

PB 与 S-Designor 联合开发应用系统

陈建宏 王 斌 (长沙中南工业大学网络与 CAD 研究室 410083)

摘要:本文对快速开发工具 PowerBuilder 与 CASE 工具 S-Designor 各自的功能进行了介绍,描述了使用两工具联合开发 C/S 模式应用的过程。举例说明了 S-Designor4.2 概念模型、物理模型、触发器、存储过程、反向引擎、扩展属性等功能的开发和用法。

关键词:S-Designor PowerBuilder 概念模型 物理模型

一、PowerBuilder 与 S-Designor 简介

PB 通过对不同数据库采用各自专用的接口或通用的 ODBC 接口,同时支持多种关系数据库。而它自身又内置关系数据库 Sybase SQL Anywhere,因此它可以充分利用来自不同数据源的数据;它使用了面向对象的开发方式,这使得开发人员在无须精通专用语言的情况下迅速转向面向对象的开发;它的核心是数据窗口对象,利用这个智能对象可以操作关系数据库的数据而无须编写 SQL 语句、用户可以直接查询、修改、插入、删除、浏览、打印,并以多种形式(Form)打开和存储数据;同时还为开发人员提供了许多实用开发工具。

S-Designor(以下简称 SDP)是 Sybase 公司推出的 CASE 工具。它是一个基于信息工程方法的数据库建模工具,建模过程分两阶段(如图 1 所示),首先是概念模型设计(Conceptual Data Model(CDM)),此阶段把现实世界中需要保存的信息抽象成信息世界中的实体和关系,从而产生实体关系图,这一阶段可以为组织高质量的应用提供结实的基础。第二个阶段是物理数据模型设计,根据概念模型设计物理数据库结构,SDP 会自动地根据具体的数据库的特征生成物理数据模型(Physical Data Model(PDM)),在此基础上设计人员可以根据需要修改物理模型。

Sybase、Informix、DB2 等),并能产生多种数据库的物理模型,可自动根据物理数据模型生成数据库(包括建表、索引、主键、外部键、触发器、存储过程)的 SQL 语言和完整文档,同时可以实现不同数据库的相互转化;它的另外一个突出优点是可以把已经存在的数据库内容逆向生成概念和物理模型,即可以进行对已存在的数据库的再设计;同时它也支持团队开发,提高设计效率,并且可对模型特定的部分进行设计;SDP 与 RAD 以及其他开发工具有良好的集成,由它设计的结果可以转入 PB 中进行下一步开发。

利用 PB 与 SDP 可以联合开发 C/S 模式应用。在系统分析阶段利用 SDP 作为辅助工具设计系统的数据库,数据库设计完成以后通过 ODBC 接口或者 SQL 脚本语言将数据库移植到具有相应 DBMS 系统的服务器中,而 PB 在客户端开发基于该数据库的图形用户界面。SDP 也给 PB 留有接口,通过 SDP 可以为 PB 进行扩展属性的定义。利用 PB、SDP 及 DBMS 的有机结合可以开发出优良的数据库应用来。

二、用 SDP 生成数据库

利用 SDP 生成数据库可分为三个步骤:①概念模型设计;②物理模型设计;③自动生成 SQL 语言或直接将设计结果移到 PB 和 DBMS 中。下面通过一个简单的例子阐述这一过程。

1. 利用 SDP 生成概念模型(CDM)

我们考虑一个作者与书的模型,其关系是:一个作者写了一本或多本书;每本书有且仅有一个作者。SDP 中将实体间的关系分为一对一的关系、一对多的关系、多对多的关系、递归关系、分类关系及排斥关系等。显然上面作者与书的关系是一个一对多的关系。而对于每一种实体间的关系都有两种选择,分别为可选和强制。考虑作者与书的关系可知:每本书必须有一个且仅有一个作者,



图 1 s-designor 建模过程

SDP 可以支持多种数据库(Oracle、SQL Server、

所以在实体作者端为“一端”并为强制端；而每个作者有一本或多本书，即只要是作者就必须写了至少一本书，所以在实体书端为“多端”为强制端。如果没有写书的‘作者’也称为作者的话，那么在实体书端为“多端”并为可选端，本例只考虑第一种情况。

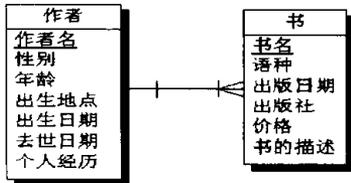


图2 概念数据模型示意图

在SDP的集成环境中，选择新建CDM后，利用放置于面板中的Tools工具箱工具。在面板中放置一个实体图形后双击它进入实体定义(Entity Definition)对话框，在Name栏中键入要建实体名称，例如：作者在Code栏中键入要建实体的代码AUTHOR，之后单击该对话框中Attributes按钮，进入实体属性(Attributes of Entity)对话框后，输入实体‘作者’的属性，即实体字段的名称、代码、类型、大小、是否是标识符等。待确定无误后单击“OK”按钮生成‘作者’实体。同样的过程可生成“书”实体。之后定义两实体的关系，首先单击Tools工具箱工具，然后单击‘作者’实体，按住鼠标左键拖到‘书’实体上，这样在两实体间建立了一个临时关系。双击该关系线进入Relationship Definition对话框，按照前面所述的原则定义‘作者’与‘书’的关系，之后单击“OK”就完成了关系的定义。这时‘作者’与‘书’两实体的概念数据模型已经完成，如果为了美观或者更完整还可以增加一个标题头或者设置窗口的颜色等。生成的概念数据模型如图2所示。从实体属性(Attributes of Entity)对话框中，选定一字段，单击Check按钮进入校验参数对话框(Check Parameters)。之后进行域(Domain)的定义，域实质上是特定字段的取值范围。定义完域后可将其与一个或多个字段相关联。

2. 利用SDP生成物理数据模型(PDM)

由上述步骤生成概念数据模型后，在SDP中选择生成物理数据模型菜单(Dictionary/Generate physical Model)，将弹出生成物理模型(Generating Physical Data Model)对话框，在DataBase Name项后，利用浏览功能选择你要生成PDM的目标数据库名称，本文选择ORACLE 7.0，接着在PDM File Name项中输入你要保存的PDM文件

名。单击“OK”之后概念模型转变为物理数据模型，在SDP自动将概念模型转变成物理模型时，实体、关系、属性转换为‘作者’、‘书’两个表，并自动产生两个表间的关系，如图2所示。



图3 物理数据模型示意图

转化过程中，CDM中的域定义在PDM中完全继承下来，在PDM中还增加了扩展属性的定义。SDP在生成物理模型时自动生成默认的触发器，对于索引，SDP自动对主键建立索引，并且自动对成为主键的外部键建立索引。

3. 利用SDP生成触发器和存储过程及其他功能

SDP除了能完成数据库设计通用功能外，还能满足具体应用的特殊要求，SDP在物理层上提供了丰富的建模功能，包括触发器、存储过程、PB扩展属性定义、视图、索引等。

(1)定义触发器。利用SDP生成触发器，首先要做的工作是改变默认的实体相关完整性，以生成用户想要的触发器。在物理数据模型的面板中双击‘作者’与‘书’的连线，进入参照属性定义(Reference Definition)窗口之后，单击Integrity按钮，这时弹出参照完整性(Referential Integrity)窗口，可以看到默认的触发器为限制删除(Restrict On Delete)和限制修改(Restrict On Update)。选择限制删除和级联修改(Cascade On Update)。由于在建表的时候设置子表外键非空(Not NULL)，所以不选Change Parent Allowed而选择Mandatory Parent。然后选择Dictionary | Triggers and Procedures | List of Triggers就会得到你想要的触发器。或者选择Interface | Generate Triggers and Procedures后进入相应的窗口对For Update和For Delete重新设置，单击Generate Script按钮，生成如下代码的触发器：

```
-- AUTHOR表的Update触发器 tu-author
create trigger tu-author before update of AUTH-
NAME on AUTHOR for each rowdeclare
integrity-error exception ;
errno integer ;
errmsg char(200) ;
dummy integer ;
```

```

found          boolean ;
begin
-- 修改与父表 AUTHOR 关联的子表 BOOK 记录
if (updating('AUTH-NAME') and :old.AUTH-
NAME != :new.AUTH-NAME) then
  update BOOK
  set AUTH-NAME = :new.AUTH-NAME
  where AUTH-NAME = :old.AUTH-NAME;
end if;
-- 错误处理
exception
when integrity-error then
  raise-application-error(errno, errmsg);
end;
/
-- AUTHOR 表的 Delete 触发器 td-author
create trigger td-author before delete on AUTHOR for
each row
declare
  integrity-error exception;
  errno          integer;
  errmsg         char(200);
  dummy         integer;
  found         boolean;
-- 为子表 BOOK 声明删除父表限制
cursor cfk1-author(var-auth-name varchar) is
  select 1 from BOOK
  where AUTH-NAME = var-auth-name
  and var-auth-name is not null;
begin
-- 当子表 BOOK 中有记录时,不能删除父表 AU-
THOR
  open cfk1-author(:old.AUTH-NAME);
  fetch cfk1-author into dummy;
  found := cfk1-author%FOUND;
  close cfk1-author;
  if found then
    errno := -20006;
    errmsg := '当子表 BOOK 中有记录时,不能删
除父表 AUTHOR.';
    raise integrity-error;
  end if;
-- 错误处理
exception
when integrity-error then
  raise-application-error(errno, errmsg);

```

end;

(2)定义存储过程。选择 Dictionary | Triggers and Procedures | List of Procedures 在弹出窗口中选择“new”,即可在新窗口中定义存储过程了。

(3)定义索引及视图。用鼠标右键单击作者表或者书表,在下拉菜单中选择 Indexes,即可进入相应表的索引定义窗口。SDP 已经对每个表的主键及非空外键建立了默认索引,也可以将经常用来作为查询条件的列建立索引。

定义视图的过程很简单,类似 PB 中为数据窗口构建数据源的过程。选择 Dictionary | Views | List of Views 即可进入定义视图的窗口。

(4)定义 PB 的扩展属性。SDP 也提供定义 PB 的扩展属性的功能,通过菜单 Dictionary | extended attributes 即可进入扩展属性定义下拉菜单。选取相应的菜单命令就可以对 PowerBuilder 的编辑风格、显示格式、有效性规则等扩展属性进行定义。

4. 利用 SDP 自动生成 Server 数据库

如上所述,SDP 可以完成从数据库设计到前端 RAD 工具的部分功能,而 PB 如何利用 SDP 的设计成果呢?一般来说有三条途径可循:

(1)使用 Interface | Generate Database 菜单命令即可完成 *.SQL 文件的生成;

(2)SDP 通过 ODBC 与数据库相联,直接在后端数据库中创建表、视图、索引、触发器等;

(3)通过 Interface | 4GL generate,直接在 PB 的系统表中生成表的扩展属性。

三、用 SDP 修改已存在的数据库

SDP 还有一个重要的功能,就是对已存在但不合理的数据库进行重新设计。它是通过一个叫做反向引擎(Reverse Engineering)的接口将后端数据库灌入 SDP 中,在 SDP 中对已有的表进行修改,直到满意为止。之后又可通过 ODBC 接口将数据库“灌回”到后端数据库中。

参考文献

- [1] 侯志平、徐常胜 珠联璧合——PowerBuilder 与数据库配合开发技术,北京晓通网络数据库研究所。
- [2] 陈建宏、王斌 基于 Powerbuilder 前端和 Oracle 后台的数据库完整性和并发控制技术,中南工业大学学报,1998,29(4)。

(来稿时间:1998年5月)