

# 交通规划数据管理中关系型数据库的拓展应用

梁衡弘

(东南大学 交通学院 江苏交通规划与管理重点实验室, 江苏 南京 210008)

**摘要:**建设城市交通规划数据管理系统中,在以 C++ Builder 为开发工具建立复杂友好的图形操作界面的同时,为解决多媒体介质存储利用和复杂计算问题,利用 Oracle DBMS 以 C/S 模式实现对包括大型媒体在内的数据的有效管理,利用 Matlab 在后台完成复杂的数值计算。

**关键词:**C++ Builder; Oracle; 大型二进制数据; Matlab

**中图分类号:**TP311.132.3

**文献标识码:**A

**文章编号:**1005-3751(2006)04-0164-03

## A Developing Design for Urban Traffic Management Planning RDBMS

LIANG Heng-hong

(Jiangsu Provincial Key Lab. of Transportation Planning and Management, Sch. of Communications, Southeast Univ., Nanjing 210008, China)

**Abstract:** Presents the design for urban traffic management planning DBMS. Besides using C++ Builder to realize complex friendly graphical interface, use Oracle database to realize the effective management to traffic data in C/S mode, and use Matlab to finish complex background calculating.

**Key words:** C++ Builder; Oracle; BLOB; Matlab

### 0 引言

城市交通规划收集的数据是自然媒体,表现形式为图、文、声、像,对采集的调查表数据还要求进行归纳、统计、计算、分析、适当的逻辑推理、预测、总结和评价。

城市交通规划的数据多样化决定了城市交通规划数据管理系统是一种“多介质”的数据管理系统,这里的“多介质”指数据不仅包括常规的关系型数据,还包括计算数据、文件、静态的图形和动态的图像等,即包括大型二进制对象类型。

对大型媒体数据,关系型数据库管理一般采用的方法是在数据库中设计一个引用表,媒体以文件的方式存储于磁盘上,由操作系统来进行管理。当需要访问这些数据时,根据引用表中相应的代码找到文件名及所在路径,再到磁盘上去访问相应的数据文件。带来的问题是:

- (1)实际的数据流放于数据库之外;
- (2)数据文件因操作系统因素易被移动或删除;
- (3)数据库管理系统(DBMS)无法保证数据的完整性,无法实现数据的统一管理和安全控制。

所以,对城市交通管理规划数据管理系统应考虑采用大型企业级数据库,充分利用这类数据库所提供的数据类

型,将所应用的数据都存储于数据库中,利用 DBMS 使数据具有规范性、完整性及安全性。

在编制城市交通管理规划方案过程中会涉及大量的数值计算和分析,因此要考虑数据库应用程序与采用的数值分析软件的接口问题。在交通规划中常用的数值分析软件为 Matlab, Matlab 是由美国 MathWORK 公司推出的用于数值计算和图形处理的数学计算环境。它集中了强大的各种数学处理功能,集数值分析、矩阵运算、信号处理和图形图像处理于一体。设计好数据库与 Matlab 的接口,就可以充分利用其提供的函数数值分析、预测、数值模拟等工作,大大节省编写低层算法的时间,避免程序设计中的重复劳动,达到事半功倍的目的<sup>[1]</sup>。

### 1 数据管理系统结构

城市交通管理规划方案是多人共同合作完成的结果,利用计算机网络进行分布式管理成为必然的选择,考虑到网络环境下数据库引擎的效率与人机交互效率问题,采用 Client/Server 模式,结合 C++ Builder + Oracle + Matlab 设计城市交通管理规划数据管理系统。系统结构见图 1<sup>[2]</sup>。

C++ Builder 是由 BORLAND 研发的基于客户/服务器应用和 Internet/Intranet 应用的数据库管理系统开发工具,它通过无缝地集成数据库部件和可视化用户接口的创建,极大地降低了开发数据库应用程序的复杂度。其使用

ODBC或自带的高速驱动程序与数据库连接,支持 Oracle,SQL Server 和 Access 等几乎所有的数据库产品。与 VB,VC++ 和 PB 等其他开发工具相比,C++ Builder 最突出的优点之一是可以可以在程序设计的任何时间浏览活动(Live)数据,真正实现应用开发的“所见即所得”<sup>[3]</sup>。我们采用是其 6.0 版本,数据库引擎 BDE 使用 SQLORA32.DLL 驱动程序直接连接 Oracle9i 服务器。

Matlab API 接口程序利用 DDE 通信机制,由应用程序服务界面直接调用 Matlab 命令,完成计算与绘图功能。考虑到所编应用程序需要在 Windows 工作环境下运行所有程序,因此,选择 Matlab 计算引擎函数库作为 Matlab 与 VC++ 间进行数据交换的接口。另外,考虑到要与交通规划与模拟软件,如“交运之星”等衔接,选择文本文件格式作为计算结果的输出文件。

城市交通规划数据管理系统的数据来源是社会调查而来的自然媒体,调查表格可以通过数据库输入模块完成数字化过程,对不规范数据文件则需经数字化预处理后方能利用,数字化预处理主要工作内容为规范数据格式,比如对照片,要求它为 JPG(1024 \* 768)格式。它的功能虽然不在系统的设计范围之内,但它的结果会直接影响系统的功能与效率,因此不容忽视。

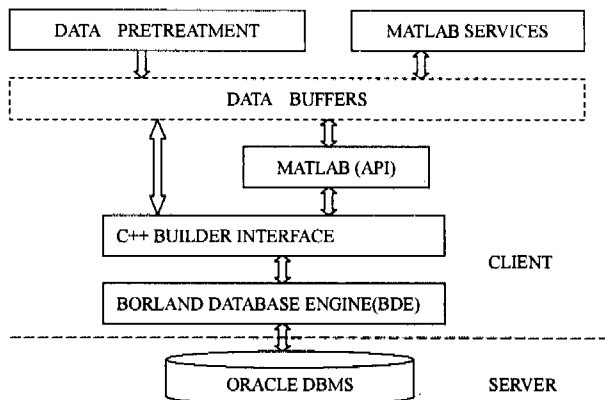


图1 城市交通管理规划数据管理系统结构框图

## 2 大型二进制文件的数据库读取

Oracle 数据库的 CLOB 和 BLOB 数据类型是 Oracle 数据库提供的巨型对象(LOB)数据类型,用来存储大型数据。其中 CLOB 类型用于存储字符对象,如文本文档。BLOB 类型用于存储二进制对象,如图像、视频剪辑或声音文件。这两种数据类型最大都能保存 4G 的数据容量<sup>[4]</sup>。程序设计的基本思路是:插入或更新 CLOB 或 BLOB 类型字段时,首先将数据写入到一个 TmemoryStream 对象中(可以从文件中获取数据或从其它对象中倒入数据),利用 TmemoryStream 类提供的 ReadBuffer 方法从该对象中读取数据,然后利用 TstoredProc 控件调用 Oracle 包 LOB- ACT 中的函数插入记录。查询 CLOB 和 BLOB 类型字段的数据时,先利用 TmemoryStream 控件调用 Oracle 包 LOB- ACT 的函数读取数据,再利用 Tmemo-

ryStream 类提供的 WriteBuffer 方法将数据写入 TmemoryStream 对象中,最后利用 TmemoryStream 类的其它方法和 C++ Builder 的其它控件或类对数据进行处理。以下是针对 BMP 或 JPG 格式的图形文件操作的源代码:

```
void TEditForm::ChangePicture(TField * field, TDBImage
* dbimage)
{
TOpenDialog * PictureOpenDialog = new TOpenDialog(this);
PictureOpenDialog->Filter = "Jpeg 图像文件 (*.jpg) | *.jpg |
BMP 图像文件 (*.bmp) | *.bmp"; //定义 TmemoryStream 对象
if(PictureOpenDialog->Execute()){
field->DataSet->Edit();
TMemoryStream * pms = new TMemoryStream;
TBlobField * fi = (TBlobField *)field;
fi->BlobType = ftOraBlob;
dbimage->Picture->LoadFromFile(PictureOpenDialog->FileName);
dbimage->CopyToClipboard();
dbimage->PasteFromClipboard();
dbimage->Picture->Graphic->SaveToStream(pms);
fi->LoadFromStream(pms);
//从 TmemoryStream 对象载入图像
delete pms;
}
delete PictureOpenDialog;
//TOpenDialog * PictureOpenDialog = new TOpenDialog(this);
/* if(PictureOpenDialog->Execute())
{
TClientBlobStream * Stream1;
TClientBlobStream * Stream2;
field->DataSet->Edit();
Stream1 = (TClientBlobStream *) field->DataSet->CreateBlob-
Stream(field, bmReadWrite);
Stream1->LoadFromFile(PictureOpenDialog->FileName);
Stream1->Position = 0;
//dbimage->Picture->LoadFromFile(PictureOpenDialog->File-
Name);
try
{Stream2 = (TClientBlobStream *) field->DataSet->CreateBlob-
Stream(field, bmReadWrite);
try{
Stream2->CopyFrom(Stream1, Stream1->Size);
((TBlobField *) (field))->LoadFromStream(Stream2);
//dbimage->LoadPicture(); //输出图像到文件
}
finally
{
delete Stream2;
}
finally
{
}
```

```
delete Stream1;
|
|
delete PictureOpenDialog;
```

### 3 应用程序界面对 Matlab 的调用

Matlab 中提供了 DDE 接口,用户可以通过 Windows 的 DDE 通信机制实现外部调用。在 Windows 系统中, DLL 是一种很特别的可执行文件,可以被多个 Windows 应用程序同时访问,具有固定的共享数据段。该数据段的数据在 DLL 被 Windows 下载前会一直保留在内存中,因此可以通过 DLL 实现用户程序与 Matlab 之间的数据传输和函数调用。具体地说,就是利用 Matlab 的 32 位动态连接库(DLL),生成相应的可以被 C++ Builder 调用的 DLL,用来提供二者之间的基本支撑环境。只需在用户程序中加载该 DLL,即可实现其数据段的共享。

然后在用户程序中操作 DLL 数据段的数据,并通过某种方式在用户程序中使 Matlab 执行该 DLL,就可实现用户程序对 Matlab 的调用。其形式可以是混合编程或函数调用,非常方便而高效<sup>[5]</sup>。具体实现方法如下:

Matlab 提供了可外部连接的 DLL 文件,通过将其转换为相应的 Lib 文件,并加以必要的设置,就可以在 C++ Builder 中直接进行 Matlab 函数调用,实现 C++ Builder 语言与 Matlab 语言的混合编程。

#### (1) C++ Builder 下 Lib 文件的生成。

Matlab 提供的 Def 文件允许用户通过 Implib 命令生成相应的 Lib 文件。

在 <matlab>\extern\include 目录下,提供了如下 3 个 Def 文件:

```
__libeng.def, __libmat.def, __libmx.def
```

通过“Implib ??? .lib ??? .def”命令可以生成相应的 3 个 Lib 文件。这些 Lib 文件中包含了可外部调用的 Matlab 函数的必要信息。

#### (2) 实现计算和绘图。

为清楚起见,通过一个简单的例程进行说明。该实例通过调用 Matlab 实现矩阵运算并绘制图形,来演示 CBuilder 对 Matlab 的调用。

在 CBuilder 编辑环境中,建立一个新的窗体 MyForm,并放置一个按钮 Demo。将工程文件命名为 Try.prj,其主函数为 try.cpp。在主函数中,将使用一个实现 Matlab 调用的子函数 DemoMatlab,作为按钮 Demo 的响应事件。源代码较简单,就此略过。

为了调用 Matlab 中的函数,必须进行必要的设置,将包含这些函数的文件加入工程文件 Try.prj。操作过程:

a. 在头文件中加入 Engine.h。其包含了启动 Matlab 调用和关闭的函数声明。

b. 打开 Project | Option... 对话框,点击 Directories/Conditionals。

c. 在 Include Path 中,加入目录路径 < matlab > \extern\include,该路径包含了 engine.h 和 matlab.h 等有用的头文件。

d. 在 Library Path 中,加入 < matlab > \bin 和 < matlab > \extern\include。这两个目录路径包含了可外部调用的 DLL 和 LIB 文件。

e. 点选 Project | Add to Project... 对话框,加入库文件:

```
__libeng.lib, __libmat.lib 和 __libmx.lib
```

在进行了这些必要的设置之后,就可以选用适当的函数来实现目标。以下是子函数 DemoMatlab 的程序代码。

```
void DemoMatlab
{
    Engine *eng; //定义 Matlab 引擎
    char buffer[200]; //定义数据缓冲区
    int array[6] = {1,2,3,4,5,6};
    mxArray *S=NULL, *T=NULL;
    engOpen(NULL); //打开 MATLAB 引擎... (1)
    S = mxCreateDoubleMatrix(1,6,mxREAL); //产生矩阵变量
    mxSetName(S, "S");
    memcpy((char *) mxGetPr(S), (char *) array, 6 * sizeof(int));
    engPutArray(eng, S);
    //将变量 X 置入 Matlab 的工作空间
    engEvalString(eng, "T=S/S.2;"); //计算
    engEvalString(eng, "plot(S,T);"); //绘制图形
    .....
    engOutputBuffer(eng, buffer, 200); //获取 Matlab 输出
    T = engGetArray(eng, "T"); //获得计算结果... (2)
    engClose(eng); //关闭 Matlab 引擎,结束调用
    mxDestroyArray(S); //释放变量
    mxDestroyArray(T);
}
```

若还需执行其他功能和任务,那么按上面介绍方法,进行变量声明后,在(1)、(2)处加写需要的语句即可。

当然,使用这种方法调用 Matlab 不能脱离 Matlab 环境的支撑。但当不需要看到 Matlab 的命令窗口时,可将其赋予 Swhide 属性而加以隐藏。

值得一提的是 MathWORKS 公司在 Matlab 6.5 中推出了将 Matlab 中的 .m 函数编译为 COM 组件的工具 COM Builder,使用这个工具可以方便地制作出想要的 COM 组件,在不依赖 Matlab 的环境下可以供其它支持 COM 的编程语言调用。这里使用的是 Matlab 5,没有此组件,所以没有采用此方法。

### 4 结束语

通过对普通关系数据库的拓展,在城市交通规划数据管理系统中解决了多媒体数据管理与一部分数值数据要求加工处理的问题,在与交通规划软件(如“交运之星”)衔

(下转第 169 页)

旧图的操作。所有图形的橡皮筋技术的工作原理和过程如图2所示。

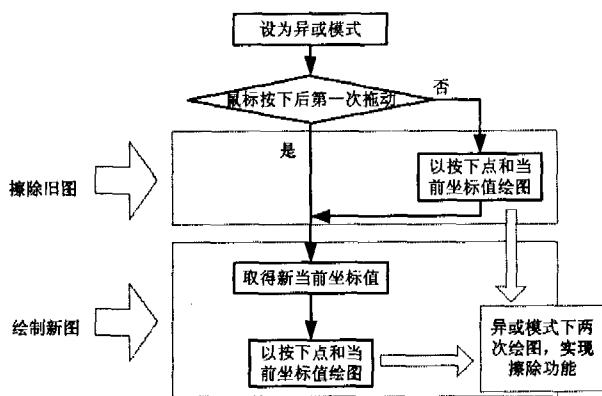


图2 橡皮筋技术的工作过程

## 2.5 图形的保持

Java 的绘图程序代码不负责显示在白板区域的图形的保持问题,所以,当白板窗口状态改变时,白板上的图形信息会部分或全部丢失,这显然不符合用户的要求,用户希望可以随意改变白板窗口的状态而保持原有的图形信息。为了解决图形丢失问题,首先需要为图形信息准备一个存储空间,然后将所有有用的图形信息及时放入存储空间,最后在必要的时候,从存储空间调出数据并显示在屏幕上。

存储空间里的数据只有符合前面定义的通信协议集,才能在重新绘制时被识别并再现。在只存储规则图形如圆、直线和矩形时,存储空间可以设置得比较小,但是徒手画和橡皮擦包含的信息量非常大,对存储空间要求比较高。通过实践,文中定义存储空间为符合通信协议集的1000个数组。

数据的存储和调出用循环队列来实现。循环队列有许多基本操作,根据功能需要,文中只涉及入队、判断队列是否满、设置队列空这三项操作。有两种图形信息需要加入到循环队列中,一种是用户在白板上绘制的图形,另一种是客户终端接受由白板服务器传来的图形。客户端接受传来的图形信息,打包成一个通信协议集类实例 data,再将 data 作为入队元素插入到循环队列中。

在用户未点击清除白板屏幕命令前,所有需要保存的数据以循环队列容量为限,最大可能地保留。如果用户点击了清除白板屏幕命令,此时需要设置队列为空。设置队

列为空的过程比较简单,只要依次将队列中的所有数据清除即可,其过程是用空数据包给队列中的所有元素赋值,达到清除数据的目的。

存储数据的目的是为了在必要的时候,对循环队列中的数据进行出队操作,再现这些信息的内容。当白板窗口状态发生改变,比如窗口缩小后放大,最小化后最大化等,此时需要调用循环队列中的图形数据并重绘白板绘图区。文中利用 Java 中 Swing 的基础类 JComponent 的 paint 方法进行重绘。paint 方法用于绘制组件,用户通过调用 repaint 方法来激发 paint 方法。由于 Java 中默认的 paint 方法不符合网络电子白板中重绘白板绘图区的要求,为了实现特殊的功能,文中重载了该方法。在窗口状态改变时,由系统自动调用 paint 方法。由于重载了 paint 方法,系统会依次读取循环队列中的所有图形数据,并将其显示在白板绘图区,从而完成重绘功能,达到了窗口状态改变而图形仍然保持的效果。

## 3 结 论

网络电子白板是传统意义的黑板在网络上的延伸,是网络教育中重要的工具。文中对电子白板的设计进行了较为系统的分析,实现了一个基于 Java 的网络电子白板系统。首先提出了适用于网络电子白板的体系结构,接着在体系结构的基础上,利用 Socket 编程,实现了电子白板网络间的通信,并制定了相应的通信协议集,最后解决了电子白板在具体使用中的一些相应功能,如橡皮筋功能以及在白板窗口移动时的图形恢复问题。

## 参考文献:

- [1] 周 蔚.现代远程教育建构性学习环境探析[J].教育理论与实践,2005(1):56-58.
- [2] 何光明,何丕廉,孟昭鹏,等.虚拟教室中电子白板的关键技术与实现[J].计算机工程与应用,2003(10):121-123.
- [3] Fujii S, Iwata J, Yoshida K, et al. Development of a Remote Communication System for Computer Novices and Their Instructors[J]. Lecture Notes in Computer Science, 2003, 1774: 764-770.
- [4] 杜宗霞,余雪丽,李世平.基于 WWW 的多媒体电子白板的设计及实现[J].计算机工程,2000(10):581-586.
- [5] 顾俊杰,曹 宁.C/S 模式下电子白板系统的设计与实现[J].淮海大学常州分校学报,2005(3):57-60.

(上接第 166 页)

接还是用文件交换的方式,其人工智能与数据挖掘技术尚不够完善,这需在进一步的工作中加以解决。

## 参考文献:

- [1] 刘志俭,张志勇.MATLAB 应用程序接口用户指南[M].北京:科学出版社,2000.
- [2] KOTH K G, KEVIN L. Oracle8 完全参考手册[M]. 梅钢

译.北京:机械工业出版社,1998.

- [3] 朱时银,马承志,杨 飞,等.C++ + BUILDER5 编程实例与技巧[M].北京:机械工业出版社,2001.
- [4] 王 瑜,王 昕.C++ + BUILDER 程序对 ORACLE 数据库巨型对象类型的处理[J].计算机时代,2003(3):30-32.
- [5] 何毅斌,陈定方,沈元浩,等.Windows 工作环境下 Oracle、VC++ 和 Matlab 的联合应用[J].计算机应用,2001(11):99-100.