

# 开发 Oracle WebServer 应用程序的方法

Approaches for Development of Applied Programs on Oracle WebServer

秦存良

(胜利油田东辛地质研究所计算机室 山东东辛257094)

**Abstract** With several practical cases, this paper has outlined the system management and methods for developing software on Oracle WebServer, which included its basic structure, system configuration, approaches of applied development, advanced practical cases, mechanism of security and so on.

**Keywords** Oracle Web, Application, Management, Development

## 1 前言

Oracle WebServer 是一个功能较为强大的、使 WWW 与数据库紧紧联系在一起、集 Web 服务与开发环境为一体的 Internet 产品。Oracle WebServer 的全部管理工作由 Webserver 管理员在管理页面上配置完成,然后开发人员可以进行开发工作。程序源代码存储在 Oracle 数据库中,每次收到请求都将动态生成一个 Web 页面。Web Server 从客户端浏览器收悉 URL,由数据库和操作系统文件中返回相应的信息以响应请求,文件系统被用作生成静态 Web 主页或作为不访问数据库的 CGI 脚本,而利用数据库 PL/SQL 过程可在 Web 页面上动态地从库中提取相关数据,从而生成动态的 Web 页面。Oracle WebServer 的优势就在于它是 Webserver 与 Oracle 数据库的集成体,能使数据库广泛应用于 Web 领域中。

## 2 Oracle WebServer 的结构

Oracle WebServer 附在 Oracle 数据库之上,Webserver 管理员对数据库操作的权限一般与数据库管理员的权限是等同的,他们同属于 dba 组,也能正常启动和关闭 Oracle 数据库。Webserver 管理员独立地管理 Oracle WebServer,它在安装 Oracle WebServer 时,由操作系统管理员创建。其全部管理工作均可以简单地在浏览器 Web 页面上配置完成,当然,也可以在操作系统里通过修改某些文件,达到管理 WebServer 的目的,尽管这种方式是不可取的。Oracle WebServer 的管理工作主要有以下三个部分

1) The Web Listener: 是 Oracle WebServer 与局域网或 WWW 的一个接口,它遵循超文本传输协议,以便监听、响应来自客户端 Web 浏览器的请求。一个 Listener 可以建在一个或几个通讯端口上,而一个端口只能有一个 Listener。只有当某一 Listener 启动时,用户才能从相应的端口浏览到相关信息,WebServer 管理员修改配置后,都应先停止(stop)相应的 Listener,然后,再从新启动(start)Listener。

2) The PL/SQL Agent: 主要用来管理 DCD (Database Connection Description), DCD 包含了欲连接的数据库的帐户、口令以及 ORACLE\_SID, 端口号等等,是客户端与数据库建立连接的保障。另外,Oracle WebServer 应用可以用 PL/SQL 写过程完成,当然,也可以用 Java 语言嵌入 PL/SQL 过程,每一个 PL/SQL 过程都存储在的数据库中,PL/SQL 也是 Oracle WebServer 应用的开发语言。

3) The Web Request Broker (WRB): 是 Oracle WebServ-

er 的内核,是一个在 API (Application Programm Interface) 的异步请求代理,在这个 API 上采用了被叫做“WRB 服务”的技术,能够迅速生成动态 Web 页面。Web Listener 把请求传递给 WRB 调度,此时 WRB 调度再去安排请求、执行请求。WRB 应用程序接口被设计成可扩展的,以便可加入第三方厂商产品。编程实现 WRB 执行机制可以是以下类型。

PL/SQL: 执行存储在数据库中的 PL/SQL 过程,它在访问数据库方面比 Java 有优势,但远不如 Java 功能强大。

The Java Interpreter: 在服务器上执行 Java 程序生成动态的 Web Pages,也可以在 Java 中嵌入 PL/SQL 过程。

## 3 应用开发

Oracle WebServer 的应用开发一般有两类组件:

### 3.1 客户端浏览器的 Web

pages: 不管是静态代码还是由 PL/SQL 或 Java 生成的。它们都由用户界面和最终结果组成,最终都存放在 Web 服务器上。

### 3.2 在服务器上的执行代码

有多种代码能在服务器上执行,但它们最终都是生成 HTML 产品。现以 PL/SQL 的开发举例如下:

服务器: ALPHA8200 Digial UNIX Alpha System, Oracle7.2 数据库, Oracle WebServer 2.0

客户端: PC 机, Windows95, Netscape3.0 浏览器

数据库: 所涉及到的表在 xxsc 帐户下

DCD 名称: xxsc 在任一编辑器中编辑如下代码存入文件 test.sql 中。

```
create or replace package ptest as
procedure bytime(vtime in varchar2);
end ptest;
/
create or replace package body ptest as
procedure bytime(vtime in varchar2) is
cursor c1 is select * from hhsq
where sj=vtime;
begin
http.print('<html><title>水情报告</title>
<body><center><h3>'||vtime||' 水情报告</h3><p>
<table border=1 width=80%');
http.print('<tr>
<th>单位</th>
<th>流量</th>
</tr>');
for c1rec in c1 loop
http.print('<tr><td>'||c1rec.dw||'</td>
<td>'||c1rec.ss||'</td></tr><p>');
end loop;
http.print('</table></center></body></html>');
end bytime;
end ptest;
```

```
/
show error
/
```

以 xxsc 帐户进入 Oracle, 在 Sqlplus 下执行 test.sql, 则生成 Package—ptest.bytime。

编写 HTML 的文件 learn.html。(可以使用工具如: Front Page2000)

```
<html><title>黄河水情</title>
<body><h2>黄河水情</h2>
<p>
<h3>请输入查询时间:</h3>
<Form Method="POST" Action=
"http://10.67.123.1/xxsc/owa/ptest.bytime">
<Input Type="text" Name=vtime value="2002/03/012">
<Input Type="submit" value="查询"></Form> </body>
</html>
```

## 4 在浏览器上 URL 的指定方法

### 4.1 直接执行 PL/SQL 的 Package 中的一个过程

http://服务器名:端口号/DCD 名称/OWA/包.过程?参数

如上例: http://slserver.sl.cnpc.co.cn:80/xxsc/owa/ptest.bytime?vtime=2002/03/12

### 4.2 执行 HTML

http://服务器名:端口号/虚拟路径/文件

如上例: http://slserver.sl.cnpc.co.cn:80/ows-doc/learn.html

## 5 高级应用开发

方法:将 Form 调用的过程写入 PL/SQL 过程中,把选择的参数作为变量传给 PL/SQL 执行过程,做到执行过程的循环调用。举例如下:

```
create or replace package hhsq1-search as
procedure lbox(item in varchar2);
procedure bytime(vtime in varchar2 default NULL);
procedure bytimex(vtime in varchar2 default NULL);
end hhsq1-search;
/
create or replace package body hhsq1-search as
procedure lbox(item in varchar2) is
cursor c2 is
select distinct sj
from hhsq order by sj DESC;
begin
for c2rec in c2 loop
if substr(item,1,10)=substr(c2rec.sj,1,10) then
http.print('<option selected>'||c2rec.sj);
else
http.print('<option>'||c2rec.sj);
end if;
end loop;
end;
procedure bytime(vtime in varchar2 default null) is
a1 char(10);
begin
select max(sj) into a1
from hhsq;
bytimex(a1);
end;
procedure bytimex(vtime in varchar2 default NULL) is
cursor c1 is select * from hhsq where sj=vtime;
begin
http.print('<body background="/ows-gif/orhmbkgn.jpg" text=" #
000088"
link="#BB0000" vlink="#BB0000">');
http.print('<html><title>水情报告</title><body><p>
<table border=1 width=80%>');
http.print('<tr>
<th>单位
<th>流量');
for c1rec in c1 loop
http.print('<tr><td>'||c1rec.dw||'</td><td>'||c1rec.ss||'</tr><p>
');
end loop;
http.print('</table></center></body></html>');
http.print('<left>');
http.print('</left><hr>');
end;
end;
```

```
http.print('<hr>');
http.print('<center><h3>'||vtime||' 黄河水情报告</h3><p><table
border=1 width=80%>');
http.print('<font color="#FF0000">(胜利油田有限公司东辛采油
厂)</font>');
http.print('<tr>
<th>单位
<th>流量');
for c1rec in c1 loop
http.print('<tr><td>'||c1rec.dw||'</td><td>'||c1rec.ss||'</tr><p>
');
end loop;
http.print('</table></center></body></html>');
http.print('<left>');
http.print('</left><hr>');
end;
end;
```

## 6 Oracle WebServer 的安全机制

OracleWeb listener 提供了四种安全保护方法:

Basic Authentication  
Digest Authentication  
IP-based Restriction  
Domain-based Restriction

其中 Basic Authentication、Digest Authentication 是验证用户浏览,IP-based Restriction、Domain-based Restriction 是限制 IP 地址和域上网浏览。Basic Authentication、Digest Authentication 两者都是需要验证用户、口令,两者的区别在于后者在将用户、口令由客户端传至服务器时加密传送,而后者则不加密。两种安全保护方式可以交叉使用(使用运算符|或&)。配置方法如下。

### 6.1 listener 的安全保护配置

打开 listener home page,打开要保护的 listener 的 configure 页面,选择安全(Security: Access Control and Encryption)一栏。如果选择验证(Authentication)方式,则需要提供赋予各种访问权限的用户及口令;如果选择限制(Restriction)方式,则需要提供各种访问权限的 IP 地址或域名

指定用户(users)、口令,创建用户的组(groups),创建组的联合(realms)。如果一个组有多个用户,每一个用户用空格隔开,同样,一个联合的多个组,也用空格隔开。创建 IP 组或域组,定义 IP 地址范围或域范围。“+”赋予权限,“-”取消权限。如果“+”、“-”并用,应“-”在前,两者用空格分开,如果多个地址范围有交叉,则第一个生效。如:-10.67.12.204+10.67.12.\* ,定义的 IP 范围应是除去10.67.12.204的所有以10.67.12打头的地址。若+10.67.12.\* -10.67.12.204则后者不生效。指定要保护的文件或目录(虚路径)(如:/或/OWS-DOC/);或(,basic,digest),领域(realm)名;或运算符(,&,|);或(,IP,domain),组名。这样,所有/或/OWS-DOC/下的 html 文件都将受到保护。

### 6.2 WRB 的安全保护配置

打开 Web Request Broker 页面,打开相应 listener 的 modify 页面,选择“Protecting Application”一栏,指定要保护的虚拟路径,如:/xxsc/owa/,其它参数的指定方法同上。这样,所有/xxsc/owa/下的 PL/SQL 过程都将受到保护。

**结束语** Oracle WebServer 是创建带有动态数据库内容的 Web 应用的开发工具,给 Oracle 数据库提供开放性、扩展性的技术支持。用 WebServer 开发出的应用系统,便于维护、扩展,客户端只需要浏览器软件,维护的工作量全部集中在 Server 端。不足的是,目前 Oracle WebServer 的应用开发的

(下转第147页)

begin

在翻译登记表 transtable 中找到与 tansnum 对应的类,记为 Ttransel;  
transel:=Ttransel.create; //产生 transel 对象;  
transe:=@transel; //将 transel 对象指针赋予 transe 属性;  
end;

### 3.3 串行口通讯构架(如图7)

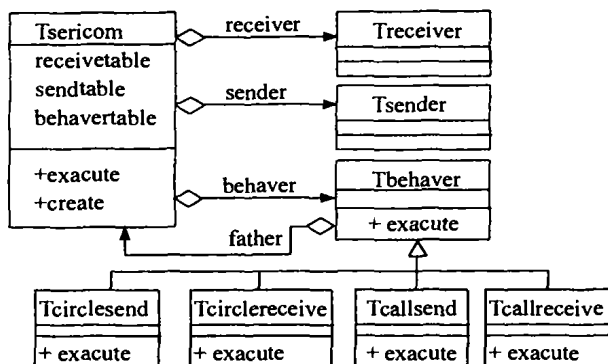


图7 串口通讯类图

Procedure create (receiveinum, sendnum, behavernum, checknum, intransnum, outtransnum)

Begin

根据 receiveinum 在 receiveivable 表中确定其对应的类记为: Treceive1;  
根据 sendnum 在 send table 表中确定其对应的类记为: Tsend1;  
根据 behavernum 在 behavertable 表中确定其对应的类记为: Tbehavior1;  
receive1:=Treceive1.create; //产生 receive1 对象;  
send1:=Tsend1.create (intransnum, checknum); //产生 send1 对象;  
behavior1:=Tbehavior1.create (intransnum, checknum); //产生 behavior1 对象;  
receiver:=@receive1; //将 receive1 的对象指针赋予 receiver;  
sender:=@send1; //将 send1 的对象指针赋予 sender;  
behavior:=@behavior1; //将 behavior1 的对象指针赋予 behavior;  
Behavior ^ . father:=self; //将 behavior1 的 father 属性置为本对象的指针;

End;

Procedure execute

Begin

behavior ^ . execute; //调用 behavior 所指对象的方法;

End;

对于类 Tcirclesend,方法 execute 如下设计:

procedure execute

begin

father ^ . sender ^ . execute; //调用其 father 属性所指对象的 sender 属性所指对象的 execute 操作;

end;

对于类 circlereceive,方法 execute 如下设计:

procedure execute

begin

father ^ . receiver ^ . execute; //调用其 father 属性所指对象的 receiver 属性所指对象的 execute 操作;

end;

对于类 callsend 中的 execute 如下设计:

procedure execute

begin

father ^ . receiver ^ . execute; //调用其 father 属性所指对象的 receiver 属性所指对象的 execute 操作;

end;

(上接第151页)

可视化程度较低,程序调试起来显得不是那么得心应手。

### 参考文献

- 1 宁建民 主编. WEB 数据库开发. 大恒电子出版社, 2000
- 2 阮家栋, 施美雅 著译. WEB 数据库技术. 科学出版社, 2002.

if father ^ . receiver ^ . data ∈ 发送命令 then father ^ . send ^ . execute;  
//若收到发送命令,则发送一帧数据;

end;

对于 callreceive 中的 execute 如下设计:

procedure execute

begin

father ^ . sender ^ . datasendto com = 发送命令; //送出请求发送帧;

father ^ . sender ^ . execute; //调用本对象的 father 属性所指的对象 sender 属性的 execute 操作;

do

father ^ . receiver ^ . execute; //准备接受一帧数据;

until father ^ . receiver ^ . data <> null; //直到接收完毕一帧数据;

end;

关于多串口通讯构架结构图及实现模块可仿照上述给出,这里因篇幅所限,不再具体讨论。

### 4 构件、构架的组装

构件构架的组装分为插合和粘合两种情况。插合是指构件构架的组装不再需要编程,而是通过参数配置的方法,由构件构架内部的机制,实现根据用户的需要自动组合相应的控件。粘合是指通过程序语言将不同的构件构架组合起来。前面所提到的设计方法,能够实现在串口通讯功能内部的插合,至于串口外的程序的数据通讯,可通过数据库或 DDE 实现。因此,本设计方案完全可实现自动插合。

构件构架组合的原则是先将构件组合为小粒度的构架,再由小构架组合为大粒度的构架。本设计方案正是根据这一原则,由大构架提出组合要求,分配给小构架完成。小构架再将任务分配给更小的构架或构件完成。

结论 通过对上面的串行口通讯的分析,可以总结出开发基于构件的可复用程序的开发方法:

(1)全面细致地进行领域分析,区分出哪些行为是变化的,哪些行为是不变的,既找出变化点。

(2)要根据各个变化点性质的不同,确定使用哪种复用方法。

(3)结合前面的分析,确定构件及构架的结构及其组装方式,选择合适的设计模式。

(4)在设计时先设计构件,再设计小粒度的构架,最后设计大粒度的构架。当然也可以先设计整体构架,再将其逐步细化。

### 参考文献

- 1 徐正权. 软件复用方法与技术[M]. 武汉: 华中理工大学出版社, 1998
- 2 王文福, 张世琨, 朱冰. 软件工程[M]. 北京: 北京大学出版社, 1997
- 3 杨美清. 软件复用及相关技术[J]. 计算机科学, 1999(5)
- 4 李景峰, 刘西洋, 陈平. 一种可重用构件模型——类属构件[J]. 计算机科学, 1999(9)
- 5 Ivar Jacobson. Software Reuse [M]. ACM Press, 1997

3 杨浩 编译. JAVA SCRIPT 入门与提高. 清华大学出版社, 2000

4 王燕, 崔雨柏 译. ORACLE WEB 应用培训教程. 机械工业出版社, 1998

5 黄樱, 刘绍中. ORACLE WAS 的安全机制. 计算机应用, 1998

6 傅豫波, 李云静 编译. ORACLE PL/SQL 程序设计. 机械工业出版社, 1999