

集成于 DBMS 的应用生成器

陈 启 泉

[计算机科学(电脑)系]

摘要 本文介绍一种把应用生成器集成于数据库管理系统的设计技术,讨论各应用生成器的功能和用户界面,对应用生成器与批命令语言的比较和结合方法也做了探讨。

关键词 应用生成器,数据库管理系统,批命令语言

0 引言

计算机(特别是微机)应用系统的开发者正面临着如下问题:(1)开发新应用系统的需求不断增加,各种应用系统的程序编写有不少共性,不同开发者进行着不同程度的重复劳动。(2)新应用系统的功能和数据流越来越复杂,开发周期长难以满足用户单位的急切需求。(3)缺乏熟练的系统分析员和程序员,非计算机专业人员无法独立开发自己的应用系统,而计算机专业人员也需花较长时间才能熟悉用户单位的业务处理和数据流过程。为了解决面临的问题,各种应用自动生成工具的研究应运而生,但研究成果一般都只是局部的,仅作为软件包提供系统开发人员选用。如何把最常用的自动生成工具有效地集成于数据库管理系统中,作为开发应用系统的支持软件和友好的用户界面是一个很有实用价值和值得研究的问题。

应用生成器(Application Generators,简称 AG)的提出和研究,最早起源于 IBM 公司的报表生成系统 RPG^[1],它在用户界面上提供用户定义,并自动生成各类报表而不必为页面布局和数据表示详细书写程序。之后又相继出现了图形生成器、菜单生成器、屏幕生成器、实用表格生成器^[2]等,以及非常方便友好的数据输入输出视窗和用户界面。应用生成器的特点是在很高的层次上用第四代软件语言 4GL 描述现实世界,只要告诉计算机想要什么,而不必为计算机的实现细节编写程序。回顾计算机的发展历史,硬件经历了由电子管→晶体管→集成电路→大规模集成电路,软件也经历了由机器指令→汇编语言→高级语言→第四代语言(4GL)等四个阶段,各种应用生成器所使用的描述语言均属于第四代语言^[3]。本文介绍一种把应用生成器集成于数据库管理系统中的设计技术和方法,讨论各应用生成器的功能和用户界面,其描述语言主要参

本文1991-05-31收到。

• 省自然科学基金资助项目

照我们在 xdb 关系数据库系统基础上加以扩充和完善并具有汉字处理能力的 HDDDB 系统.

1 报表生成器(Report Generator, 简称 RG)

报表生成器用来在较高层次上支持用户利用数据库文件或数据库查询结果所获得的数据自动生成用户所需要的各类报表, 用户界面可使用交互式对话和报表批命令描述语言两种方式. 使用 RG 生成报表, 用户只要把注意力集中于描述报表的数据源、精心设计报表的框架和布局, 而不必为实现的细节编写程序. 报表生成器描述语言就其功能而言主要有以下四类命令语句.

1.1 指定数据源 报表中所用的数据可以来自数据库文件, 这时只要在关键词 `report` 之后给出文件名即可. 例如

```
report employee
```

表示当前要生成的报表所使用的数据均来自数据库文件 `employee` 中的数据项. 报表的数据源也可以来自数据库查询结果, 例如

```
select * from employee
```

```
where 工资 > 150
```

```
report
```

表示当前要生成的报表来自满足某一约束条件的数据库查询结果临时文件中的数据项. 至于系统如何从物理数据库中提取数据的实现细节对用户是完全透明的.

1.2 表项设计与排列 描述报表中表项数据在打印时的位置、宽度与排列次序等. 主要命令语句有

```
Format <域名> i/j/w
```

其中参数 `i, j, w` 可选用于指定该表域输出时在报表中的准确位置和宽度. `<域名>` 是指定数据库文件或查询结果文件的数据项名.

```
ADD n, w <表项名> = <表达式>
```

其中 `<表项名>` 是根据报表的数据源进行指定的计算操作, 并把计算结果作为报表中的一个新表项, 参数 `n, w` 用来指定新表项在报表中的位置和宽度.

```
Exclude <域名> 和 include <域名> <位置>
```

调整源数据文件中数据项出现在报表中的排列次序, `Exclude` 语句用于把源文件中的某些数据项暂时排除在报表外, `include` 语句用于把已被暂时排除在外的某些数据项重新安装在报表中的指定位置.

```
Rename <旧名> <新名>
```

源文件中数据项的名字在报表中出现时可以重新命名, 以适应不同应用环境的用户各自使用的习惯称谓.

1.3 报表的框架和布局 为使输出的报表更加美观和具有易读性, 符合用户单位习惯使用的报表格式, 有必要精心设计报表的表头标题和小标题, 对重要数据分组排序和分组统计打印使报表更清晰美观. 常用的命令语句有

`Heading` 和 `Title`: 用于描述报表的表头标题及分组统计时的小标题说明及标题位置.

Sort 和 Group:对指定的表项列排序及分组隔空行输出。

统计函数(Total, Average, Count, Max, Min):对指定表项进行整表或分组统计计算,并把计算结果按指定格式放入报表中指定位置,可与 Title 语句配合使用为统计值所在行加入小标题。

Date 和 Prompt:指定报表每页页头要否打印日期和运行 RG 时显示必要的用户提示信息。

Break:指定分组排序时每组数据之间要留出的空白行数。

1.4 打印输出格式 描述硬拷贝时报表的打印输出规格。主要命令语句有

Printmode:指定输出打印所要的字体和大小,包括选择所用的汉字字体和大小。

Page onoff:指定打印纸每页要否加上编号。

Size:规定每页打印纸要印出的行数。

Margin 和 line:规定每页打印纸打印时上下左右应留出的空白行、列数。

Draw:指定要否为报表画边框以及边框线的空格位置,标题要否画底线等。

上述命令语句都是在很高的层次上告诉计算机要什么,而不必关心如何实现。实际上要实现每一命令操作系统均需要执行一段不短的程序段。由此可见应用生成器描述语言的效率比起一般高级语言要高20倍以上。下面给出一个用报表命令语言书写的批命令例子,即

prompt 1 "输入部门号:"

prompt 2 "输入当天日期:"

```
select * from 职工,部门 where 职工.部门号=部门.部门号 and 职工.部门号=&1
report
```

sort 2 部门号,部门名

add 5 10 日工资=时薪*工时

format 姓名 0/5/12

rename 时薪 计时工资

exclude 部门号

include 部门号 1

Total 工资>>工资额>>部门号 y, btitle

Total for 部门 %1:累计工资额

合计:

Title

福华针织有限公司

部门?%1 日工资报表

page on

size 66

Group 2 "部门名:%2 部门经理:%3"

exclude 出生日期,住址,邮政编码

Margin 5 5 9

line top

heading all

```
prtmode c  
save b:payroll  
print.
```

2 图形生成器(Graph Generator,简称 GG)

图形生成器的研究是为了满足商业和企业部门数据处理时经常要在报表中配插图以便使重要信息更加一目了然. 用户只要用图形描述语言指定图形的类型、图形标题和坐标说明、刻度和数据源,就可由 GG 自动生成用户所需要的图形信息. GG 和 RG 有效地结合就可在很大程度上满足企事业单位数据处理和报表呈递的要求,从而大大缩短应用系统的开发周期.

图形生成器是根据用户所指定的数据库文件或数据库查询结果文件所获得的数值型数据集自动生成所需的图形. 图形生成器绘制的图形有两类,即基本图形和专用图形. 基本图形通常有六种类型:圆饼图(pie),直方图(bar),折线图(curve),线点图(line),坐标点图(x-y)和散点图(scatter)^[3]. 专用图形则是为各种特定业务要求而设计的图形库作为软件包供系统开发者调用. 用户可根据自身的业务需要指定某些类图形在报表的指定位置内绘制输出. GG 的主要功能及其相应的描述语言如下进行论述.

(1)指定图形数据源:用于产生图形的数据源可来自数据库文件或查询结果文件中的数值型数据,但产生图形标号及坐标说明的信息也可来自两类文件中的字符型数据. 其命令语句有 Graph〈文件名〉或〈查询块〉Graph 两种.

(2)指定图形类型和空间位置:为了绘制图形信息,用户必须描述所要绘制的图形类型、图形标号、所使用的数据项以及该图形所占空间的对角坐标位置. 语句为

Graph〈类型〉〈标号源〉〈数据项名列〉〈对角坐标〉

(3)安排图形标题和坐标轴说明:用于安排所绘制图形的总标题、小标题、图线和刻度说明以及坐标轴含意说明等. 主要语句有

Title〈字符串〉

Location〈行列号〉〈参数〉

xLegend〈字符串〉

yLegend〈字符串〉

其中〈参数〉可选择要否划底线、横排、竖排等特定要求.

(4)刻度说明和图线特征:指定数值型数据源数值的变化范围和刻度单位、图线特征等信息. 本系统提供的图线特征有实心、空心 and 横条心三种. 描述语句有

scale〈最小值〉〈最大值〉

unit〈坐标位置〉〈字符串〉

feature〈特征参数〉

3 菜单生成器(Menu Generator,简称 MG)

菜单生成器用于支持用户把数据库管理系统功能模块和各种应用生成工具以及任何 DOS

可执行文件构筑在菜单屏幕中依序执行,菜单屏幕可以嵌套使用(即一个菜单屏幕允许调用另一菜单屏幕),形成菜单树结构.一个应用系统的开发,只要适当运用 DBMS 的模块和应用生成器,按照人们所认识的外部世界定义数据流通中作为数据流和信息流的各类实用表格和报表,对个别特别复杂的数据处理和计算适当编写一点程序,就可在很短时间里开发和生成整个应用软件系统.用这种技术开发的软件系统有着明显的优点,即开发周期短,非熟练计算机人员可以自己开发本部门的应用系统,软件系统自身文档化(self-documenting),源程序易于理解和维护等.

菜单生成器包括菜单描述和菜单命令两部分,后部分为 DBMS 和 DOS 可执行的模块、语句和批命令文件名,前部分则为对相应命令和语句的说明和描述,使系统更具有易读性.下面给出一个由 MG 所定义的菜单屏幕例子.

豪华电器公司通用计账系统		
菜单1——输入数据		
1.	输入现金支出单据	
2.	输入现金收入单据	
3.	菜单2——账目查询	
4.	运行账目处理程序	
5.	输入文件格式转换	
6.	出口	

其中入口项3表示由菜单1调用菜单2的嵌套结构.菜单描述和菜单命令的相应关系在表1中定义.

表1 计账系统菜单命令

入口项	菜 单 描 述	菜单命令
1	输入现金支出单据	Form disburse
2	输入现金收入单据	Form receipts
3	菜单2——账目查询	run menu2
4	运行账目处理程序	Account
5	输入文件格式转换	import
6	出口	

其中 Form disburse、Form receipts、run menu2 为 DBMS 中应用生成工具模块名字^[3],Account 为批命令文件名字,import 为由 DBMS 所提供的把 ASCII 文件转换为数据库文件的格式自动转换模块名字.菜单生成器常用的命令语句有

- 定义菜单标题:Title
- 输入菜单描述:Description
- 指定菜单命令:Command
- 输入用户提示:Prompt
- 保存菜单命令:Save

·运行菜单屏幕:Run

4 屏幕生成器(Screen Generator,简称 SG)

屏幕生成器是由软件系统生成并提供给用户的友好视窗,使用户在使用应用软件系统时感到很直观方便,它的应用范围贯穿于 DBMS 各功能模块和应用生成工具中. SG 常用的技术有四种主要类型:数据库文件框架生成、输入输出屏幕接口、文本编辑多屏幕技术和菜单屏幕显示技术.

4.1 数据库文件框架生成 先由系统显示空白框架让用户指定要激活的文件名,系统立即显示出原文件的框架,包括每一数据项名字、类型、宽度和是否允许空值等,如为新文件则显示必要的提示信息. 用户可以进一步要求浏览(Browse)该数据库文件的记录值,可按页显示,在页中上下进退行,必要时可立即更新数据值等,用户不必为这些工作编写程序.

4.2 输入输出屏幕接口 用户只要指定要录入数据的文件名,系统就自动显示记录中每一数据项的描述信息,用户按每一数据项逐一录入数据,系统具有数据完整性检查功能以尽量确保输入的数据准确无误. 这些过程都是既自动又直观,系统还提供方便灵活的让用户操纵光标和编辑修改录入数据的功能,使用户感到非常方便,对输出数据也采用类似过程. 据文献[2]统计指出,一个应用系统的开发,按照常规编程需花50%的程序行处理输入和输出. 由此可见使用 AG 技术可大大缩短应用系统的开发时间.

4.3 数据编辑多屏幕技术 通常有分区编辑技术(一个屏幕分成几个区域同时显示多个文本文件供编辑和对照);多文件换屏编辑技术(同时编辑多个文件,每个文件信息内容占用一个屏幕,但可藉助功能键控制触发换屏编辑别的文件,不同文件部分信息内容可以互相拷贝或迁移);屏幕滚动技术,包括全局滚动、局部滚动和左右移动等.

4.4 菜单屏幕显示技术 有静态显示和动态显示两种方式. 静态显示通常采用分级菜单,上弹和下拉菜单,集内容、提示、出错信息和菜单选择于一屏幕,触发显示求助以及交互对话屏幕管理等技术. 动态显示则采用活动图像和文字信息流显示并配以适当的响声以引起用户的注意和兴趣等屏幕显示技术.

5 应用生成器与批命令语言的比较与接口

集成于数据库管理系统的应用生成器是以现实世界的实体为处理对象,其描述语言没有使用明确的循环和递归操作,它的命令语句是在比 DBMS 批命令语言更高的界面上告诉计算机想要什么,诸如报表的数据源和布局,图形的类型和安排,实用表格的样式和数据抽取方式,菜单和屏幕的设计和构筑等,而不必为实现这些目的设计应执行的命令序列,它只是在高层上运用系统所涉及的物理数据库内在的数据类型而已. 因此可以说 AG 的数据结构、数据类型以及循环和递归操作都是“蕴含”的,对用户而言是透明的.

DBMS 的批命令语言仍需为实现某一目的书写程序,它仍然保留有类型和结构定义,分枝和循环操作,但对处理复杂的计算比 AG 有更大的灵活性,因为它可直接运用诸如数组、记录、指针等结构. 如何把应用生成器与批命令有效地结合是 AG 研究的重要方向. 通常有以下三种

处理方法.

(1)让 AG 与批命令语言彼此独立,在层次上使批命令语言处于 AG 之上,也即把 AG 作为批命令语言与 DBMS 的接口.其组织方式如图1所示.这一组织方式的优点是能较大幅度地保留批命令语言有效进行复杂计算和数据处理和灵活性的能力,实现也较容易,但缺点是用户仍感不方便,因为程序员必须记住放置在共享工作区中有关数据的格式和体积,且要为这些数据在通讯前进行格式转换.这一工作尚需熟练的程序员才能完成.

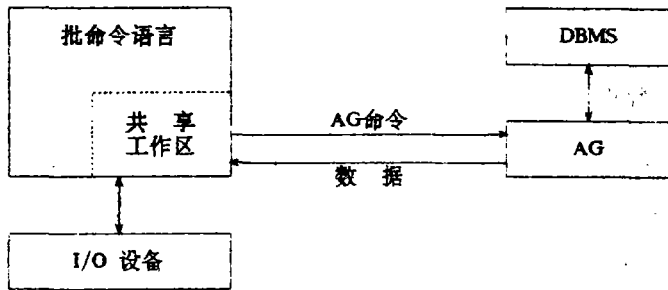


图1 AG作为批命令与DBMS的接口

(2)把 AG 命令语句嵌入于批命令语言:扩充批命令语言的语句使之直接包含 AG 命令语句,操作时不必进行格式转换.也即在批命令语言中增加如下语句:Report〈报表名〉,Graph〈图形名〉,Menu〈菜单名〉,Form〈表格名〉...等.其组织方式如图2.这一组织方法较可取,它可集批命令语言和 AG 的优点于一身,大大增强了批命令语言的能力.

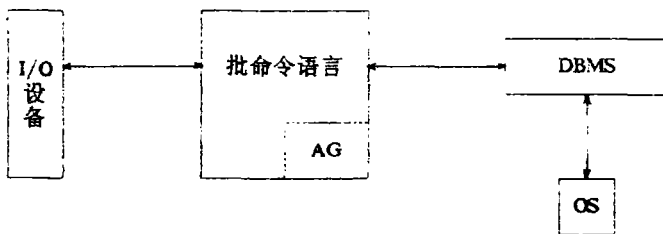


图2 AG语句嵌入批命令语言

(3)把批命令语言嵌入于 AG 中:允许应用生成工具运行到某一个入口项转去执行一段事先已定义好的批命令文件而后返回原处.我们在 HDDB 系统中目前采用这一方式.在实用表格生成器(Form Generator)^[3]中增加 ON Entry、ON changed 和 ON Exit 功能.当进入一个新表项、当前表项已被修改过或从当前表项出口时允许转去执行一段批命令程序而后再返回原处,以增强复杂计算和数据处理能力.其组织方式如图3.

AG 与批命令语言较好的结合应该融(2)(3)于一体,即既可把 AG 嵌入于批命令语言,也可把批命令语言嵌入于各应用生成工具的任一入口项.集成于数据库管理系统中的应用生成器可与批命令语言一样作为 DBMS 的独立模块出现,但两者又互相渗透可互相调用.这一结合

可大大增强 DBMS 的能力,使其提供给用户的外部界面上升到一个新的数量级. 这时 DBMS 的组织方式如图4所示:

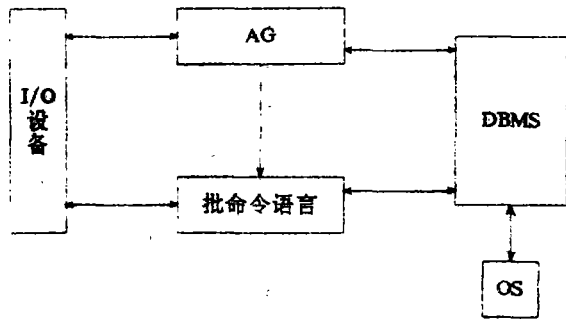


图3 批命令语言嵌入AG

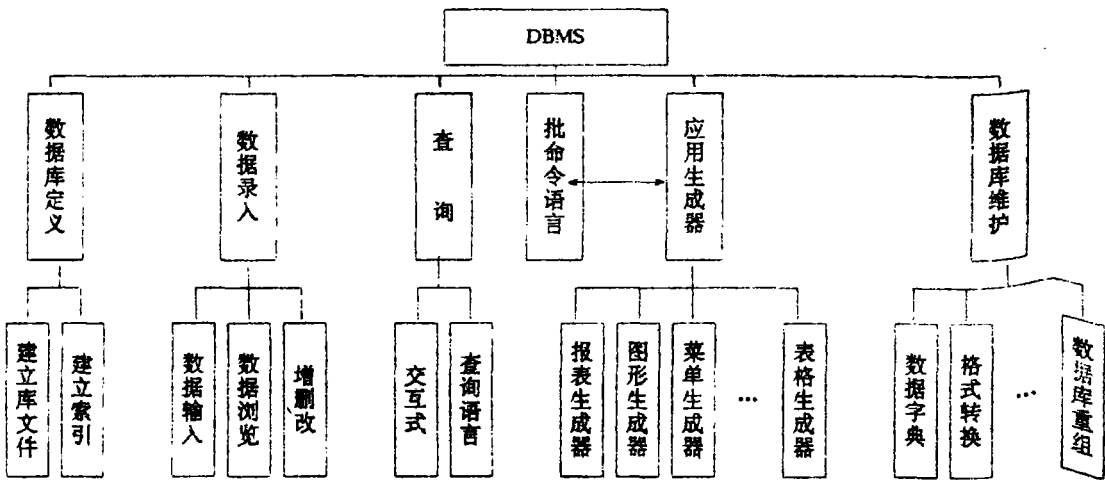


图4 DBMS组织结构

6 结束语

应用生成器的研究正与第四代语言(4GL)的研究并行发展,前者是后者的研究成果和具体体现. 由于目前尚没有公认的规范化4GL 语法文本,因而各应用生成工具所使用的描述语言也没有统一的语法规范. 本文所介绍的 AG 描述语言与其它类似软件产品所用的描述语言无论在处理方法上或语句格式都不尽相同,各有特色. 随着对4GL 的研究不断趋于成熟,将来对应用生成器的描述语言也将会逐步趋于统一和规范化.

参 考 文 献

- [1] Horowitz, Uemper, A., *IEEE Software*, 1(1985), 40—54.
- [2] Rowe, L. A. and Shoens, K. A. *IEEE Trans*, 9, 1(1983), 31—39.
- [3] 陈启泉, 应用生成工具 *HDPORM* 系统的设计技术, 华侨大学学报(自然科学版), 12, 2(1991), 222—228.
- [4] Tucker, A. B. *Programming Languages*, McGraw—Hill, Computer Science Series(1977).

The Integration of Application Generator into Data Base Management System(DBMS)

Chen Qiquan

(Department of Computer Science)

Abstract In this paper, the design effort for intergating application generator into data base management system(DBMS)is introduced; the function and user interface of each application are discussed; the comparison and association between application generator and batch command language are also examined.

Key word application generator, data base management(DBMS), batch command language