基于 DWGDirectX 读写 DWG 图纸文件的实现①

A Practical Method of Reading and Writing DWG File Based on DWGDirectX

肖起年　(同济大学 电信学院 上海 200092)

摘　要： 介绍了一种基于 DWGDirectX 在不依赖于 AutoCAD 平台的情况下实现 DWG 文件的显示、操作、添加的简单的实体的方法，并对该方法进行了分析和实现。

关键词： DWG DWGDirectX 非 AutoCAD 平台

1　引言

AutoCAD 软件作为当今世界最为流行的绘图软件之一，在 CAD 市场有着举足轻重的地位，尤其是 Autodesk 公司所制定的图形文件格式 DWG 被全球数十亿的工程图纸所广泛采用。

在查看 DWG 文件和对其操作方面，使用的最为广泛的就是直接使用 AutoCAD 软件打开。除此之外，也有大量其他软件可以通过调用 AutoCAD 软件的一些库来实现这些功能，但终究无法脱离 AutoCAD 系统，这就使用户还是不得不安装 AutoCAD 这个庞大的平台[1]。但是，AutoCAD 高昂的使用费用却让很多人望而却步。

本文基于 Open Design Alliance 开发一个叫 DWGDirectX 的 ActiveX 控件提出了一种方法，它可以在脱离 AutoCAD 的环境下解读 DWG 文件，并可以使基于它开发的应用程序拥有读写 DWG 文件的功能[2]。

2　DWG文件及DWGDirectX

2.1 DWG 图纸文件组成

DWG 是一种工业上用来存储 2 维或 3 维设计数据的格式。包括 AutoCAD，IntelliCAD 和 PowerCAD 在内的数百种应用软件都支持该格式并大多以它作为最原始的数据存储格式。自从 1982 年起，它一直由 Autodesk 公司设计、定义和诠释的。

一般一个 DWG 图纸文件由 5 部分组成，分别是：文件头(HEADER)、实体部(ENTITY)、表部(TABLE)、块实体部(BLOCK)和应急头部。

DWG 文件头存放 DWG 文件的一些重要信息，如 DWG 文件标志、版本信息、各种索引用的地址，以及 AutoCAD 软件中的一部分系统变量。文件头后面是图形数据区，图形数据区包含实体(entity)数据区和块实体数据区(block)，以及表(table)数据区。

实体数据区存放各种实体图形。在 AutoCAD 中，图形实体指基本图形元素，如：点(Point)、线(UBe，Ray，Xline，Mline，Pline，Spline 等)、圆(Circle)、圆弧(Arc)、块(Block)、尺寸标注(Dimen．sion)等：在 DWG 文件中实体由实体头和实体尾部组成，实体头描述各种实体的一些共同属性，所以不同实体的实体头结构是一样的，实体尾部描述了实体的几何参数，因此不同实体的实体尾部结构不一样。

表数据区存放块表(block table)，图层表（layer table)，线型表(1inetype table)等。

块实体数据区是由每个块所对应的实体表组成。一个块包含多个图形实体，每个块包含的实体都以一个表的形式存放。

应急头部，主要存放一些重要索引信息的副本[3]。

2.2 Open Design Alliance 和 DWGDirectX

Open Design Alliance，是一个致力于开发开源的具有工业标准的 CAD 文件交换格式的软件开发者组织。自 1982 年来，Autodesk 公司发布过至少 18 种版本的 DWG 文件格式，但却从来没有公开过其任何读写技术。Autodesk 公司也出售一个叫做 RealDWG 的库，人们可以通过花费一笔高昂的费用来

① 收稿时间:2008-09-07

获得有限的权限来读写 DWG 文件。因此有相当多的公司和团体都尝试去揭开 DWG 格式的技术面纱，其中最成功的就是 Open Design Alliance[4]。

Open Design Alliance 这个非营利组织自 1998 年组建来一直致力于 DWG 文件的解读工作，他们推出了一套开发库使开发人员可以较为容易的读写流行的各个版本的 DWG 文件，并且与近年推出了 DWGDirectX 这个 ActiveX 控件，使得程序员可以更加容易的开发读写 DWG 文件的程序[5]。

3 基于DWGDirectX读写DWG文件原理

DWGDirectX 的解读工作就是把 DWG 格式图纸的所包含的各类信息以一系列对象的形式存在于内存中。这些对象分门别类、各司其职，这样我们就能易于理解和操作这些信息，而不像 DWG 格式这样扑朔迷离。其中有些对象尤其重要[6]：

OdaHostApp OdaHostApp 对象代表着整个应用程序。也就是说，OdaHostApp 对象提供的一系列接口让我们能以面向对象方式来处理应用程序。它有一个 Application 属性，我们可以通过这个属性来操作主应用程序。同时它作为主函数，也可以作为一个监控函数起着控制数据库的作用。

AcadApplication AcadApplication 类的对象只能由 OdaHostApp 的 Application 属性获得。它负责控制整个应用程序的各个属性和应用程序层面的方法。所有应用程序层面的一些工作都是由它完成的，而且各个其他对象也是通过它来与操作系统等外界的环境进行交互。

AcadDocument AcadApplication 对象的 Documents 属性返回一个 AcadDocument 对象集合，如下图所示。其中每个 AcadDocument 对象都代表一张图纸。这意味着这个对象集合可以包含多张图纸。每张图纸都在该集合中有一个对应的索引。一个 AcadDocument 对象包含了这张图纸的属性和图纸中的所有对象所普遍具有的共同方法。可以使用 OdaHostApp.Application.ActiveDocument 来获得应用程序当前处理的图纸对象。

AcadModelSpace,AcadPaperSpaceAcadDocument 对象的 ModelSpace 属性和 PaperSpace 属性可以各得到一个 AcadModelSpace 集合和 AcadPaperSpace 集合。一张图纸中的绝大部分的几何对象都是可以在在两个集合中访问到[7]。

大部分用户画图时创建的模型是存储在图纸文件的一个叫模型空间(ModelSpace)的地方的。AcadMo

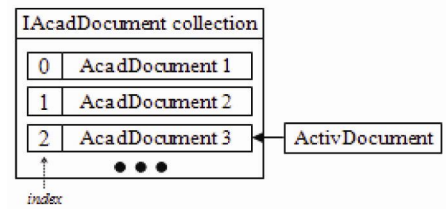

图 1　IAcadDocument 组成示例

-delSpace 集合就代表了这里面所有的对象。每张图纸都有一个模型空间，所以对应在内存中就会有一个 AcadModelSpace 集合。

AcadPaperSpace 集合则代表了模型空间中各个对象的一个视图，还有文件中图纸空间的一些额外的对象。这些额外的对象通常只是用于打印的。每张图纸可以有多个 AcadPaperSpace 集合。

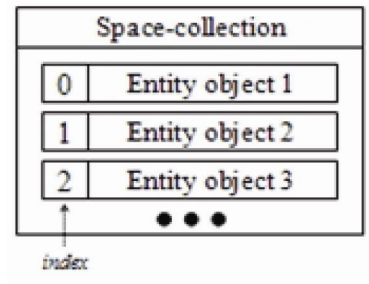

图 2　ModeSpace 组成示例

DWG 图纸中那些最基本的元素比如点、线、圆、弧、以及文字都是以对应的点实体、线实体、圆实体等存储于模型空间或图纸空间中的。每个存储于空间集合中的实体都有个对应的索引，我们可以在模型空间中以该索引找到这个实体并访问它。

因此某种意义上说，我们读取 DWG 图纸文件的各个图元就是读取模型空间和图纸空间中的各个实体，而我们在 DWG 图纸文件中写入图元也就是在模型空间和图纸空间中添加实体。因为模型空间和图纸空间中可以存储各种不同的实体，而这些实体又有着不同的属性和不同的参数，所以当我们向它们中添加各类实体时就要调用各种实体在 AcadModelSpace 和 AcadPaperSpace 中各自对应的方法。同样的道理，当我们要删除一个实体的时候，也必须在各自实体对应的对象中找到对应的删除方法才可以在模型空间和图纸空间中删除掉它们。

除了模型空间和图纸空间这两个最重要的属性外，图层、线型、材质、多线型、文字样式以及用户坐标系等其他信息也都存在 AcadDocument 对象的

各个对应的属性中。

一张 DWG 图纸文件中所包含的各种信息就这样分门别类以各自对应的对象的形式存储于内存中并相互联系起来的以供读写操作的。

4 基于DWGDirectX读写DWG文件操作

DWGDirectX 不仅能够解读 DWG 图纸文件，也提供了基本的对图纸的呈现功能。有了这样一个强大的工具后，开发一个简单的能读写 DWG 图纸文件的程序就变得十分简单了。

4.1 读取文件操作

对 DWG 文件操作的第一步就是要打开这个文件了，正如上文所说的，DWGDirectX 在内存中用 AcadDocument 对象代表一张图纸文件，所以我们要做的就是用这张图纸生成一个 AcadDocument 对象。

```
oDoc = odaApp.Documents.Open(FileName, false, "Password");
```

其中，参数 Filename 是图纸文件的完全路径名字；参数 false 代表该文件不是只读的；而最后的参数“Password”是用于打开加密的文件的，此处如果打开的文件没有加密，则该参数设为空字符串即可。这样我们就把图纸按上文所提到的结构解读于内存中了。

接下来就是要呈现这些解读后的信息。DWGDirectX 中封装了一个 OdaDevice 类，它可以用于接受这些解读后的信息并把它们绘制在屏幕上

```
OdaDevice.SetupActiveLayoutViews((VIEWXLib.IAcadDatabase)oDoc.Database)
```

此时只要再调用 OdaDevice 的 update 方法即可让它重新绘制屏幕，也就是把图纸文件画在屏幕上了。

4.2 写入文件操作

与读取操作类似，经过 DWGDirectX 解读后，写入 DWG 文件也变得易于实现了。正如本文第二部分解释的那样，由于 DWG 文件是把各个信息分解为各个类存在内存中的，因此我们需要添加新内容到文件中时，所要做的工作就是把信息对号入座添加到各自的位置上。

4.2.1 输入参数绘图

我们以绘制图纸时做得最多的工作，也就是在图纸中绘制点、线、多边形这些新图元为例，这个操作需要打交道的对象就是第二部分着重提到的 ModelSpace 和 PaperSpace。

事实上，写入 DWG 图纸文件要做的工作就是把画出来的图形以 DWGDirectX 规定的格式存入到 ModelSpace 中去。比如我们需要加入一条线段时，我们就可以调用为 ModelSpace 加入线段的方法 AddLine,如下例所示：

```
oDoc.ModelSpace.AddLine(startpoint,endpoint)
```

实际程序中就是通过填入起始点和结束点坐标来给程序提供对应的参数的。

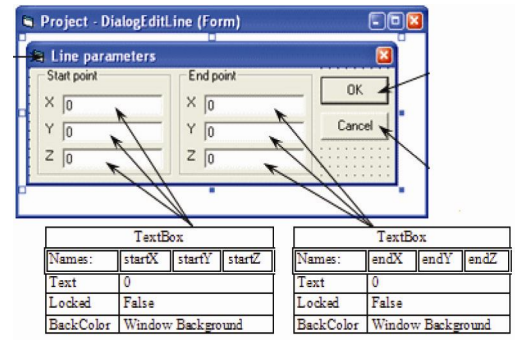

图 3　输入参数绘制线的界面

4.2.2 屏幕直接绘图

不过仅仅依靠输入坐标点来确定一条线要从来开始画到哪结束这样是不够便捷的。一般情况下，更加普遍的做法是利用鼠标直接在屏幕上画图，也就是说要让程序把用户鼠标画在屏幕上的动作转化为对应在图纸中相应的坐标位置上的绘图动作。

实现这个所画即所得的功能，我们所要做的首先是获得用户鼠标在屏幕上绘图时的位置信息，这些信息就是用户期望画制出的东西。其次是要把这些屏幕上的坐标信息与 DWG 图纸中的坐标信息对应起来，让程序知道用户画在屏幕上的点对应在图纸中应该是哪个点；而屏幕上用户画出的轨迹，在图纸中又应该是什么样的比例什么样的轨迹。这样，我们最后才能像上文中输入参数绘图那样的把正确的参数信息传递给 DWGDirectX 以绘制出正确的图形。

由于 DWGDirectX 只是封装了对 DWG 文件的解读和一些基础的显示功能，我们无法让显示控件直接感知鼠标点击来判断鼠标点击在图纸的什么位置。而且这个显示窗口无法接受鼠标单击事件。

本文采用的方法是使用一个全局钩子来捕获鼠标单击事件以得到鼠标的位置。再通过窗口的区域与图纸在窗口中显示部分的区域之间的比例来计算得到鼠标点击在图纸中相应的坐标。

具体方法是首先在设计的程序中加入一个能够捕获 Windows 系统消息事件的钩子[8]：

```
public void setHook()
{
```

```
if (hHook == 0)
{
MouseHookProcedure=new HookProc(MainForm.MouseHookProc4Line);
hHook = SetWindowsHookEx(WH_MOUSE,
MouseHookProcedure,
(IntPtr)0,
AppDomain.GetCurrentThreadId());
……
```

这段代码把一个自己定义的钩子程序嵌入到 Windows 系统处理鼠标事件的钩子队列中去。这样我们就可以在自定义的钩子程序中获得用户在屏幕上绘图时的鼠标的坐标。

然后我们要获得转化屏幕坐标和图纸坐标所需要的信息。当前应用程序窗口的实际像素大小我们可以通过显示窗口的 size 属性获得。而 DWG 图纸在应用程序窗口中显示出的部分的大小及坐标信息我们可以用 oDoc.GetVariable("VIEWCTR")和 oDoc.GetVariable("SCREENSIZE") 来计算得出。"VIEWCTR" 和 "SCREENSIZE"是两个 CAD 系统变量，分别可以获得 UCS 坐标表示的当前视口中视图的中心点和以像素为单位存储当前视口的大小(X 和 Y 值)。

计算出图纸在应用程序窗口显示出的区域的坐标信息后，我们带入屏幕坐标信息以及鼠标在屏幕上的坐标后即可以计算出用户绘制动作应该在图纸中对应的坐标了。然后把这个作为参数传给上文中输入参数绘图的代码，就可以实现所画即所得的这种通过屏幕上直接绘制来写入图纸的功能了。

4.3 程序实例

基于上文的思想，笔者制作出了一个简单的小程序。它可以实现基本而常用的 DWG 文件的读写操作。运行界面如下图所示：

该程序“文件”菜单下实现了打开、关闭、保存 DWG 文件的功能。“绘图”菜单下则实现了一些常用的图元的绘制功能。菜单栏下的工具栏则实现了一些查看图纸所最常用的一些放大、缩小、平移、旋转等功能。

5 技术分析和展望

本文主要讲解了 DWGDirectX 是如何把 DWG 文件解析为我们可以便于访问的格式，以及如何在脱离 AutoCAD 这个平台的情况下基于这个控件来制作一个简单的读写 DWG 文件的程序。

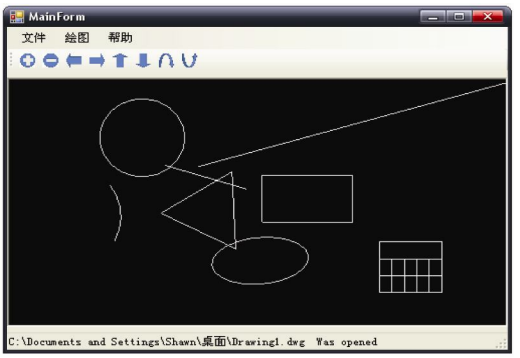

图 4　应用程序界面图

DWGDirectX 这个控件已经基本上实现了对目前流行的所有版本的 DWG 格式文件的解读功能。但如何利用解读后的内容来读写 DWG 文件则需要我们自己来实现。AutoCAD 这个软件之所以能畅销全世界，除了他拥有着对 DWG 文件良好的解读能力外，他所提供给用户那些全面而便捷的绘图功能更是他的优势所在。因此，开源工作者们想要自己制作出不依赖与 AutoCAD 平台的好软件，在实现对绘图所需要的各种功能上还有很多很多的工作要做。

笔者由于时间仓促，只实现了一些最基本最常用的功能。还有更多的其他的功能和特性，还需要有兴趣的读者进一步探索和研究。

参考文献

1 李桥梁,楼佩嫂,马万太.DWG 文件的内部结构及其访问.自动化科学技术应用学术会议.北京:电子工业出版, 2001:76－81.
2 李华.聂建国. 在非 CAD 平台环境下显示 CAD 图形的实用技术与方法.计算机工程与应用,2002, 38(9):92－94.
3 Goldberg H E. CAD data standards for AEC.Advanstar Communications, 2006:46－47.
4 孙士华.仲梁维.付良健.于涛. 基于 DWGdirect 技术的图形内容全文搜索.精密制造与自动化.
5 黄维丰.DWG 文件离线浏览 ActiveX 控件研究.中国[硕士学位论文]南京:南京航空航天大学, 2003.
6 侯颖.何援军.胡志刚.非 Aut0CAD 平台的 DWG 文件工程信息提取技术.计算机工程, 2003,29(11):180－188.
7 Open design Alliance.http://www.opendesign.com/ .
8 HOOK 专题 http://www.microsoft.com/china/community/program/originalarticles/techdoc/hook.mspx.