基于 Qt 图形库 Ingres 空间数据库驱动的实现①

Realization of Ingres Spacial Database Driver Based on Qt Library

佘静涛　田贤忠　陈长军　(浙江工业大学 信息工程学院 浙江 杭州 310032)

摘　要： 针对在 Qt 图形库无法直接操作 Ingres 开源空间数据库中的数据的问题，本文通过研究 Qt 图形库驱动和 Ingres 的 OpenAPI 函数,利用 Ingres 空间数据库提供的 OpenAPI 接口函数在 Qt 图形库中实现了对 Ingres 空间数据库的操作。通过修改 Tora 开源数据库管理软件的源码,使其能够采用 Qt 图形库来操作 Ingres 数据库。

关键词： Tora Qt 库 Ingres 数据库 空间数据库 OpenAPI

1　引言

GIS(地理信息系统）是综合分析空间数据的一种技术系统，它包括空间数据的采集、存储、管理和分析。数据是地理信息系统的重要组成部分，地理信息系统中的数据包括空间数据和非空间数据。空间数据是与空间地理位置相关的数据，其中包括矢量数据，栅格数据等。GIS 的快速发展，是与空间数据库息息相关的[1]。

空间数据库的研究始于地图制图与遥感图像处理领域，其目的是为了有效利用卫星遥感资源迅速绘制出各种地图。由于传统数据库在空间数据的表示、存储和检索上存在许多缺陷，从而形成了空间数据库这一新的数据库研究领域。[2]作为具有代表性的空间数据库 Ingres 已经引起了越来越多人的关注。在 Ingres 基础上产生了很多商业数据库软件，包括 Sybase，Microsoft SQL Server， NonStop SQL， Informix 和许多其他的系统。在 80 年代中期启动的后继项目 Postgres ，产生了 PostgreSQL， Illustra。到今天为止,Ingres 数据库系统已经发展到 Ingres2006 版，其功能特性、体系结构及其在空间数据处理方面都取得很大的进展。因此可以这么说，Ingres 是当代最有影响的计算机研究开源项目之一。

Qt，作为商业应用程序的核心，用于高性能的跨平台软件开发。除了拥有扩展的 C++ 类库以外，Qt 还提供了许多可用来直接快速编写应用程序的工具。然而 Qt 图形库无法直接操作 Ingres 开源数据库中的数据。针对此问题，我们通过研究 Qt 图形库驱动和 Ingres 的 OpenAPI 函数，利用 Ingres 数据库提供的 OpenAPI 接口函数在 Qt 图形库中实现了对 Ingres 数据库的访问，为 Ingres 数据库系统的广泛应用提供了支持。

2　Ingres开源数据库

Ingres，意思是交互式图形获取系统，以 C 高级编程语言为基础。Ingres 是比较早的关系数据库系统，开始于加利福尼亚大学柏克莱分校的一个研究项目，系统代码使用 BSD (BSD 是 Berkly Software Distribution 的简写)许可证。Ingres 数据库不仅能管理数据，而且还能管理知识和对象。同时 Ingres 数据库系统的数据模型简单，容易学习和使用、服务完备，可适用于各种环境等优点。Ingres 对推动关系数据库系统的发展做出了较大的贡献。

2.1 Ingres 空间扩展

Ingres/Spatial Object Library 是 Ingres 空间对象管理的扩展。Ingres 空间数据扩展归纳为 4 部分：空间数据类型的扩展、空间操作符扩展、空间操作函

① 基金项目:国家 863 重点项目(2007AA120400)
收稿时间:2008-08-13

数的扩展和自定义空间扩展。空间数据类型、空间操作符和空间操作函数结合起来定义在 DBMS Server 中，DBMS 能够理解并处理空间数据的多维信息。

Ingres 支持的空间数据类型包括点、线、面、圆、线段和矩形等 6 种空间数据类型。除此之外，还有一种特殊的空间数据类型 nbr，主要用来 r 树索引存储。

Ingres 空间操作符是对上述空间类型与自定义空间数据类型的操作，主要包括相等运算符(Equality Operators)、二值空间操作运算符(Binary Spatial Operators)、Nbr 函数(Nbr functions)和 Hilbert 函数。

Ingres 空间操作函数支持空间数据类型与空间操作符，包括空间函数(Spatial functions)、空间转换函数(Spatial conversion functions)和数据类型转换函数(Type conversion functions)三部分。

Ingres/Spatial Object Library 不仅实现了 Ingres 数据库管理系统在空间操作上的扩展，而且为用户提供了空间操作扩展接口，为 Ingres 数据库系统在地学领域中的应用提供了基础。

2.2 Ingres 的 OpenAPI 函数

开放应用程序接口(OpenAPI)是一组 C 语言的函数，OpenAPI 简化了在多接口，多协议和多环境下开发应用程序的任务。它为访问数据库数据提供了单一的接口。开发人员只需要关注你的应用程序要访问什么数据，而不必关心如何去访问数据的问题。这些函数使你能够编写出能访问 Ingres 和非 Ingres 数据库的应用程序 。OpenAPI 中的 C 函数集能使应用程序连接到 DBMS 服务器上，执行 SQL 语句，获得结果。

OpenAPI 能跟以下组件通信:① Ingres 服务器;②Ingres?联合数据库支持；③Ingres 企业数据库产品，可以为多个 Ingres 数据库和非 Ingres 数据库提供访问[4]。

3 Qt图形库系统

3.1 Qt 图形库简介

Qt 是一个多平台的 C++图形用户界面应用程序框架。它提供给应用程序开发者建立艺术级的图形用户界面所需的所用功能，用于高性能的跨平台软件开发。除了拥有扩展的 C++ 类库以外，Qt 还提供了许多可用来直接快速编写应用程序的工具，如 QtDesigner 工具可以用来直接生成窗体，QtLinguist 用来将应用程序翻译成本地语言。此外，Qt 还具有跨平台能力并能提供国际化支持，这一切确保了 Qt 应用程序的市场应用范围极为广泛。如 Google Earth, Skype, Opera，Quantum GIS 等都是用 Qt 开发的。

自 1995 年以来，Qt C++ 框架一直是商业应用程序的核心。无论是跨国公司和大型组织(例如:Adobe、Boeing、Google、IBM、Motorola、NASA、Skype)、还是无数小型公司和组织都在使用 Qt。Qt 的类功能全面，提供一致性接口，更易于学习使用，可减轻开发人员的工作负担、提高编程人员的效率[5]。Qt 的框架图如图 1。

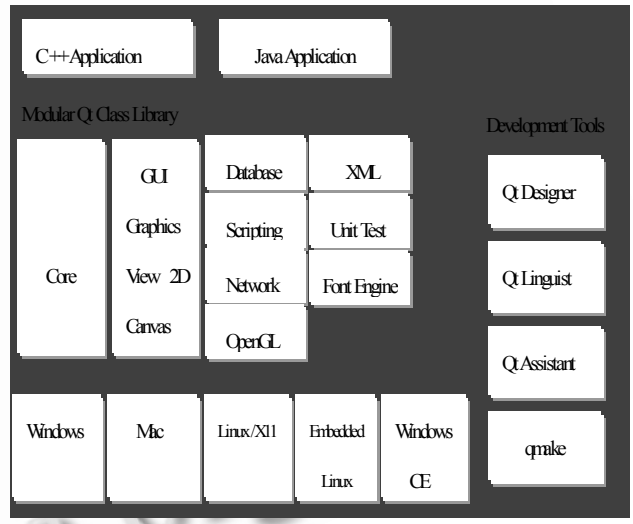

图 1 Qt 的框架图

3.2 Tora 开源软件简介

Tora(Toolkit for Oracle)开源软件起初是由 Henrik Johnson 在 2000 年编写的，是使用了 Qt 图形库来编写的，用来在 Linux 下对 Oracle 数据库进行操作的软件。一开始的时候，TOra 的设计就是基于插件的，在不需要修改既有代码的同时很容易扩展新的功能，最初发布的版本包括工作台和数据库浏览器。接着，作者开始进行 PL/SQL 调试器的开发，尽管需求并不是很迫切。这也是 TOra 区别于其他开源项目的第一个新的特性。后来是 TrollTech 发布了 Qt Free for Windows 的第一个版本(使用 MingW 编译)，使得 TOra 可以方便的移植到 Windows 平台。[6]

4　QT图形库的Ingres数据库驱动的实现

4.1 Ingres 驱动方案的设计与实现

经过研究发现实现 Qt 数据库驱动最简单的方式是采用 plugin 方式来扩展，plugin 是动态链接库，是在运行的时候动态加载的，而不是一个独立的应用程序。如果要采用 plugin 方式来实现数据库驱动必须对 SQL 的模块架构有所了解，其架构主要分为以下三层：(1)用户接口层。这些类提供了基于数据的窗口部件，这些窗口部件不仅连接数据库还可为用户所浏览。(以 QSqlCursor 作为数据源)。终端用户通过这些组件来浏览与编辑数据。Qt 设计器集成了这些类并可用来创建基于数据的窗体。这些窗口部件也可在程序中与你的 C++代码直接交互。支持这一层的类包括 QSqlEditorFactory、QSqlForm、QSqlPropertyMap、QDataTable、QDataBrowser 和 QDataView；(2)SQL 应用编程接口层。这些类存取数据库。QSqlDatabase 类用来连接数据库。数据交互的实现要么通过 QSqlQuery 类以 SQL 语句来实现，要么用 QSqlCursor 类，它封装了 SQL 命令集。除了 QSqlDatabase、QSqlCursor 和 QSqlQuery 这些类外，QSqlError、QSqlField、QSqlIndex 和 QSqlRecord 也支持该层；(3)驱动程序层。本层由三个类组成：QSqlResult、QSqlDriver 和 QSqlDriverFactoryInterface。

在分析了 Qt 的相关代码后我们得出要编写一个数据库的 plugin 要完成以下的几个关键步骤：

(1)必须编写一个子类继承数据库插件基类 QSqlDriverPlugin，?在 plugin 的.cpp 文件中，需要使用 Q_EXPORT_PLUGIN2()这个宏来确保 plugin 能够被 Qt 识别。

(2)至少需要继承 SQL 架构中驱动程序层的两个类 QsqlDriver、QSqlResult，通过对这两个类的继承以及其中一些虚函数的具体实现，用户可添加特定数据库的驱动程序，执行 QT 对特定数据库的操作。

(3)生成动态链接库并且将.dll 和.lib 文件保存在 QTDIR\ plugins\sqldrivers 目录下。这样在使用 Qt 库编写应用程序时可以动态的加载。

因此 Ingres 空间数据库驱动的实现也有以下三个步骤：

(1)继承 QSqlDriverPlugin 类，具体代码如下：

```
class QIngresDriverPlugin : public QSqlDriverPlugin
{
public:
        QIngresDriverPlugin(); //构造函数
QSqlDriver* create(const QString &); //生成新的数据库驱动
        QStringList keys() const; //驱动名列表
};
QIngresDriverPlugin::QIngresDriverPlugin()
        : QSqlDriverPlugin()
{
        //构造函数为空
}
QSqlDriver*QIngresDriverPlugin::create(const QString &name)
{
    if (name == QLatin1String("QIngres"))
    {
        //生成一个新的数据库驱动
        QIngresDriver* driver = new QIngresDriver();
            return driver;
        }
            return 0;
}
QStringList QIngresDriverPlugin::keys() const
{
    QStringList l;    //Qt 列表类的实例 l
    //将 QIngres 这个驱动加入到列表中
    l << QLatin1String("QIngres");
        return l;
}
```

//注册一个数据库驱动，驱动名字 qsqIngres，类名为 QIngresDriverPlugin

Q_EXPORT_STATIC_PLUGIN(QIngresDriverPlugin)

Q_EXPORT_PLUGIN2(qsqlIngres,QIngresDriverPlugin)

(2)继承 QsqlDriver、 QSqlResult 类，分别为 QIngresDriver、QIngresResult 类，派生类实现的重要函数如下：

QIngresDriver 类重要成员函数：

bool open(const QString & db,const QString & user,const QString & password,const QString & host,

int port,const QString & connOpts)

功能：完成对 Ingres 数据库的连接。

QSqlRecord record(const QString& tablename) const

功能：将表中各字段信息保存在一条记录中，包括字段名、字段长度、字段类型等等。

QSqlIndex primaryIndex(const QString &table) const

功能：返回表的主键信息。

void close()

功能：关闭数据库，释放所有的资源。

QIngresResult 类重要成员函数：

bool prepare(const QString & query)

功能：如果要一次插入多条记录的话就使用该函数，这使得操作更加的有效率。

bool exec()

功能：执行记录的多条插入，在函数 prepare 函数后调用。

bool reset (const QString& query)

功能：重置查询，恢复语句句柄。

Ingres 数据库驱动中实现的难点如下：

①需要在 record 函数中获取数据表的字段信息，获取 Ingres 字段使用了 Ingres OpenAPI 函数，函数具体实现代码如下：

```
static  bool IIIngres_getdescr( IngresHANDLE*stmtHandle, IIAPI_DESCRIPTOR **desCr ,int *colCount)
    {
        // 保存数据表结构的结构体
        IIAPI_GETDESCRPARM   getDescrParm;
//阻塞作用的结构体
        IIAPI_WAITPARM   waitParm = { -1};
//不采用异步方式
        getDescrParm.gd_genParm.gp_callback = NULL;
        getDescrParm.gd_genParm.gp_closure  = NULL;
    // Iiapi_query 函数产生的语句句柄
        getDescrParm.gd_stmtHandle=*stmtHandle;
//初值为 0,在调用 IIapi_getDescriptor 函数后返回表的列数
        getDescrParm.gd_descriptorCount=0;
// 缓冲区地址初值为空，调用 IIapi_getDescriptor 函数后返回保存列信息的缓冲区地址。
        getDescrParm.gd_descriptor = NULL;
        IIapi_getDescriptor( &getDescrParm );
//等待函数完成
while(getDescrParm.gd_genParm.gp_completed == FALSE)
            IIapi_wait( &waitParm );
//判断 OpenAPI 执行函数后的状态
        if(getDescrParm.gd_genParm.gp_status !=0)
            return false;
        *colCount=getDescrParm.gd_descriptorCount;
//获取列数
        *desCr = getDescrParm.gd_descriptor;
//获取缓冲区地址
        return true;
    }
```

②将 Ingres 空间数据转换成 Qt 中的数据格式

```
static  QVariant::Typeq  DecodeIngresTy
```

```
pe(short IngresType)
    {      //type 变量初始为无类型
        QVariant::Type type = QVariant::Invalid;
        switch (IngresType)     //判断 Ingres 数据库的数据格式
                {            ……
//Ingres 数据库中的数据格式，可用来保存空间数据。
                case IIAPI_LBYTE_TYPE:
//自定义一个 Qt 数据类型来保存数据库中的空间数据。
                    type = QVariant::UserType;
                break;
                  ……
    }
            return type;
    }
```

(3)生成动态链接库并且将.dll 和.lib 文件保存在%QTDIR%\ plugins\sqldrivers 目录下。

4.2 Ingres 驱动的测试

测试中我们改写了 Tora 中的相关源代码，使其在运行中加载 Qt 库中的 Ingres 驱动插件。实现对 Ingres 数据库的访问和操作，测试中，在 Ingres 数据库中存在一张保存了空间数据的表，空间数据字段的字段名为 the_geom。通过 Tora 的读取， 试验结果如图 2 所示。

5 结论

本文研究了 Qt 中的数据库驱动，并且实现了 Qt 对开源 Ingres 空间数据库的支持。测试结果表明，Ingres 数据库驱动能够较好在 Qt 库中实现对 Ingres 数据库的读取。

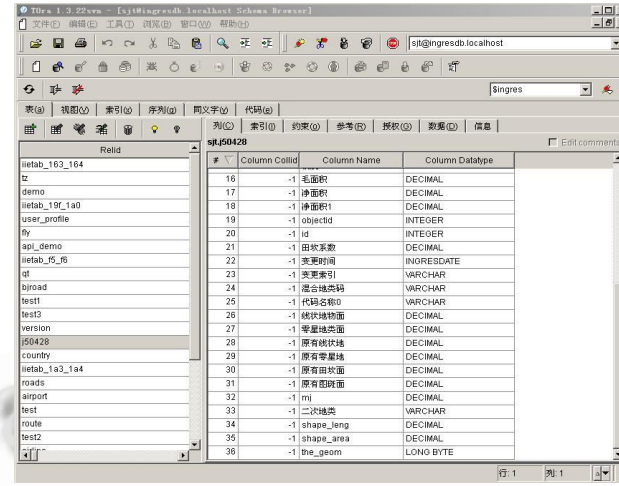

图 2　Tora 操作 Ingres 空间数据库

同时也发现了一些不尽人意的地方，如对大数据量表的读取还存在着一些问题，在 Tora 中操作数据库的功能还没有完全实现，这些都需要进一步改进。

参考文献

1 陈述彭,鲁学军,周成虎.地理信息系统导论.北京科学出版社,2000.

2 Shekhar S, Chawla S, 谢昆青等译.空间数据库.北京:机械工业出版社, 2004.

3 张成才,孙喜梅,黄慧.SDE 的实体——关系模型空间数据管理方式研究.计算机工程与应用,2003,39(2):199－201.

4 http://www.Ingres.com/.

5 http://www.qtcn.org/bbs/.

6 http://tora.sourceforge.net/.