解决方案

利用扩展模型定义对 Sybase 信息流模型进行扩展（一）

1 概述

随着经济全球化的发展，组织结构上的地理分布性特点变得越来越显著。为了提高企业的竞争力，越来越多的企业利用 Sybase 复制服务器来建立分布式的数据库应用，以此来提高服务质量以及保证数据安全。

对于初学者而言，Sybase 复制服务器无疑是一个复杂的产品，如果没有好的开发配置工具的协助，复制系统很难搭建起来。在这种背景下，Power Designer 的信息流模型应运而生。通过对复制过程的抽象，Power Designer 提供了图形化的复制系统开发工具，利用开发工具，极大地缩短了复制系统开发的时间。

随着 Sybase 复制服务器的不断发展，对复制过程的描述也要不断完善。Power Designer 提供了扩展模型定义，来对信息流模型进行补充。

在接下来的章节中，我们将详细介绍 Sybase 复制服务器、信息流模型、扩展模型定义和模板创建语言（Generation Template Language）的基本概念，在文章的最后部分，我们给出一个简单的实例。

2 Sybase 复制服务器和信息流模型的介绍

2.1 Sybase 复制服务器是什么

Sybase 复制服务器是一个基于日志复制的系统，持续的将数据的增量变化而不是整个数据快照进行复制，在数据的提取过程中不会锁定任何数据从而保持数据的可用性，事物被按照原始的提交顺序自动的写入到复制数据库中。复制服务器的内部检测机制保证事务的严格一致性和提交顺序，对事务日志的持续读取和传播使其可以进行高速度和大容量的持续复制。

Replication Server 复制服务器通过复制事务和存储过程调用，将数据在主数据库的增量变化而不是主数据库的所有数据复制到目的数据库，以及在目的数据库上复制存储过程的调用而不是复制在主数据看上执行存储过程产生的数据，在保持数据完整性的同时提供高性能的分布式数据环境。

2.2 信息流模型基本概念

信息流动是企业信息系统的主要活动。Power-Designer 信息流模型是一个图形化建模环境，让用户模拟：

（1）数据复制：数据从源数据库复制到目标数据库，途中经由复制服务器；

（2）数据转换：数据从不同数据源汇合，经过提取、转换，装载到目标数据源。

利用 Sybase PowerDesigner，用户可以：

（1）构建信息流模型，模型包含信息流图（Information Liquidity Diagrams）、转换控制流图（Transformation Control Flow Diagrams）和数据转换图（Data Transformation Diagrams）；

（2）模拟 Sybase 复制服务器的数据复制过程；

（3）产生复制服务器的开发配置文件。

2.3 信息流图

信息流图是信息流模型的核心。利用信息流图，用户可以模拟复制过程，配置 Sybase 复制服务器。

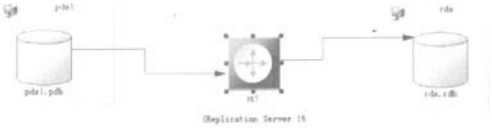

图 1 基本的信息流模型图

2.4 为什么需要对信息流模型进行扩展

到目前为止，Sybase 服务器的最新版本是 15.2，随着版本的升级，越来越多的新特性被加入复制服务器。由于信息流元模型本身的扩展涉及到 Power-Designer 的核心代码，那么提供一种简便的动态扩展方式变得非常有意义。动态性体现在扩展模型定义的按需绑定或卸载；按需自定义扩展模型定义。

3 扩展模型定义

3.1 扩展模型定义基本概念

扩展模型定义可以用来定制和扩展 PowerDesigner 元类（metaclasses）。用户可以为特定的模型类

型定义一个扩展模型定义，但是异构模型之间（不同种类的模型）不能共享同一个扩展模型定义。例如，用户自定义用于Java模型的扩展模型定义，与特定的应用程序服务器、集成开发环境或者对象关系映射框架协作。

PowerDesigner提供了许多现成的扩展模型定义，对于面向对象模型，有以下扩展模型定义：

（1）Hibernate（与Hibernate持久化框架协作，产生相关配置文件和代码）；

（2）Sybase EAServer6.x（与Sybase应用服务器6.x协作，产生一些EJB相关的配置文件和代码）；

（3）其他。

3.2 绑定扩展模型定义到模型

当用户创建或者通过反向工程建立一个新模型的时候，可以通过选择一个或者多个扩展模型定义，并且绑定到新模型上。

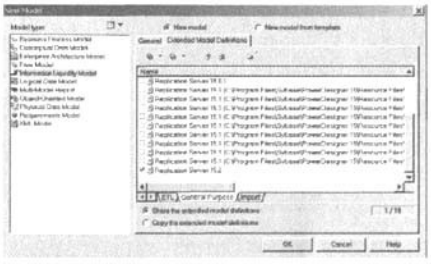

图2 创建信息流模型

用户也可以绑定扩展模型定义到已有模型上。

（1）确保信息流模型已经被打开；

（2）选择模型→扩展模型定义，扩展模型定义列表窗口打开；

图3 扩展模型定义列表

（3）选择导入扩展模型定义按钮，所有可以用于绑定到信息流模型的扩展定义模型显示在列表中，选择一个或者多个扩展定义模型绑定到用户模型中；

（4）选择相应单选按钮；

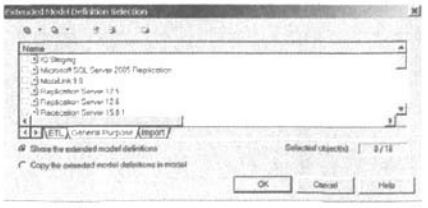

图4 扩展模型定义选择

a.共享：创建一个到扩展模型定义文件的链接。任何对扩展模型定义文件的修改都将被链接到其的所有模型共享；

b.拷贝：创建一个模型私有的扩展模型定义文件；

（5）单击OK按钮。

3.3 创建扩展模型定义

（1）打开模型，选择模型→扩展模型定义，扩展模型定义列表窗口打开；

（2）单击增加一行按钮，输入扩展模型名字；

图5 创建自定义扩展模型定义

（3）单击属性工具条，打开扩展模型定义属性窗口；

图6 扩展模型定义属性窗口

（4）为扩展模型定义文件定义对象扩展，单击确定，新的扩展模型定义被创建并且绑定到模型中。

3.4 导出扩展模型定义

用户可以导出模型中的私有扩展模型定义，使

之被同类型的模型所共享。

（1）打开模型，选择模型→扩展模型定义，扩展模型定义列表窗口打开；

（2）从列表中选择需要导出的扩展模型定义，单击导出扩展模型定义工具；

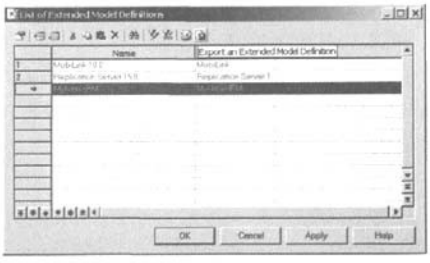

图7 导出扩展模型定义

（3）标准的另存为对话框弹出；

图8 文件另存对话框

（4）输入文件名，选择保存目录，单击保存。扩展模型定义被保持在用户输入的目录中，并且可以被同类型的其他模型所共享。

3.5 扩展模型定义中的 Profiles

PowerDesigner 中的所有的资源文件（扩展模型定义文件是资源文件的一种）都包含一个名叫 Profile 的目录，被用来扩展元模型。下面介绍几种常用的扩展方法。

3.5.1 条件扩展

用户可以在原类上定义条件，当这个原类的实例满足条件时，自动应用条件中定义的扩展，条件的定义可以嵌套，只有当父亲条件和子条件同时满足时，子条件范围下的扩展才会生效。例如，只有当 Database 实例与 复制服务器存在连接（父亲条件），并且这个连接具有读访问时（子条件），Database 实例才可以包含 Replication Agent 配置属性页。

为了创建条件扩展，执行以下步骤：

（1）右键单击原类，选择 New → Criterion；

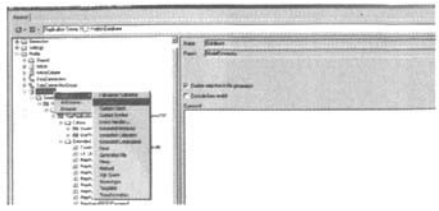

图9 为扩展模型定义增加条件扩展

（2）键入条件的名字，输入条件表达式，如果存在的话，选择父亲条件。

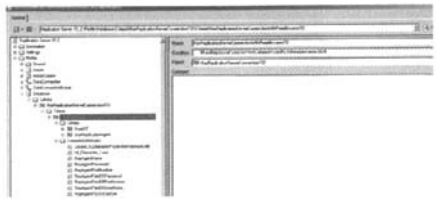

图10 条件扩展属性

3.5.2 属性扩展

用户可以为元类（metaclass）、stereotype 和 criteria 定义扩展属性。为了创建扩展属性，执行以下步骤：

（1）右键单击元类、stereotype 或者 criterion，选择 New → Extended Attribute；

图11 属性扩展

（2）填入合适的属性；

（3）点击 Apply 按钮保存。

3.5.3 表单扩展

用户可以使用表单去创建新的或取代已有的属性页。通过创建自定义表单，用户可以利用更加合

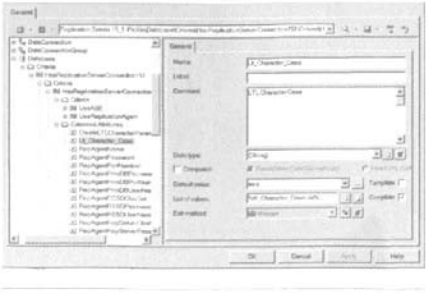

图 12　属性扩展定义

理的方式组织属性。表单可以在原类（Meta Class），类型（Stereotype）和条件（Criterion）上面进行扩展。

创建表单，执行以下步骤：

（1）右键单击原类，类型或者条件，选择 New → Form；

（2）输入表单名称，表单类型以及创建表单布局。表单类型包括以下 3 种：

图 13　表单扩展

a.属性页：创建一个新的属性 tab 页；

b.取代标准的 tab 页；

c.对话框：创建一个对话框，对话框可以通过菜单或者一个按钮来启动；

（3）使用工具条插入并且安排扩展属性的布局；

（4）单击 Preview 按钮预览表单布局。

图 14　表单扩展属性

3.5.4　模板和产生的文件

用户可以利用 PowerDesigner 的模板定义语言（Generation Template Language）为原类产生文件（脚本）。模板 (templates)和产生文件（generated file）可以应用于原类（meta class），类型（stereotype）和条件（criteria）。

执行以下步骤创建模板：

（1）右键单击原类，类型或者条件，选择 New → Template；

（2）键入模板名称和注释；

（3）使用 GTL 创建模板主体。

执行以下步骤创建产生文件（Generated Files）：

（1）右键单击原类，类型或者条件，选择 New → Generated File；

（2）键入文件名称，选择文件类型；

（3）键入用来产生文件的模板。

执行以下步骤在产生文件中引用模板：

（1）打开资源文件编辑器，选择产生文件；

（2）在编辑区域键入引用的模板名称，模板名称用两个百分号括起。例如：%myTemplate%。

图 15　产生文件

4　GTL 介绍

Sybase PowerDesigner 使用模板为模型对象生成相应的源码和配置文件，而模板是由模板定义语言来描述的。每个模板必须与特定的元类(metaclass)关联，一个元类可以关联多个模板。当用户为模型生成脚本时，PowerDesigner 遍历所有包含产生文件（generated file）的对象，应用产生文件中定义的模板文件，生成相应的脚本。

4.1　访问对象属性

在 GTL 语法中，模型对象的属性(包括标准属性和扩展属性)被当作变量来对待。通过把属性名称置于两个百分号之间来访问对象属性，例如：%variable%。

（1）.样例模板：This file is generated for %Name%. It has the form of a %Color% %Shape%；

（2）模板执行输出：This file is generated for MyObject. It has the form of a Red Triangle。

4.2 定义局部变量

通过宏.set_value 来定义局部变量。

例如：.set_value(_characterCase," default",new)，定义一个名为 _characterCase 的局部变量，初始值为" default"。

4.3 作用域

计算模板时，PowerDesigner 首先需要了解的模板所处的作用域。作用域在模板计算过程中可以变化，在任何时刻，只有一个对象在作用域中是有效的。

模板的初始作用域位于模板定义的元类，这个元类的活动对象上的所有属性、集合以及对象的父亲此时都是可见的。用户可以使用“·”符号来改变作用域。有 3 种类型的作用域：

（1）对象作用域：为了访问一个非活动对象的成员，需要指定对象作用域；

（2）集合作用域：为了访问集合的一个成员，必须指定集合作用域。例如：假设目前的作用域位于 table 级，%Columns% 就表示一个集合作用域，而 %Columns.First% 就表示对象作用域，如果用户需要访问第 1 个列的名字，则可以使用 %Columns.First.Name%；

（3）外层作用域：既然可以使用“·”符号来进入更深层次的作用域。同理，也可以通过 Outer 关键字来返回到上层作用域，使用规则如下：

a.当一个新的作用域产生的时候，旧的作用域成为新的作用域的 outer；

b.当一个作用域退出时，outer 作用域成为活动作用域。

下面例子使用 Class 模板介绍了作用域机制如图 16。

4.4 遍历集合子对象

在物理数据模型中，表包含多个列。在面向对象模型中，类包含多个属性。为了迭代这类相关联的子对象，使用宏.foreach_item。

例如：

```
Table %Name% contains the following columns:
.foreach_item(Columns)
  \n\t%Name%
```

输出：

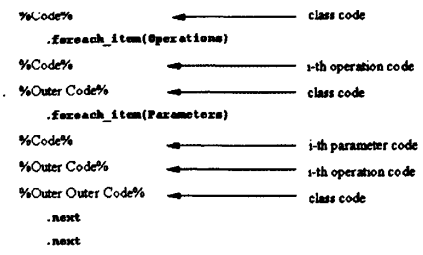

图 16 嵌套作用域

```
Table Customer contains the following columns:
  UserID
  UserName
  Birthday
```

4.5 集合成员

每个对象可以包含一个或者多个集合。PowerDesigner 提供了 3 个集合成员。

（1）First：返回集合中第 1 个元素；

（2）IsEmpty：用来测试集合是否为空，返回布尔类型的值；

（3）Count：返回集合中元素个数。

例如：假设目前 template 所处的作用域在某个数据库表级。%Columns.Count% 表示表格所包含的列数。

4.6 格式化输出

可以通过嵌入格式化选项来格式化变量输出。语法如下：

```
%.format:variable%
```

PowerDesigner 提供了许多现成的格式化选项，例如：

L：把变量中的字符转换成小写。

```
%.L:variable%
```

具体的格式化选项，请参阅 PowerDesigner 手册。

4.7 条件语句

条件语句语法如下：

```
.if [not] condition
  Complex-template
  [(.elsif [not] condition
      Complex-template)*]
  [.else
   Complex-template]
   .endif
```

文/赛贝斯软件上海研发中心 邓 学

（未完待续）