女

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAME NAME:TEST1.PRG
* 功能:编辑
* 生成参数 提示 1 和 2,条件 1 和 2 的字段名
* 参数行号: TS1=012,TS2=013,TJ1=020,TJ2=025
* 生成日期:
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
USE STUD.DBF
STORE .T. TO LOP
DO WHILE LOP
   CLEAR
   @ 10,10 SAY TS1
   @ 12,10 SAY TS2
   @ 14,10 SAY "0.....返 回"
   STORE "0" TOKEY
   @ 20,10 SAY "请输入数字 [0,1,2,..]" GET KEY PICTURE "!"
   READ
   DO CASE
   CASE KEY="1"
        STORE TJ1 TO TJJ1
        @ 22,10 SAY "编辑条件" GET TJJ1
        READ
        EDIT FOR &TJJ1
  CASE  KEY="2"
        STORE TJ2 TO TJJ2
        @ 22,10 SAY "编辑条件" GET TJJ2
        EDIT FOR &TJJ2
  CASE  KEY="0"
        STORE .F. TO LOP
  ENDCASE
ENDDO
USE
RETURN
```

程序 TEST1.PRG 就是根据要求所设计的样板程序,设计这种程序应该注意:①文件名将要在生成时被替换,所以它所在行和列不能随便改变,最好放在第二行;②程序生成日期在生成时根据当日系统日期添写;③程序第 5 行是要进行变换的内容描述行,TS1,TS2 是菜单提示信息变量,TJ1,TJ2 是要进行编辑的条件变量。其中 USE STUD.DBF,如果把文件名改为变量,通过对话确定所打开的文件,这样就对任何文件都适合了。当然在实际中可以使该样板具有更强的功能。将若干个样板程序合在一起就形成所谓的样板程序库。

(2)生成器的设计　根据用户输入的适用性条件,从样板库中提取相应的样板程序,启动生成器实现一系列变换,就能生成可执行代码。生成器实际上是一个字符串替代程序,它根据样板程序头部(第 5 行)所标识的变量内容,用户在运行变换器时,将这些变量赋值(输

入),然后实现字符串替换,就生成出代码。程序 TEST2. PRG 是针对例子的生成程序,它主要完成 ①将程序名改为用户的;②生成当日程序生成日期;③根据输入菜单信息替换样掉板程序中的 TS1 和 TS2 变量;④根据输入的条件信息(条件表达式初值)给条件表达式变量赋初值。它基本工作方式是利用一个数据库(字段名为 CC,类型为 C,长度=78),将样板程序拷入该库内,然后在相应记录上根据输入信息处理替代,完后再将记录拷贝成. PRG 文件,就生成了可执行程序(SDEIT. PRG)。

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAME NAME,TEST2. PRG
* 功能:生成学生文件的编辑程序
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
USE SC. DBF
DELETE ALL
PACK
APPEND FROM TEST1. PRG SDF GO 2
REPLACE CC WITH SUBSTR(CC,1,16);+"SEDIT. PRG"
GO 6
REPLACE CC WITH SUBSTR(CC,1,13);+DTOC(DATE())
* 取参数和行号
GO 5
STORE SUBSTR(CC,13,3) TO TS1
STORE SUBSTR(CC,21,3) TO TS2
STORE SUBSTR(CC,29,3) TO TJ1
STORE SUBSTR(CC,37,3) TO TJ2
STORE VAL(SUBSTR(CC,17,3)) TO XH1
STORE VAL(SUBSTR(CC,25,3)) TO XH2
STORE VAL(SUBSTR(CC,33,3)) TO XH3
STORE VAL(SUBSTR(CC,41,3)) TO XH4
* 区参数名和行号
GO XH1
STORE SPACE(20) TO T1,T2,T3,T4
@ 20,10 SAY "输入提示 1" GET T1
READ
STORE "'"+TRIM(T1)+"'" TO T1
DO TEST3 WITH T1,TS1
GO XH3
@ 20,42 SAY "输入字段名" GET T3
READ
STORE "'"+T3+"'" TO T3
DO TEST3 WITH T3,TJ1
GO XH2
@ 22,10 SAY "输入提示 2" GET T2
READ
STORE "'"+TRIM(T2)+"'" TO T2
DO TEST3 WITH T2,TS2
GO XH4
@ 22,42 SAY "输入字段名" GET T4
READ
```

```
STORE "'"+T4+"'" TO T4
DO TEST3 WITH T4,TJ2
GO 4
DELETE NEXT 2
PACK
COPY TO SEDIT.PRG SDF
USE
RETURN
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAME NAME, TEST3.PRG
* 替换内容
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PARAMETERS TT,TTT
STORE 1 TO N
DO WHILE N<78
   IF SUBSTR(CC,N,3)=TTT
   REPLACE CC WITH SUBSTR(CC,1,N-1)+TT+SUBSTR(CC,N+3,78)
   ENDIF
   STORE N+1 TO N
ENDDO
 RETURN
```

(3)程序生成　在 dBASE Ⅲ PLUS 运行 TEST2.PRG 程序，屏幕要求输入：

输入提示 1 1.....根据年龄编辑　输入字段名 OLD

输入提示 2 2.....根据性别编辑　输入字段名 SEX

完成后，就生成出 SEDIT.PRG 程序。

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAME NAME,SEDIT.PRG
* 功能:编辑
* 生成日期: 01/27/90
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
USE STUD.DBF
STORE .T. TO LOP
DO WHILE LOP
   CLEAR
   @ 10,10 SAY '1.....根据年龄编辑'
   @ 12,10 SAY '2.....根据性别编辑'
   @ 14,10 SAY "0.....返 回"
   STORE "0" TO KEY
   @ 20,10 SAY "请输入数字[0,1,2,..]" GET KEY PICTURE "!"
   READ
   DO CASE
   CASE KEY="1"
   STORE 'OLD     ' TO TJJ1
   @ 22,10 SAY "编辑条件" GET TJJ1
   READ
   EDIT FOR &TJJ1
CASE KEY="2"
    STORE 'SEX     ' TO TJJ2
```

```
        @ 22,10 SAY "编辑条件" GET TJJ2
        EDIT FOR &TJJ2
    CASE KEY="0"
        STORE .F. TO LOP
    ENDCASE
ENDDO
USE
RETURN
```

该方法的最大缺点是应用程序的冗余量大，对一个大中型应用项目尤为突出。

这三种变换方法在一个生成器中可能同时存在，可能采用一两种方法。关于变换正确性问题，一般说来，变换是正确性的，意思是匹配的代码与替代在逻辑上表示等价计算。变换法的主要优势是，它们为某些种类的程序设计知识提供了一个十分清楚的表示。在数据库自动化中，由于数据库应用程序本身所固有的特性，用变换法无疑是最好的方法。

1.3 数据库系统的不稳定性

长期以来，人们对数据库系统本身的特性没有进行具体的刻划，在开发方法学上费了很大功夫，最终的结果，仍然是不尽人意，仍然是在费时费力。我们想是否可以回过头来仔细研究一下数据库系统本身的特性。下面我们通过一个具体的实例来看一看一个实际系统是如何变化的。

例：教学管理数据库系统(部分)有如下几个文件：

学生文件：STUDENT. DBF

SNO	SNAME	SOLD	SSEX	SCLASS
870101	王小艳	18	女	微机871
870201	李明	18	男	微机872
870102	司马奋进	19	男	微机871
870202	李明	18	女	微机872
870103	成功	18	男	微机871

成绩文件：SCORE. DBF

SNO	CNO	SCORE
870101	C601	98.00
870102	C604	78.00
870201	C601	88.00
870101	C602	99.00
870202	C603	89.22

授课文件：TEACHING. DBF

TNAME	CNO	CTIME	SCLASS
崔正科	C601	200	微机87
李成刚	C602	100	微机871
严明	C603	120	微机871
李刚	C604	220	微机871

课程文件：COURSE. DBF

CNO	CNAME
C601	数据库原理与应用
C602	操作系统
C603	数据结构
C604	微机原理与应用

当用户提出这样一个数据库系统的文件，把它开发成一个可运行的系统，我想不会太难，在很短时间内，就能够实现。但是，在运行过程中，发生如下情况，结果会如何呢？

1. **信息定义改变。**最初约定一个教员只教一门课程，后来由于教员不够，允许一个教员教两个到三个门课程，这样教员和课程的关系由一对一变为一对多关系。授课文件中的TNAME就不能作为关键字了。

2. **新的字段信息需要加到数据库中去。**例如，需要在某学生文件中增加通信地址的字段，对姓名字段需要增加宽度以适合处理姓名有五个汉字的情况等等。

3. **数据表示的改变。**例如，有的字段开始用代码表示，可在使用过程中，发现反而不方便，要求改过来，或者反过来，希由非编码方式换为编码方式。

4. 访问方法的改变。例如，原来采用顺序查询方法，使用一段时间后，发现速度太慢，需要改变查询方法，比如用索引或索引顺序方法等等。

5. 次要关系性质的改变。例如，在数据库最初建立时并不重要的教学工作量关系（教师姓名，学时数，班级，学期）只要求打印出报表，可使用一段时间后，对该关系增加几个字段，可以用来作为教员考评的重要关系，它由次要关系变为重要关系，必须对该关系增加一系列的操作功能。

6. 关系系统的闭合性质。关系数据库的最大特点是查询过程的得到的新关系又可以作为系统的基本关系加以确定，从而使用更方便。例如，学生成绩视图关系（学生姓名，班级，课名，成绩）本来作为一个查询结果，但将它确定下来作为学生成绩输入和修改，确显得更为方便直观。

7. 关系的划分。例如，在使用过程中，发现如果将某个关系化分成两个或多个关系会使得更方便。例如授课文件，划分为一个教师文件和一个授课文件。

8. 关系的合并。在使用过程中，多关系的查询可能是影响速度的主要原因，如果将几个关系合并起来，速度可能得到大大地加快。

上述情况可以概括为：①数据库结构是时变的，②数据库应用模式也是时变的，由于这两者的变化，使得你所开发的程序也要跟着变化。这三种变化是随时变化的，那么任何一个一成不变的数据库系统都不能适应客观对象。一个软件的开发完成，只说明这个软件的诞生，不能算就能生存。要保证能够健慷的运行下去，必须不断地随着客观要求的变化而变化。由于客观环境是不断变化的，所以说数据库系统是不稳定性的，需要随时维护。

1.4 自适应数据库的定义

由于数据库系统的不稳定性，就迫使人们来寻找一种克服不稳定性方法，我们从自适应控制系统的角度来论述这个问题。

自适应控制系统被定义为：能够按某一给定准则或优良度来连续地测量系统品质，并且靠闭环作用自动地改变系统可调参数，以满足该优良度。该系统的框图如下：

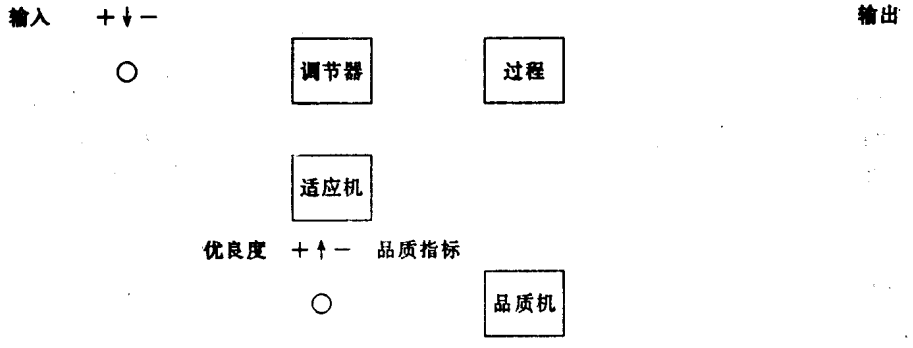


图 1.4.1 典型的自适应控制系统

从定义和图 1.4.1 可以看出，自适应控制系统是通过对实际输出的不断测量，得到实际品质，然后和希望的标准进行比较，由自适应计算机算出的信号去进行参数调整。这个定义告诉我们，自适应系统的三个主要职能是识别——决策——变更，所谓识别指将系统的动态特性化为一个可为调节器所采用的形式；决策指根据当前的品质和希望的品质的有关信息

而动作，决定要产生什么样的校正作用，确定所需之调节器的特征或必须的驱动信号；修改指根据所作的决策起作用，进行所需求的修改使系统趋向最优。自适应控制系统的这个概念可以通过适当的修改来定义自适应数据库系统，其框图可修改如下：

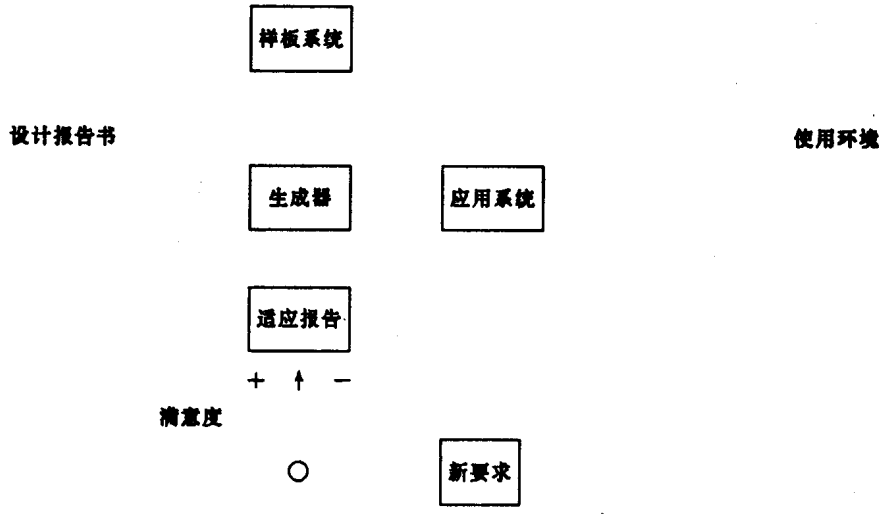


图 1.4.2 典型的自适应数据库系统

在这个框图的上半部分是典型的生成系统，下面是自适应回路。一般来说，一个数据库应用的设计，开始根据一个预想的应用模式和结构进行设计，当使用环境发生变化时，提出新的要求，和满意度进行比较，决定要不要进行适应调节，需要，产生一份适应报告书，供生成器进行调节生成，使得应用系统满足新的使用环境，达到满意度。由此可以得到自适应数据库系统的定义。

定义：能够根据使用环境的变化，通过决策，迅速产生适应报告书，靠生成器来自动改变数据库应用系统的结构和操作，以满足满意度。自适应数据库系统的三大职能也是识别——决策——变更，不过，和控制系统不同的是后者识别是最困难，其次是决策，最容易是调节部分。然而数据库则相反，识别最容易，因为使用环境变化必然提出新要求，决策不仅要考虑使用要求，还要看修改成本。适应报告不仅要依据新要求，还要根据生成器的规定。最后的修改动作是最重要的一环，如果象以前通过手工的方式修改，几乎是办不到的，那么生成器就决定了系统能否进行自适应操作。下面在讨论生成器的结构之前，先来分析一下数据库应用系统的实质。

1.5 数据库应用程序准则

从程序设计语言的角度看，数据库是一个(大的、结构化)变量。数据库模式(或数据库内涵)定义这个变量的类型。类似地，关系也是一个变量，相应的关系模式(关系内涵)给出变量的类型，如此等等。数据库值(或数据库状态或数据库外延)是由数据库模式所定义的变量的值。类似地，关系的值(状态，外延)是由某个关系模式所定义的值等等。其中的关系当然指实关系。

我们这样理解是有许多好处，程序设计的若干经验就可以使用。所谓数据库应用系统就是数据库＋操作。操作可能是一组命令，可能是一个应用程序系统。象在 dBASE 数据库和

FOXBASE 数据库系统下开发的应用系统，操作必然是一个应用程序系统，因为它门是面向程序员的 DBMS。那么数据库应用系统的实质是数据库＋应用程序系统。数据库的基本特性是结构时变性和应用模式时变性。应用程序系统具有什么基本特性呢？这类程序本质是什么呢？通过仔细地研究我们发现这类程序本质上就是围绕着文件名和字段信息操作的一系列指令的集合。长期以来，人们习惯地把文件名和字段信息写在程序中，造成程序和结构的紧偶合，系统开发之后，经过运行，一旦发现结构有问题，就得去改程序。如果时间一长，改程序成为不可能，只有抛弃该系统，这就是目前的普遍现象，使得无数的人无味地重复劳动，造成不可估量的经济损失和智力浪费。那么是否可以将结构和操作指令分开来呢？形成所谓的松偶合，事实上完全可以，也只有这样做，这是数据库应用系统能够自适应的必要条件。一旦我们把程序和文件名及字段信息分开，那么应用程序系统的时变内容就是输入格式、输出格式和未命名关系。为此我们给出数据库应用程序的准则如下：

准则 1：程序应该与文件名分离。这就是说在程序中不写文件名，因为如果写上，一旦文件名变化，就得将所有写该文件名的程序进行修改。要想自动修改，能够做到，但较繁杂。

准则 2：程序应该与字段信息分离。即在程序中不固定描述字段信息，否则自动修改将非常困难。

准则 3：屏幕格式文件（输入和输出）应该自动生成。因为该文件包含字段的大量信息，如果字段信息变化，那么相对位置、名称、格式要作一系列修改。手工修改程序很费事，必须自动生成。

准则 4：报表应该是自动生成。如果一个应用系统有 20 个报表输出，一旦逻辑数据库结构变化，这些报表程序就得作一系列修改，这种维护是非常痛苦的，因此必须采用自动形式。

准则 5：未命名关系（即表达式）应该具有共同的编辑方式，并能存储。程序中如检索，插入，编辑，删除，统计，计算等等，到处都存在表达式，而这些表达式都包含了字段的大量信息，一旦字段变化就得在程序中找这些表达式进行修改，手工修改很繁杂。如果这些表达式是存在文件中，且具有统一形式方式编辑修改，程序适应就很容易。

这 5 条准则是保证应用程序系统能够自适应的必要条件。否则不可能实现自适应。

1.6 自适应数据库方法

从上两节的分析可以总结出数据库应用系统的三大特性，(1)由于使用环境变化，数据库结构是时变的，(2)数据库应用模式是时变的，(3)从而导致程序也是时变的。要使得数据库应用系统能够自适应，就必须使得这三个时变内容能够自动变更，这就是所谓的自适应数据库系统。知道了自适应数据库系统的特性之后，如何实现这三大内容的自动变更呢？下面给出具体的方法思想，具体算法在以后章节中介绍。

方法 1：数据字典方法：在软件工程中，数据字典通常作为分析，设计和维护的重要手段，但为什么不能作为程序的一部分呢？事实上数据字典可作为数据库自动变更的重要手段，这才有它的真正应用之处。为此，可建立三个字典，即文件名字典，字段信息字典和未命名关系字典。

文件名字典包括数据库的文件名称，中文说明和与该文件有关的文件名。程序中需要用到文件名时可以到该字典中取，当文件名发生变化时，只要修改字典，与程序无关。

字段信息字典包括数据库的字段信息，即字段名，中文描述，字段长度，类型等。

未命名关系字典包括每个数据库文件操作过程中的希望存储的表达式。

方法 2：格式文件自动生成方法：对于屏幕格式文件可以自动计算生成，可采用算法变

换型方法。

方法 3：通用报表方法：通常报表有两类，一是一个报表一个程序（有时有一些公用部分），二是通用报表。前者，如果逻辑数据库变化，与此有关的报表程序都需要修改，这要实现程序自动适应很困难。后者才是解决问题的唯一方法。通用报表国内有许多，都有一个共同的特点，功能很强，操作繁杂，因为它们把查询、计算、统计、打印全都溶在一起。事实上可以把查询、计算、统计的处理留给应用系统去解决，而通用报表只解决如何根据一个输出库，自动地设计出一个用户满意的表格并输出结果。这里介绍一下它的基本思想。首先将各式各样的报表总结出来（约 18 至 30 类型），将这些报表基本型装在一个库内，根据用户要输出的某个库，到数据字典中找到相应的文件中文详细说明，字段的中文说明和字段的其它信息，由用户选择一个报表基本型，自动地设计出表格，用户若不满意可稍作修改，尔后，用表格和输出库数据装备成一个打印库，即完成一个报表的生成输出任务。这个过程完全借助于字典进行的，因此应用环境变化，字典是要变化的，报表只要重新生成即可。这个输出程序本身是自动适应的。

这三种方法，方法 1 是 CASE 字典的主要内容，方法 2 至 3 与其说是自适应方法，倒不如说是生成器的重要方法。关于自适应回路的程序支持，主要任务是将使用环境变化的要求转变成上述方法规定的输入格式，供生成器调节生成。这一部分转换程序有待于进一步开发。

1.7 数据库自动化模型

从前几节的论述，我们可以抽象为一个具体的开发维护模型，首先讨论一下传统的数据库开发模型，如图 1.7.1 所示。注意该模型与一般教课书中的模型有所区别，增加了“编码”这个环节，是因为实际数据库应用系统开发过程中，总是存在这一步骤。一个应用系统不编码是不可能的。这个开发模型提出了不少有益的概念、步骤、方法和技术，为提高软件生产率，保证软件产品可靠性，指导人们开发各种类型的数据库系统起了积极的作用。但是，也必须指出：

第一，在这样的开发模型中，开发中的每一步只给出最后结果，而开发过程是非形式化的，以手工劳动为主，其中的决策及其理由并未记录下来，而这些决策及其理由对维护来说是至关重要的，因此，就带来维护上的极端困难。

第二，传统的开发模型本质上是一个开环系统，人们根据预想的目标进行设计，所设计的应用系统在短时间内能够适合用户要求，随着时间的推移和使用环境的变化，原设计的系统很可能不再适应使用要求而被费弃，或者经过复杂的维护才能继续使用。

第三，传统的开发过程速度太慢，由于若干简单劳动必须重复进行，导致开发费用过高，周期太长。

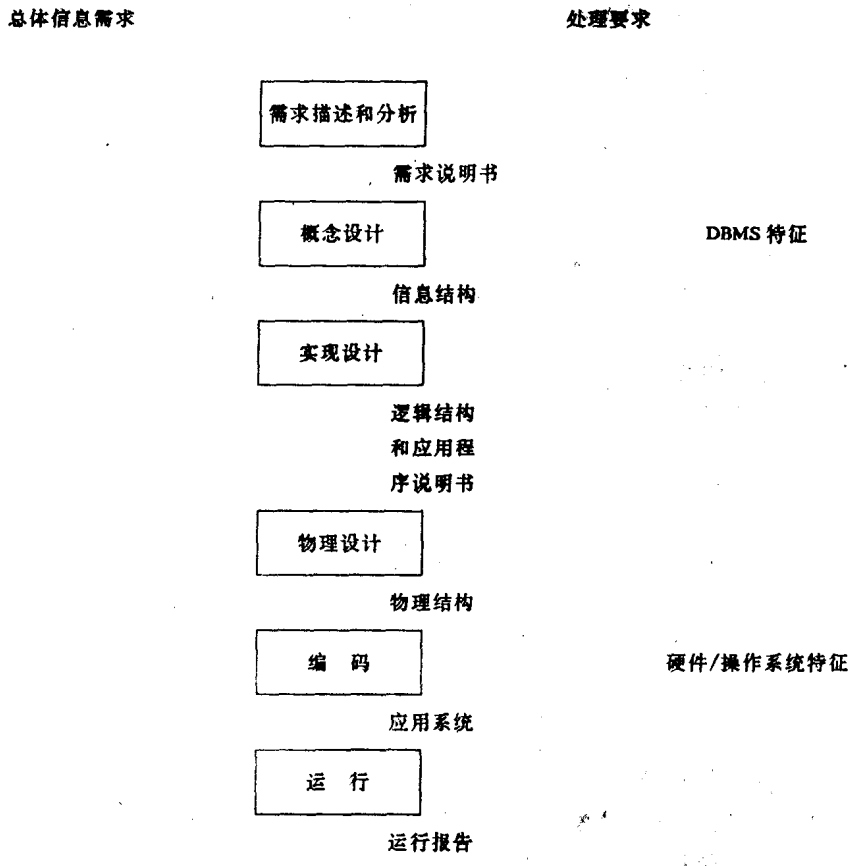


图 1.7.1 传统的数据库开发模型

因此，有必要研究其它开发模型，以提高开发数据库应用系统的生产率，在量的提高的同时在质上也有所突破。

图 1.7.2 是数据库自动化模型，它的基本出发点是使人们尽量减少重复劳动，让机器更多的参与数据库的整个开发过程。这个模型真正地实现系统闭环，由自适应环节来自动跟踪使用环境的变化。