Developer/2000 开发工具集成的方法

马宏余 范牧昌 (华东船舶工业学院机械工程系 212003)

摘要:本文通过实例详细论述了一种如何在 Oracle Developer/2000 环境下,利用 Oracle * forms 对另外两种开发工具进行集成的方案,并对其他可行的方案进行了探讨。

关键词:分布式的数据库 触发子 参数调用

1. 引言

Developer/2000 包含三个主要开发工具:Oracle * forms, Oracle * reports, Oracle * graphics。Oracle * forms 是一个开发和运行 Windows 下的基于表格的开发工具,使用该工具开发的 Forms 可以灵活地定义、操纵(增、删、改、查询)和控制数据库对象。Oracle * reports 是一个运行和开发 Windows 下报表的工具,其主要功能是从 Oracle 数据库中检索数据,并对数据进行必要的处理,增添适当的文字、标题和图像,最终生成用户所需的报表。用 Oracle * graphics 可以将数据库中的数据用一些常见的图形表示出来,使数据间的比例关系,数据的变化趋势以及数据的分布特征一目了然。

2. 在 Forms 模块中对 Reports 模块和 Graphics 模块进行集成的方法

Forms 模块向其他模块传递参数和数据必须借助于参数表,因为 Forms 模块往往是开发集成应用的核心模块,因此准确地将数据和参数从数据模块传给其他模块显得尤为重要。

一般而言,Forms 模块和 Reports 模块及 Graphics 模块之间的调用是双向的,既能由 Forms 模块调用 Reports 模块或 Graphics 模块,也能在 Graphics 和 Reports 模块中调用 Forms 模块,信息的传递也是双向的。也就是说用户可在同一画面既看到数据又能看到表示数据及相互关系的图形,并且可以得到最终产品——报表。

Forms 模块可以通过下列两种途径来调用 Reports 模块:

(1)用封装例程 RUN-PRODUCT()来调用 Reports 模块,此方法中被调用的模块可以以 SYNCHRONOUS 或 ASYNCHRONOUS 的方式运行,数据可以通过参数表中的文本参数和数据参数进行传递,这是一种比较受欢迎的方法。例如:RUN-PRODUCT(REPORTS,'REP-ORD', SYNCHRONOUS, BATCH, FILESYSTEM, PL-ID,'ITEM')其中的参数 REPORTS 表示被调用的模块为 Reports 模块;'REP-ORD'表示被调用的模块名称;SYNCHRONOUS 表示同步通信模式,BATCH 表示以后台方式运行;FILESYSTEM 表示被调用的模块存储在文件系统中;PL-ID 表示传递的参数列表名;'ITEM'表示 Forms 模块中的项目名称。

(2)用封装例程 HOST 调用 Reports 模块,这种方法较前一种方法的功能要弱一些,一个新的完全分离的运行会话以 SYNCHRONOUS 方式建立,且数据只能通过文本参数作为命令行的一部分进行传递。

上述两种方法也适用于 Forms 模块调用 Graphics 模块,除此之外,在 Developer/2000 工具集中有一个特殊的库 OG.PLL,可以通过 OG.PLL 来实现 Forms 对 Graphics 的调用。三种方法中 OG.PLL 的集成能力最强,可以实现 Forms 模块和 Graphics 模块的双向信息传递,而且还可以通过鼠标事件的传递来实现更强的功能,但此方法比较繁琐复杂。封装例程 RUN-PRODUCT()是最简单而有效的集成途径。

下面通过举例详细说明如何在 Forms 模块中用 RUN-PRODUCT()实现其与 Reports 模块和 Graphics 模块的有效集成。

3. 用 RUN-PRODUCT()实现集成的具体方法

本例中 Forms 模块通过对 Reports 模块的调用,将得到某公司员工信息的报表,这些信息包含员工的工号、姓名、工资及所在的部门等等,并可对此报表进行打印,若经过校确认无误后,再提交数据库。而 Forms 模块对 Graphics 模块的调用,将使用图表的形式来表示各个部门雇员的工资,使人一目了然。具体的实现方法如下:

(1)使用 Oracle * reports 定义一个独立的 Reports 模块,其名为 EMP-REP,其中 Query 定义为:SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP;将 default layout 指定为 Tabular。

最后使用菜单 FILE —→ ADMINISTRATION—→ GENERATE 成功地生成名为'EMP-REP'的可执行模块。

(2)使用 Oracle * graphics designer 创建一个图表用以表示各部门雇员工资的分布。

首先定义参数 DNO,类型为 NUMBER,初始值为 10,参数值将决定显示哪一部门雇员的工资。定义查询为 Query0,相关的 SELECT 语句为:

SELECT ENAME , SAL FROM EMP WHERE DEPTNO= :DNO;最后将此模块生成名为'SAL.OGD'的可执行模块。

(3)使用 Oracle * forms 定义一个 Form 模块,取名为'EMP-INFO'。

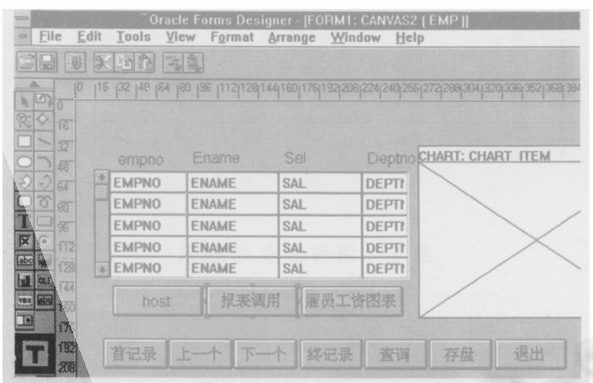

图 1　定义完成的 FORM 表格

在此 Form 模块中定义一个基于表 EMP 的块,取名为 emp-block,在该 BLOCK 中包含 EMP 表中的列 empno, ename, sal, deptno。并用 Forms Designer 的 Layout 打开表项画布,用图表工具在画布上拉出一个小区域,以用来显示由 Oracle * graphics 模块生成的图形,将此图表项命名为'chart-item',并将它的 display items 定义为 1。

(4)在 Form 中定义实现报表调用功能以及显示雇员工资分布的按钮,分别命名为 REPORT 和 DISPLAY。

至此已定义完 Form 图表,如图 1 所示:

(5)为实现报表调用及图形显示功能的按钮添加触发子,这将是实现集成功能的关键。

为按钮 Print 定义一个 WHEN-BUTTON-PRESSED 触发子,具体触发程序如下:

```
declare
    pl-id          paramlist;
    rg-id          recordgroup;
    gc-empno       groupcolumn;
    gc-ename       groupcolumn;
    gc-sal         groupcolumn;
    gc-hiredate    groupcolumn;
    gc-deptno      groupcolumn;

begin
/ * 创建记录组,如存在,则删除 * /
rg-id: = find-group('emp-group');
  if id-null(rg-id) then
    rg-id: = create-group('emp-group');
    gc-empno: = add-group-column(rg-id, 'empno', number-column);
    gc-ename: = add-group-column(rg-id, 'ename', char-column, 20);
    gc-sal: = add-group-column(rg-id, 'sal', number-column);
    gc-deptno: = add-group-column(rg-id, 'deptno', number-column);

  else delete-group-row(rg-id, all-rows);
    gc-empno: = find-column('emp-group.empno');
    gc-ename: = find-column('emp-group.ename');
    gc-sal: = find-column('emp-group.sal');
    gc-deptno: = find-column('emp-group.deptno');
      end if;
/ * 将表中数据填入块中 * /
go-block('emp-block');
  execute-query;
  first-record;
for row-count in 1..20 loop
    add-group-row(rg-id, end-of-group);
    set-group-number-cell (gc-empno, row-count, :emp-block.empno);
    set-group-char-cell(gc-ename, row-count, :emp-block.
```

```
ename);
    set-group-number-cell ( gc-sal, row-count, : emp-block.
sal);
    set-group-date-cell ( gc-hiredate, row-count, : emp-
block. hiredate);
    set-group-number-cell ( gc-deptno, row-count, : emp-
block. deptno);
    exit when :system. last-record = 'TRUE';
    next-record;
  end loop;/ * 创建参数表,并把参数填入参数表中 * /
  pl-id: = get-parameter-list('emp-list');
  if not id-null(pl-id) then
    destroy-parameter-list(pl-id);
  end if ;
  pl-id: = create-parameter-list('emp-list');
  add-parameter ( pl-id, ' Q-emp', data-parameter, ' emp-
group');
  / * 调用报表模块功能 * /
  run-product(reports, 'c: \ orawin \ bin \ emp-rep', syn-
chronous, batch, filesystem, pl-id);
    end;
```

同样为显示按钮 display 也定义一个 WHEN-BUTTON-PRESSED 触发子,Forms 模块将通过此触发子向 Graphics 模块传递文本参数,将 Forms 模块中的部门号 emp-block. deptno 传递给 Graphics 中定义的参数:DNO。Graphics 模块将该部门(部门号为:DNO)中的雇员的工资从数据库中查询出来,并用图表对之进行显示。具体的触发程序如下:

```
declare
    pl-id paramlist;
begin
/ * 建立参数表,若存在,则删除

* / pl-id: = get-parameter-list('tmpdata');
  if not id-null(pl-id) then
    destroy-parameter-list(pl-id);
  end if;
pl-id: = create-parameter-list('tmdata');
/ * 将 Form 中的参数部门号:emp-block. deptno 转化成文
本类型,添加到参数表中, 并与 Graphics 中的参数:DNO
建立对应关系。
* /
add-parameter(pl-id, 'DNO', text-parameter, to-char(:emp-
block. deptno));
/ * 以同步后台模式运行 Graphics 模块
* /
    run-product(graphics, 'sal. ogd', synchronous, batch,
filesystem, pl-id, 'chart-item');
end;
```

(6)运行所建立的 Forms 模块,生成相应的可执行程序。运行时按报表调用按钮就能运行经过编辑后所生成的报表。图 2 是报表的一次运行结果,并可对此报表进行打印。按显示按钮就可显示当前记录所在部门号的雇员工资分布图。图 3 是一运行结果,表示部门为 30 的雇员工资分布图。

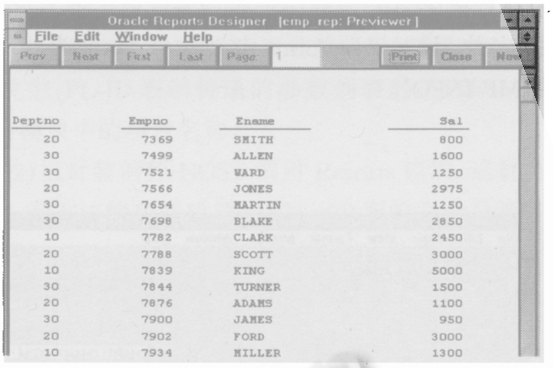

图 2 调用报表的结果

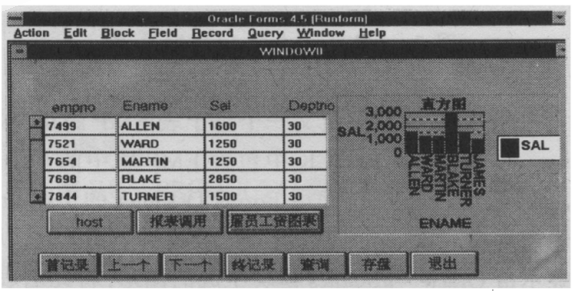

图 3 调用图表的结果

(下转第 72 页)

(上接第 64 页)

4.利用 RUN-PRODUCT()进行集成的关键技术

用 RUN-PRODUCT()对 Oracle * reports 以及 Oracle * graphics 进行集成的关键主要有以下两点:

(1)由 Forms 模块正确地向 Reports 及 Graphics 模块传递文本参数或者数据参数,参数作为联系不同模块的桥梁必须准确。否则即使是微小的偏差,也会造成意想不到的错误,从而导致模块调用的失败。

(2)合理搭配 RUN-PRODUCT()的调用参数。用 RUN-PRODUCT()在 Forms 模块中对 Reports 及 Graphics 模块进行集成时,调用参数的设计是关键技术,其中的通信模式 COMMMODE(SYNCHRONOUS, ASYNCHRONOUS)和运行模式 EXECMODE(BATCH, RUNTIME)的选择最为关键,传递文本参数时通信方式既可为同步(SYNCHRONOUS)也可为异步方式(ASYNCHRONOUS),但传递数据参数时只能为同步方式。同样如果将运行模式设置为前台(RUNTIME)则会建立一个新的完全分离的用户对话,不能实现其与 Forms 模块的有效集成,即所调用的模块对参数不予理会,而使调用失败。

5.结论

在集成 Forms 与其他模块的集成过程中,RUN-PRODUCT()是 Forms 模块与 Reports 模块集成时功能最强的一种。但对于 Forms 模块与 Graphics 模块之间的集成,功能最强的方法是利用库函数 OG.PLL,但调用方法比较复杂,难以掌握。可见灵活有效地运用 RUN-PRODUCT()函数对于数据库的开发很有必要。

参考文献

[1] 刘丽、罗含等译.《Oracle8 数据库构造工具实用指南》.北京:机械工业出版社.1998.6

[2] 刘金亭、朱莉、蔡蔚编.《CDE 协同开发环境 developer/2000》.北京:电子工业出版社.1996.8

[3] 刘晓荣编.《ORACLE DEVELOPER/2000 使用技术与方法》.北京:科学技术出版社.1996.3

(来稿时间:1999 年 1 月)