

文章编号:1671-9352(2007)11-0073-04

# 数据库存取技术在 CAD 系统中的应用

许文杰<sup>1</sup>, 刘希玉<sup>2</sup>

(1. 山东师范大学 信息科学与工程学院, 山东 济南 250014;

2. 山东师范大学 管理与经济学院, 山东 济南 250014)

**摘要:**针对计算机辅助设计中需要大量数据存入数据库并从数据库中读出的问题,提出了在支持复用的外观设计系统中数据库的设计方法和连接方法,并且以 VC++ .NET 2003 为开发平台,结合 SQL Server 2000 数据库,成功地实现了大量不同类型数据地存取,特别是对 BLOB 的存取方法进行了详细地论述。

**关键词:**二进制大对象; ActiveX 数据对象; 数据库

**中图分类号:** TP311.13      **文献标志码:** A

## Application of database accessing technology in the CAD system

XU Wen-jie<sup>1</sup>, LIU Xi-yu<sup>2</sup>

(1. School of Information Science and Engineering, Shandong Normal University, Jinan 250014, Shandong, China;

2. School of Management and Economy, Shandong Normal University, Jinan 250014, Shandong, China)

**Abstract:** Aiming at the question of accessing a great deal of data from database in CAD, the method of design and connection of the database was introduced in a configuration design system, which supports reuse. The accessing problem in a large number of different kinds of data was finished with the combination of SQL Server 2000 database by VC++ .NET 2003. The accessing method of BLOB(Binary Large Object) was particularly expatiated.

**Key words:** binary large object; ActiveX data object; database

## 0 引言

近年来,数据库开发人员经常需要对图像、音频以及其他二进制的数据进行存取和使用。由于这类数据占用的存储空间较大,一般称之为二进制大对象,即 BLOB(binary large object)<sup>[1]</sup>,其存储方式也与普通数据有所区别。通常会有 2 种处理的办法:一种是将这些数据存放于磁盘上,数据库中则存放相应的路径;另一种则直接存放于数据库中。第一种方式对于开发人员来说简单而易于实现,降低了程序操作的数据库复杂性,但随之出现了一系列的问题:(1) 为了便于管理,用户必须将数据文件存放在

一个固定的目录下,将路径存入数据库之后,文件就不能移动和更改;(2) 无法保证数据的安全性,易于损坏、丢失;(3) 对于大的开发项目,异地分布的网络环境下,设计人员并行协作地参与同一产品的设计与生产装配,带来了很大的困难。所以,通常情况下使用后者,将大对象文档以二进制的形式保存在 BLOB 类型的字段中,目前绝大多数数据库都是支持 BLOB 类型的字段。

## 1 环境介绍

VC++ 具有友好的人机交互界面和面向对象编程的强大功能,支持面向对象编程、模块化、代码可

收稿日期:2007-05-07

基金项目:国家自然科学基金资助项目(60374054);“泰山学者”建设工程专项经费资助项目;山东省自然科学基金重大资助项目(Z2004G02);山东省中青年科学家奖励基金资助项目(NO.03BS003);山东省教育厅科学技术资助项目(J05G01)

作者简介:许文杰(1983-),男,硕士研究生,研究方向:进化计算,数据挖掘.Email:jiesparking@163.com

重用、组件共享等技术,其在数据库方面的应用广泛。VC++ 开发的应用程序与数据库的连接常用的实现方法<sup>[2]</sup>有: ODBC (open database connectivity), DAO(data access object), OLE DB(object link and embedding database), ADO(ActiveX data object)。其中, ADO 是微软提供的面向对象的接口,与 OLE DB 类似,但接口更简单,具有更广泛的特征数组和更高层次的灵活性。ADO 基于 COM,提供编程语言可利用的对象。ADO 对象结构类似于 OLE DB,但并不依靠对象层次。大多数情况下,用户只需要创建并只使用需要处理的对象。

ACIS 是美国 Spatial Technology 公司推出的三维几何造型引擎<sup>[3]</sup>,它集线框、曲面和实体造型于一体,并允许这 3 种表示共存于统一的数据结构中,为各种 3D 造型应用的开发提供了几何造型平台。许多著名的大型系统都是以 ACIS 作为造型内核,如 AutoCAD, CADKEY, TurboCAD 等。它通过提供一个灵活的 3D 建模为建立模块提高了可扩展性,利用 Tech Soft American's HOOPS 3D 应用框架<sup>[4]</sup>可以很容易的开发出图形应用,这个开发框架可以从 Spatial 获得并且集成 ACIS,是应用较为广泛的计算机辅助设计开发<sup>[5]</sup>的 3D 环境。

本文结合参与开发的一个 VC++ .NET2003 及 3D 支持环境为 ACIS 16/HOOPS13 的手机外观 CAD 系统来介绍如何利用 ADO 连接 SQL Server 数据库,存取 BLOB 的具体实现过程。

## 2 实现步骤

### 2.1 构件的分类及构件库的建立

支持集成的构件库是本手机外观设计系统的重要组成部分,是实现构件集成的基础。构件库要求实现如下功能:多种格式构件的存储和管理;海量构件的存储;快速响应客户端的请求;提供方便友好的后台管理界面;提供构件的相关统计管理功能等。

创新设计提供了各种不同的构件,为了对这些构件进行有效地管理并达到更高的可重用性,需要对其进行科学的分类。根据本项目所设计的各种构件对象的特点,将全部构件分为三大类:常规构件库、创新构件库、SAT 构件库,如图 1 所示。

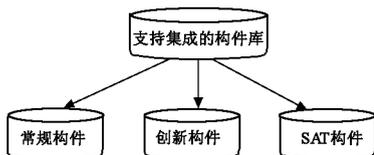


图 1 构件库的分类

Fig. 1 Classification of component library

其中,常规构件是指在手机设计中按常见分类所涉及各个构件,比如直板和翻盖手机外壳、键盘等,要存储的是各个构件的参数以及为了让用户有一个直观认识的图片信息,其界面如图 2 所示;创新构件是指在本系统下通过利用自行进化算法设计的基本造型,设计出的新颖造型,采用了 XML 数据类型作为它的参数存储对象;SAT 构件是指以 SAT 文件提供的构件,它们可来自于任何的可以将三维设计对象保存为 SAT 文件的软件系统,它既可以作为成品使用,又可以在新的手机中作为一部分构件复用。

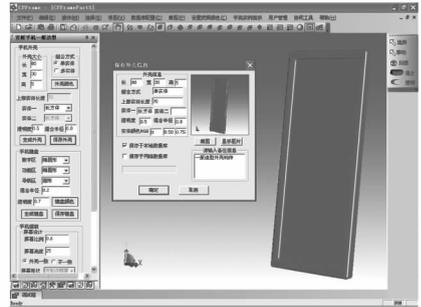


图 2 系统界面及外壳保存对话框

Fig. 2 Interface of the system and conservation dialog of the shell

### 2.2 开发 ADO 应用程序

开发 ADO 应用程序的基本过程<sup>[6,9]</sup>:首先,引入 ADO 库定义文件,初始化 COM 库;其次,用 Connection 对象连接数据库;然后,利用建立好的连接,通过 Connection, Command 对象执行 SQL 命令,或利用 Recordset 对象取得结果记录集进行查询、处理;最后,使用完毕后关闭连接释放对象。

具体实现方法如下:

(1) 使用 ADO 连接数据库:

①导入 ADO 接口。

首先,用 # import 编译指令提示符来引用支持 ADO 的组件类型库(\*.tlb),需要一个类型库文件名(Msado15.dll),并生成对应的头文件。在应用程序中,如果 # import 了 msado15.dll, Visual C++ 将产生 msado15.tlh 和 msado15.tli 2 个文件。可以直接在 Stdafx.h 文件中加入下面的语句来实现 ADO 的数据库访问机制:

```
# import "c: \ programfiles \ commonfiles \ system \ ado \ msado15.dll"
```

no\_namespaceerename("EOF", "adoEOF")

其中,路径名可以根据系统安装的 ADO 支持文件的路径来自行设定。因为 ADO 对象未使用命名空间,因此重命名文件结束符(EOF)为 adoEOF,以避免和其他库中命名相冲突。

②组件初始化,建立连接。

在应用程序中,首先初始化 OLE/COM 库环境,然后再建立数据库连接。使用 Connection 对象的 open 方法建立连接,在 open 方法的参数中设置连接字符串。

```
CoInitialize(NULL); //初始化 OLE/COM 库环境
try {
    m_pConnection.CreateInstance("ADODB.
Connection"); //创建 Connection 对象
    _bstr_t strConnect = "Provider = MS-
DASQL.1; database = jcz; Data Source = jcz; uid =
sa";
    //设置连接字符串
    m_pConnection->Open(strConnect, "",
    "", adModeUnknown);
}
catch(_com_error e) {
    AfxMessageBox(e.Description());
}
```

如不出现异常,只要在使用完 ptrConn, 引用 Close 方法即可。

③创建 Recordset 对象实例,ADO 记录集。过程如下:

```
try {
    m_pRecordset.CreateInstance(_uuidof
(Recordset));
    m_pRecordset->Open(bstrSQL, m_pCon-
nection.GetInterfacePtr(), adOpenDynamic, adLock-
Optimistic, adCmdText);
}
catch(_com_error e) {
    AfxMessageBox(e.Description());
}
```

## (2) 常规构建库的存取

在设计本系统时,在数据库中预存储了大量基本类型的构件和特殊造型的构件,为了方便用户查看,除了将大量最基本的参数信息保存到数据库之外,本文将生成三维构件的截图也要存储到数据库之中。

### ①图片的存储

与普通二进制数据的存储方式有一定区别,需要使用 Field 对象的 AppendChunk 方法进行写操作, AppendChunk 方法的原型如下:

```
HRESULT AppendChunk(const Variant_t &Data);
```

从 AppendChunk 的原型可以看出写操作的关键问题是需要把二进制数据赋值给 VARIANT 类型的变量,以下给出系统实现的具体代码:

```
ado.m_pRecordset->MoveFirst(); //数据库表
中记录的位置
while(ado.m_pRecordset->adoEOF == VARIAN-
T_FALSE)
{
    char * pBuf = m_pBmpBuff;
    VARIANT varBLOB;
    SAFEARRAY * psa;
    SAFEARRAYBOUND rgsabound[1];
    if(pBuf) {
        rgsabound[0].lLbound = 0;
        rgsabound[0].cElements = m_BmpLen;
        psa = SafeArrayCreate(VT_UI1, 1, rgsab-
ound); //创建 safeArray 对象
        for(long i = 0; i < ((long)m_BmpLen2; i++)
        SafeArrayPutElement(psa, &i, pBuf++);
        //将 pBuf 指向的二进制数据保存到
SAFEARRAY 对象 psa 中
        varBLOB.vt = VT_ARRAY | VT_UI1;
        //将 varBLOB 的类型设置为 BYTE 类型的数
组
        varBLOB.parray = psa; //为 varBLOB
变量赋值
        ado.m_pRecordset->GetFields()->Ge-
tItem("Image")->AppendChunk(varBLOB);
        //加入 BLOB 类型的数据
    }
    ado.m_pRecordset->Update(); //保存 BLOB
的数据到库中
}
```

```
ado.m_pRecordset->MoveNext();
```

### ②图片的读取与显示

通常使用 Field 对象的 GetChunk 方法读取 BLOB 类型的数据,GetChunk 的原型如下:

```
variant_t GetChunk(longLength);
```

可以看出,给出数据长度之后,GetChunk 将返回包含数据的 VARIANT 类型变量。在系统中,从数据库中读取 BMP 图象的详细过程如下:

```
long lDataSize = ado.m_pRecordset->GetFields()
->GetItem("Image")->ActualSize;
if(lDataSize > 0)
```

```

{
    _variant_t varBLOB;
    varBLOB = ado.m_pRecordset->GetFields()->
GetItem("Image")->GetChunk(IDataSize);
    if(varBLOB.vt == (VT_ARRAY|VT_UI1))
    {
        if(m_pBmpBuff = new char[IDataSize + 1])
//重新分配必要的存储空间
        {
            char * pBuf = NULL;
            SafeArrayAccessData ( varBLOB. parray,
( void * * )&pBuf);
            m_pBmpBuff memcpy ( m_pBmpBuff,
pBuf, IDataSize); //复制数据到缓冲区
            SafeArrayUnaccessData ( varBLOB. parray);
            m_BmpLen = IDataSize;
            m_hBitmap = BufferToHBITMAP(); //生
成 BITMAP 对象
        }
    }
}

```

### (3) 创新型构件的存储

对于特殊造型的手机构件,它的特殊性主要是体现在构件形状的不规则性。由于特殊造型的形状不具有普遍性,通常以其形状的特殊轮廓线(由奇异函数或随机点生成)扫掠而产生,所以对其参数的设定与对一般造型构件的参数有所不同,它的存储参数主要是构件的生成函数或随机点位置坐标。例如,心型外壳的存储参数为函数:

$$\begin{cases} X = 2 * (1 - \cos(t)) * \cos(t), \\ Y = 2 * (1 - \cos(t)) * \sin(t). \end{cases} \quad (1)$$

通过遗传算法生成的构件,在生成过程中以 2 个参数表示,一个是多节点坐标数组,一个是数组中实际存在的节点个数,并且不同构件的节点数目不同,如果统一使用最大数组表示,那么在存储过程中,对数据库空间将是一种极大的浪费,所以,将数组中的坐标转换为 XML 格式文件进行存储和管理。这既有利于构件格式的统一,又有利于数据库的优化和数据的传输。遗传算法生成构件的 XML Schema 格式设置如下:

```

<element name = "cross"/>
    <complexType content = "elementOnly">
    <complexType name = "Point_num">
    </complexType>

```

```

</element>
    <element name = "Point_num"/>
        <attribute name = "point_x" type =
"DOUBLE"/>
        <attribute name = "point_y" type =
"DOUBLE"/>
        <attribute name = "point_z" type =
"DOUBLE"/>
    </element>

```

通过遗传算法生成的构件,XML 文件中,每个 Point\_Num 元素表示一个节点,每个元素拥有 3 种属性,分别是 point\_x, point\_y, point\_z。表示当前节点的三维坐标,分别是 point\_x, point\_y, point\_z。

### (4) SAT 构件的存储

SAT 构件库中的构件主要有 2 个来源,一个是在 hoop/acis 环境下生成的构件经过组装而集成的能够作为成品的复杂构件,另一个是来自于其他设计环境下的产品作为本系统的构件。对于第一个类的复杂构件,如果继续依据参数或者 XML 数据类型对其进行存储,存在如下问题:参数结构过于复杂,从而使数据库的结构过于庞大,冗余数据过多,造成传输上的瓶颈;在配置一般的系统回显构件过程中,读取参数后调用生成时使用的函数对构件进行重绘,需占用系统的大量内存,容易给用户感觉系统反应迟钝。基于以上各方面的考虑并结合在本环境下生成的构件可以存储为二进制 SAT 文件,并大大减小参数存储和回显构件过程中时间和空间的复杂度等原因,对于这部分的构件,采用存储 SAT 文件的方式,实现方法与存储图片方法类似,存储结构见表 1。

表 1 SAT 构件表的设计  
Table 1 Design of the SAT component table

序号	字段名	数据类型	说明
1	Sfile	binary	SAT 文件
2	Name	nchar	SAT 文件名称
3	Description	Nchar	详细描述

通过以上 3 种构件的分类存储,实现了系统构件库的集成。

## 3 结束语

本文介绍了几种 VC++ 和数据库的常用连接方式,论述了在 VC++ 环境 CAD 项目中利用 ADO 数据库访问技术,对不同 BLOB 数据(下转第 81 页)

(上接第 76 页) 的存储和读取方法,给出了程序的实现代码,完成了项目所要实现的功能。但是,由于 ADO 自身的限制,对数据进行更新修改时不能断开与数据源的连接,将会占用大量数据库的服务器资源;以后可以尝试通过用 ADO.NET 连接数据库来实现相应的功能。

#### 参考文献:

- [1] 朱春花,张子彪,路新春. BLOB 数据类型存取方法的研究及应用[J]. 计算机应用与软件,2002,19(10):52-54.
- [2] 费巧玲,徐向阳. VC 中用 ADO 实现大数据的存取[J]. 计算机工程与应用,2005,24:182-184.
- [3] Spatial Technology. HOOPS 3D graphics system overview[EB/OL]. (2005-07-01)[2007-04-25] <http://www.spatial.com.cn>.
- [4] Spatial Technology. HOOPS 3D application framework[EB/OL]. (2005-07-01)[2007-04-25] <http://www.spatial.com.cn>.
- [5] 刘弘,刘希玉,唐明晰. 支持外观造型创新设计的计算机辅助设计环境[J]. 计算机辅助设计与图形学学报,2003,15(10):1258-1263.
- [6] 刘生平. Visual C++ .NET 数据库开发技术与实践[M]. 北京:清华大学出版社,2005.
- [7] 李光举,刘四方,隽峰. VC++ 环境下数据库巨量数据存取技术研究[J]. 微计算机信息,2006,22(10-3):150-152.
- [8] 王瑜涛,王宁生. VC++ 实现数据库大字段存取[J]. 计算机应用与软件,2001,18(8):51-53.
- [9] 刘显荣. 用 C++ 实现数据库中图片的存取与显示[J]. 计算机应用研究,2006,23(5):163-165.

(编辑:孙培芹)