

面向对象数据库技术及其应用

——Ontos: 集成对象系统

陈文萍 徐宏哲

董丽丽

(西安交通大学)

(西安建筑科技大学)

陈文萍, 徐宏哲, 董丽丽. 面向对象数据库技术及其应用——Ontos: 集成对象系统. 石油仪器, 1999, 13(3): 34~37

摘 要 Ontos 是一个在扩展 C++ 编程环境下集成的面向对象的数据库管理系统。被存储在数据库中的持久对象, 例如实体, 可以在 C++ 应用程序中访问和控制。Ontos 对象基础包括用户自定义实例对象。每个对象作为一个实体封装了描述数据和访问及控制实例内部状态的操作, 所有这些与数据库实例相联的公共特征(以操作和属性形式提供的公共特征)可以被 C++ 应用程序调用。

主题词 数据库 数据管理 计算机程序 面向对象 Ontos

作者介绍 陈文萍 研究生, 1976 年生, 1997 年毕业于西安交通大学计算机系, 现在西安交通大学计算机应用专业攻读硕士学位, 曾应用面向对象技术参与开发了电力行业管理信息系统。邮编: 710049

Ontos 应用程序基本体系结构

Ontos 是一个提供 C++ 应用程序接口的面向对象的数据库。数据库对象可以被“激活”, 这就意味着可以把对象状态从二级存储数据库传输到 C++ 应用程序虚拟内存, 相反, 可以解除对象的激活状态, 包括把它传输回数据库。在激活状态和非激活状态之间, 处理过程是通过标准的 C++ 编程实现的。

Ontos 对象数据库是一个分布式数据库, 并且是一个客户/服务器体系结构的数据接口。服务器端管理数据存储, 客户端提供对用户处理过程的接口并负责应用程序虚拟内存空间的数据映射。客户和服务端可以在局域网上交互。Ontos 的客户/服务端体系结构如图 1 所示。

为了用 Ontos 实现大型强调数据处理的应用程序, 可以依赖于集成对象系统的三个主要组件:

(1) 核心数据库 Ontos 数据库内部体系结构是面向对象的。而且, Ontos 系统有一套基类、核心类。这些类可以被分为两大类: 一是描述 Ontos 系统的类, 被称为“内部模式”; 二是通过提供一些基本功能帮助开发应用程序的类。

(2) 类库和头文件 核心库包括描述核心类的数据库对象。类库和头文件提供写一个使用核心类

实例应用程序所需的支持。每个类都有一个与之相联系的头文件, 在应用程序中使用这些类时必须用 C++ 的 #include 声明。

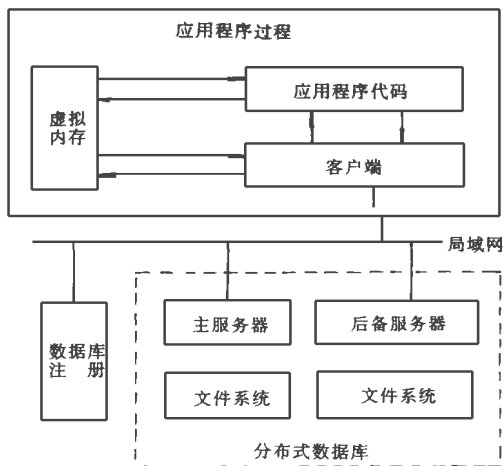


图 1 Ontos 的客户/服务体系结构

(3) 实用程序 它们是对开发大型应用程序的补充支持, 由各种软件工程工具, 例如浏览器、编辑器等组成。因为我们更加关注面向对象语言和数据库概念, 这些工具就不在这里讨论了。

Ontos/ C++ 中的类型定义

Ontos 在概念模式上的类型定义可用图 2 来描

述。这个二维几何物体的概念表示包括广义性表示, 它可以在 Ontos 中通过子类/超类的概念操作。一个几何形状可以是椭圆、多边形、或是其它几何形状, 圆是一种特殊的椭圆; 多边形可以是三角形、四边形等等。这个广义层次根据二维几何物体不同的几何形状区别它们, 这也将导致它们不同的表示和行为。例如, 多边形可以最自然的用顶点确定形状, 而椭圆的自然表示依赖于中心和两个半径的值。在每个对象类型的层次中都会提供一些属性, 可以由它的后代继承。

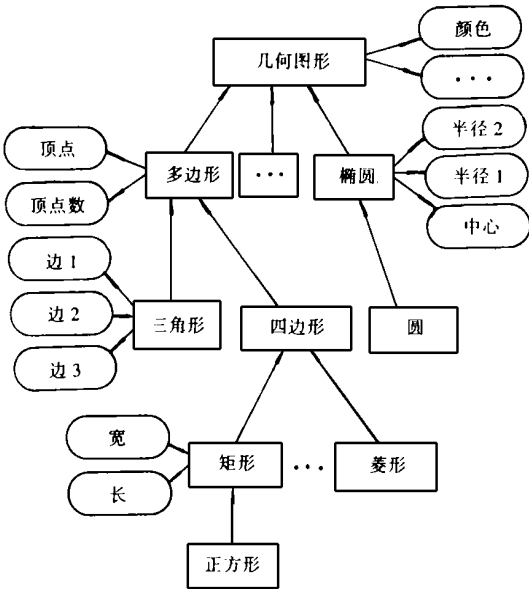


图 2 二维几何图形层次

我们由在 C++ 中的类 GeometricShape 开始, 它在 Ontos 中使用如下:

```
class GeometricShape : public Object {
protected :
    char *color;
    ...
public :
    ...
};
```

第一行指明了 GeometricShape 是 Object 的子类, Object 是 Ontos 的固定类型。Object 是所有持久类的父类, 也就是说, 只有 Object 的直接或间接子类对象有可能归于 Ontos 数据库。Object 为数据库中的存储对象定义成员函数, 并且删除数据库中已删除对象的函数, 也为对象名和数据库中对对象群定义属性。

为了把我们的概念几何模式映射到 Ontos 类型定义上, 可以通过为指定类型引进子类的方法编写每个普遍抽象的代码。这些子类继承超类的所有特征, 也就是一般类型。这样, 我们的代码为

```
class Polygon: public GeometricShape {
protected :
    int noVert;
    List *vertices;
public;
    ...
};
```

这个类型定义使 Polygon 成为 GeometricShape 的子类, 这样 Polygon 可以继承 GeometricShape 的属性, 例如 color 等。另外, 每个 Polygon 获得 noVert 属性, 它存储了多边形的顶点数, 而 vertices 指向一个包含所有顶点的列表集。

属性 vertices, 从有超过一个对象与此属性相连的意义上讲, 它是一种集合属性。在本例中, 它由一个 Vertex 实例列表组成, 更精确的讲, 是由一个指向 Vertex 实例的列表组成。

访问持久对象

1. 对象的命名

Ontos 的类对象为数据库中持久对象的命名提供函数。在某种程度上, 类似于 GOM 中的持久变量。一个对象名可以作为指向数据库的路径。这样, 应用程序可以通过名字或重新获得的对象的关联访问下一层对象。Ontos 不强调单一的平板式命名空间, 而是提供了层次命名空间, 称为目录。一个对象的名字可通过由根目录开始的查找路径指定。

一个访问特定的叫做“myFavoriteEllipsis”的 Ellipsis 对象的范例如下所示:

```
...
Ellipsis *myEllipsis= (Ellipsis *)OC—lookup(“myFavoriteEllipsis”)
...
```

在这个声明中, 变量 myEllipsis 被赋值为通过执行 OC—lookup 程序从数据库获得的 Ellipsis 对象。这个程序把获得的对象名作为参数——可能包括层次目录结构的查询路径。OC—lookup 调用以所谓的投射操作为前缀——即在本例中的(*Ellipsis), 以保证关联 Ellipsis 对象的 C++ 的虚拟内存通过 OC—lookup 返回。OC—lookup 是为 Object 类抽象定义的操作。

比较典型的对象名字在初始化时设定, 也就是说, 它是构造器操作的一部分。例如, Ellipsis 可以被如下定义:

```
class Ellipsis : public GeometricShape {
protected;
    float radius1, radius2;
    Vertex *center;
public :
    Ellipsis(float r1, float r2, Vertex *c, char *theName);
```

```
Float getRadius1();
Float getRadius2();
...
```

};

公共成员函数 Ellipsis 初始化域 radius1 为 r1, radius2 为 r2, center 为 c, 实现留给读者作为练习。而且, Ellipsis 的 Name 被置为 theName, Name 是从根类 Object 继承的属性。Ellipsis 可以被创建如下:

```
MyEllipsis = new Ellipsis(5.0, 6.5 someVertex, "myFavoriteEllipsis");
```

这里, 我们假定 someVertex 是关联 Vertex 对象表示 Ellipsis 的 center 的变量。

2. 持久对象的激活和非激活

对象从数据库传到应用程序的虚拟内存并且通过操作实现激活和非激活。

(1) 对象的激活

激活初始化对象并把它从数据库调入内存, 从而把所有对象关联中包括的激活对象, 让它们指向虚拟内存, 如果这些对象已经被激活。而且, 所有虚拟内存中的对象, 如果其指向新激活的对象也需要改变, 它们的指向变为一个新的虚拟内存指针。

激活一个对象可以影响应用程序的性能。如果有太多的对象被激活, 很多磁盘的读写操作都浪费了, 因为这些对象从未被访问过。另一方面, 每次对象移动时检查关联的对象是否进入内存加重了操作的复杂性。而且, 在 Ontos 中对象的激活是由编程人员决定的——在这种假定下, 编程人员最好决定那个对象最有可能在应用程序中访问。

对象被激活有下列三种不同的情况:

1) 单个对象可以用 OC—lookup 操作激活。而单个对象可以通过方向或抽象关联的移动激活。

2) 一个集合中的所有对象——例如一个列表或一个集合——可以仅通过一个操作 OC—getCluster 的调用激活, 它要求集合的名字作为参数。

3) 通过一个特定的对象可以访问的所有对象可以通过操作 OC—getClosure 激活。在这种情况下, 一定要保证所有对象的关联可以安全的移动, 因为要保证虚拟内存指针的有效。

这两种技术——单一对象和逻辑簇(集合)的激活, 在大多数应用程序中很有效。它们可以约束在处理阶段有可能用到的对象的激活。但是, 这两种技术留有至少某些关联不能解决的缺点。

相反, 激活关联一个应用程序所有入口是最安全的技术, 因为它可以保证所有可访问的对象都在虚拟内存中, 而且, 解决了所有虚拟内存的关联问

题。但是, 考虑到昂贵的磁盘读写和对象状态移动代价, 读取可能被禁止。在最坏的情况下, 它包括读取整个数据库。图 3 描述了黑色对象群的激活。所有阴影对象被隐含激活, 因为它们通过对象间参照链是可达的。白色对象不能通过黑色对象的移动访问, 因此不能被激活。

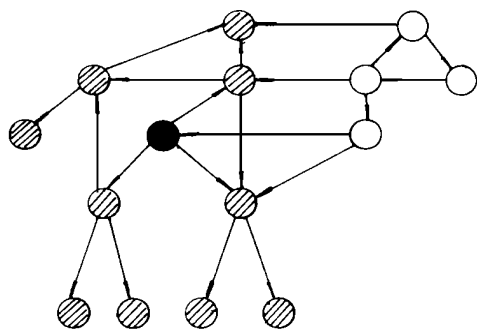


图 3 激活“黑色对象”群

(2) 对象的非激活

要使一个对象变成非激活状态用下面的操作:

```
void Object::putObject();
```

例如, 指向 Ellipsis 的 C++ 变量 myEllipsis 可以按下面的方法变为非激活状态:

```
myEllipsis->putObject();
```

在当前事务成功完成后, 将在数据库中创建一个 Ellipsis 对象的新拷贝。在事务结束后, 调用 putObject 把所有需要存储的对象放入数据库中是程序员的责任。忘记调用 putObject 将导致应用程序对特定对象操作结果的丢失。

在 Ontos 中, 对 Ellipsis 调用 putObject 不会导致关联对象 Vertex 的存储。它必须直接执行, 例如:

```
myEllipsis->getcenter->putObject();
```

但是, 可以用 putCluster 使整个一个集合变为非激活状态, 如同下面的语法:

```
void Aggregate::putCluster(...);
```

非激活也可以通过从数据库中删除一个或整个簇对象来隐含完成。操作的语法如下

```
void Object::deleteObject(...);
```

```
void Aggregate::deleteCluster(...);
```

在这个例子中, 对象没有写回数据库, 但是它们起始的数据库状态已经改变。

事务控制

对象的激活和非激活隐含地假定了一个单一用户。在一个有很多当前用户的数据库, 必须检测和解决用户间的冲突。Ontos 通过下面讨论的事务机制实现这样的控制。在大多数事务机制, 它保证改

变的原子数,也就是说无论包含一个事务的改变提交还是未提交,随着时间的过去,应用程序把对象写进数据库。但这些变化并不立即写入数据库中,而是先放入事务缓冲池中,当事务成功的提交后,再完整的写入库内。但是,在事务内这些行为是不可见的。应用程序在自己私有数据库的状态下运行,这对于外部事务不可见,直到事务提交为止。

1. 基本事务

Ontos 的上锁机制为上锁对象提供了三个等级。所有的对象激活要调用 OC-lookup 或 OC-getCluster 等,这有一个上锁方式问题,锁方式是可选的:可以指定读取锁、写入锁、或根本就不上锁。如果没有指定,则缺省为 readlock,它是隐含假定的。锁的授予遵循通常的锁管理机制。

事务机制接口包含三个函数:

(1)TransactionStart: 事务开始调用定义了事务范围的开始。它还定义了几个参数。例如,它指定:

— 事务名(用于共享事务);

— 处理在事务过程中可能出现的读写对象冲突的功能。因为如果发生锁冲突,处理动作可能随不同情况而变化(一个冲突可能引起应用程序放弃当前事务,等待并试图不久获得锁或简单的没有获得对象而继续处理),这个功能可以是系统提供的功能之一或由用户提供。

(2)TransactionCommit: 在事务结束时调用,并且试图把所有非激活对象写入数据库。

(3)TransactionAbort: 在事务结束时调用,但是不改变数据库。

2. 嵌入事务

事务可以嵌入创建原子事务组成更长的事务(也就是说,多个没有插入提交或放弃的 TransactionStart 调用被激活)。在初始化时,子事务的状态正是父事务的状态。在子事务结束时,如果提交,则父事务的状态包括子事务的所有改变,如果放弃,则不包括任何改变。而且,直到父事务成功提交,子事务的改变对外部父事务不可见。

嵌入事务使长事务更加实用。一个单独的长事务可以被分为一系列小的嵌入事务。然后,如果激活时发生数据冲突,只有嵌入事务需要放弃,而已经完成很长的事务不会受到影响。

3. 共享事务

协作处理可以共享一个事务。这个便利允许处理从数据库观点看来是原子的但可以很容易由一些

协作处理实现的任务。例如,考虑多窗口应用程序中,各个处理分别操作每个窗口。

事务名字用来使一个正在进行的事务对另一个事务“可连接”。第一个过程指定一个特定名字参数调用 TransactionStart 初始化这个事务。接下来的具有相同名字的过程调用 TransactionStart 只是加入已存在事务。共享事务可以认为是一种事务数据池对所有共享事务中的过程都可见的事务。数据池存储每个对象最近的状态,当一个过程激活一个池中已有的对象,它获得了池中对象的副本,而不是数据库中对象的副本。一个过程可以随时更新池中对象的缓冲区副本,这样就可以立即访问由其它协作过程最近所做的工作。

共享事务把所有参与的过程所做的工作“捆”进一个原子单元。一个共享事务被提交,它所做的改变仅为所有过程都成功的提交。一个过程的 TransactionAbort,则会放弃整个共享事务。

结束语

Ontos 的版本和选择性机制允许一个对象在任何版本中存在。同一个版本的所有对象被收集进 configuration 对象。配置通过 derivation links 相互联系。每个配置(除了第一个)有一个父配置,也可能有几个子配置。这样,选择性和连续版本都是支持的。父子配置的对象可以任意修改,处于内部配置的对象不能更改。但是,内部配置可以被删除。当一个内部配置被删除,它的父配置继承被删除配置的所有子配置。应用程序通过新配置和已有的先前配置指定一个派生连接来创建新配置。新配置被创建时,由它的父配置中的所有对象的状态组成。对象被隐含地写进配置中。通过派生,一个对象传给当前激活配置。这样,单独对象的版本是不可表示的,而配置是可以表示的。

参 考 文 献

- 1 Sertrag Khoshafian. Object-oriented database. New York: John Wiley, 1993
- 2 J. G. Hughes. Object-oriented database. New York: Prentice Hall, 1990
- 3 蔡希饶, 陈平. 面向对象技术. 西安电子科技大学出版社, 1993
- 4 杨行建等. 面向对象技术与面向对象数据库. 西安: 西北工业大学出版社, 1994

(收稿日期: 1998-12-06 编辑: 李来顺)

Jiang Lei, Bei Yuangen, Li Qiaolin and Shi Wenhui. The theory of γ -radioactivity water cut meter and its practice. PI, 1999, 13(3): 16 ~ 18

Nowadays, in measuring water cut of on-line crude oil, as a match technology, isotope γ -radioactivity measurement is not only trend to perfection in theoretical model but also more perfect in corollary supputation. It has excellent stability, high accuracy and wide measurement scope. The water cut of crude oil is determined by establishing three phases model of oil, gas and water. practices in spot are described in the paper.

Subject Words: gamma ray, crude oil, water cut meter, measuring, application

Tian Jingwen and Gao Meijuan. Sensors composing by use of Hall switch UGN3020. PI, 1999, 13(3): 19 ~ 21

UGN3020, a new integrated Hall switch element, is used to compose three type sensors: analog type liquid level sensor, digital type liquid level sensor and pulse type rotary speed sensor. The sensors have low temperature draft, simple in principle and widely use. The construct principle and features of the sensors are introduced.

Subject Words: transducer, Hall effect, switch, liquid level, rotary speed, measuring, Hall switch component UGN3020

Zhang Yongjun. An Automatic time shooting system for electroseismic prospecting. PI, 1999, 13(3): 22 ~ 24

Auto-time shooting system is a shooting control equipment to design for electroseismic prospecting. A notebook computer is used as the control center, which makes the system easy to carry, safe to operate, high interactivity and short detonating delay. The paper mainly introduces the circuit functions and an idea of hardware design, including of optic-coupling isolation, decoding logic and drive control and so on, as well as software functions. It will satisfy the requirements of detonating control in electroseismic prospecting to successfully develop such the system, and which will makes it possible for dynamite source activation to be introduced into non-seismic prospecting.

Subject Words: electromagnetic exploration, shofiring, interface, electroseismic prospecting

Cui Ying, Liu Guohui and Huang Shengli. General development of disc brake system used in oil rig. PI, 1999, 13(3): 25 ~ 27

The paper provides the statistical information about the latest development of disc brake either at home or abroad, analyses the advantage and disadvantage of disc brake. The application of disc brake in drawworks has become a major tendency. In view of disc brake development in recent years, the paper proposes some proposals on disc brake system.

Subject Words: drilling rig, brake, developing trend, proposal

Chen Wenping, Xu Hongzhe and Dong Lili. Object-oriented database technique and its application —— Ontos: integrated object system. PI, 1999, 13(3): 34 ~ 37

Ontos is an object-oriented database management system which is integrated in an extended C++ programming environment. Persistent objects, i. e. , entities that are stored in the database can be visited and manipulated within a C++ application program. The Ontos object base contains objects that are instances of user-defined types. Each object as an entity encapsulates the describing data and the operations to visit and manipulate the internal state of the instance. All the public features associated with a database entity (providing in the form of operations and attributes) can be invoked by the C++ application program.

Subject Words: database, data management, computer program, object-oriented, Ontos

Gao Meijuan and Tian Jingwen. Communication interface between 80C552 microcontroller and 586 personal computer. PI, 1999, 13(3): 38 ~ 40

80C552 microcontroller has a full duplex series interface SIO0. The special interface can be interfaced with RS-232C by varying voltage. Portable or intelligent instruments used 80C552 as central processor have data readback and network data transmission functions. The interface method between 80C552 and 586 computer is introduced, and the two part software programs are designed in this paper. The two programs are good practice.

Subject Words: microcomputer, communication, computer program, interface, 80C552 chip computer

Qu Fengshan, Wang Xiaoling and Liu Feng. Performance analysis and application of the parallel computer. PI, 1999, 13(3): 41 ~ 44

The paper analyses major factors affecting performance and stability of the paragon parallel computer system. With combining closely seismic data process application software, the paper presents schemes of adjusting computer system performance and improving system stability. The limitations of performance and stability to be considered are provided while extending paragon parallel computer, to guide sciencely paragon parallel computer system multiple extension.

Subject Words: parallel computer, seismic data processing, stability, analysis, computer application, paragon