基于 DWG DirectX 的电子产品图样结构关系轻量化构建方法研究

杨平 黄巍

（中国电子科技集团第三十研究所，四川，成都，610041）

摘要：为便于样机研发阶段显示产品图样结构关系，准确统计产品零部件信息，提高对图样装配栏与明细栏信息的审核效率，利用 VC 2008 平台与 DWGdirectX 开发工具实现了在非 AutoCAD 环境下的图样信息提取。利用嵌套循环比对算法深度遍历图样信息集合，并构建图样之间的装配映射关系。利用 MFC 控件直观显示图样结构关系与 BOM 表信息，自动统计与核对产品零部件数量，实现缺漏图样提醒。在检入 PDM 系统前，有效降低了图样 BOM 信息的错误发生率，并最终生成符合企业标准的产品清单。

关键词：图样结构树 DWGdirectX 动态链接库 信息提取 递归算法

0 引言

电子产品的图档文件依据构成要素可分为结构零部件图表、印制板板卡图表和机内外通信互联线缆图表等，其图样关系构成较为复杂，如大型网络交换机与新型互联网控制器等。其设备表现出功能更加密集化、融合度更高的特点[1]，同时对处于样机研制阶段中图样模型的完整性与正确性提出了更高的要求。要满足产品完整性与正确性，就需要构建清晰的产品结构树关系。目前商用产品数据管理系统（PDM）采用自顶向下方式构建产品结构树关系，而在产品样机研制阶段往往缺乏设备顶层装配图表，因此商用 PDM 系统很难全面、清晰地反映出样机阶段产品的结构树关系。

电子通信产品的 BOM 信息是产品的核心组织，其中的产品结构树与零部件数量信息是检验产品完整性的重要信息。这些信息来源于产品的每一张设计图样。如何高效的读取产品 BOM 信息，检验产品零部件图样的正确性与齐套性，不仅是设计者同时也是产品信息管理者所关心的问题。目前 Autodesk 公司的 DWG 格式文件是国内外应用最为广泛的产品设计信息存储文件[2]。批量读取 DWG 格式图样信息方法较为多样，例如，仇巍基于 ObjectARX 利用单元化信息匹配的方式提取图样信息[3]，董玉德等利用明细表坐标比对方式识别图样明细栏信息并提取[4]，此两种方法都无法完全摆脱 AutoCAD 环境，因此存在耗用计算机资源较多，提取速度慢的问题。黄维丰和李桥梁等利用 ODT 技术实现了对 DWG 格式图样直接读取[5]，但只能提取 2004 版本之前的图样文件。本文基于 Open Design Alliance 开发的 DWGDirectX 的 ActiveX 控件[6]，脱离 AutoCAD 环境，直接读取 DWG 文件，提取所需图样 BOM 信息，采用自底向上的方式构建图样结构树与零部件统计信息，实现对图样 BOM 信息的集中统计、检查、核对与缺漏提醒。

1 图样信息提取方法

1.1 DWG 图样格式分析

DWG 文件主要由表段（TABLE）、实体段（ENTITY）、块段（BLOCK）和文件头（HEADER）等组合而成，文件

收稿日期：2016 年 5 月 6 日

杨平（1968—），男，高级工程师，主要研究方向为军用电子装备结构设计。
黄巍（1982—），男，工程师，主要研究方向为军用电子装备结构设计。

以二进制方式书写，用于存储 AutoCAD 图形文件。

其中 DWG 文件的版本号与各段中的起始地址、图形因子以及 DWG 整参数等记录于 DWG 的头文件中。因 AutoCAD 版本的差异，带来 DWG 文件会有些许细微差别。

图块由多个不同类型的实体组成，同时每个图块对应属于自己的索引用于区别不同的图块，这些索引组成了包含块名、块长度、标志字和结束标志的块表。

实体段用于记录 AutoCAD 中称之为实体的图元数据。这些图元是指直线、点、圆、弧、文字、多义线、轨迹线和插入块等。块段用于记录组成 DWG 文件每个块的颜色、线型和插入基点等实体信息。由每个块所对应的实体表组成。表段在 DWG 文件中存储在实体段之后，实际上它是由块表、层表、字型表、线型表、视图表顺序组成的。层表由描述每个图层的表记录依次组成，表记录包括以下内容：层状态标志、层名、层颜色、层线型。

文件中一些重要索引信息的副本被存放于应急头部之中。

1.2 基于 DWGDirectX 读取原理

DWGDirectX 通过 API 函数提供对 DWG 或 DXF 文件的内容的读写通道。DWGDirectX 提供的接口中含有许多类型的对象，这些对象是以一种层次化的方式来组织的。DWGDirectX 描述的 DWG 文件对象模型结构图如图 1 所示。

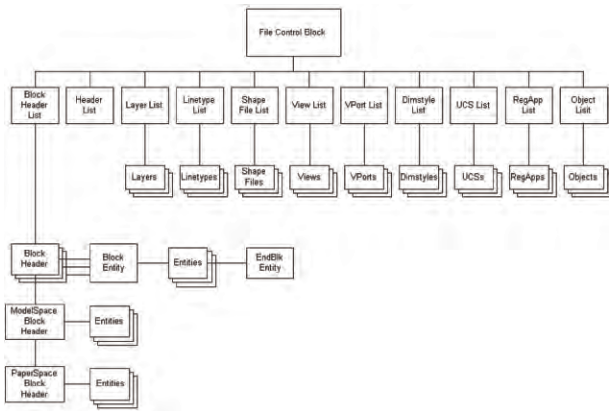

图 1 DWG 文件结构图

读取图样的核心机制在于将 Autodesk 公司加密的二进制图样文件转化为数据自定义的模型文件，DWGDirectX 组织数据的方式与 ObjectARX 非常相似，包括线型(LineType)、形文件(shapeFile)、图块(Bloek)、图层(Layer)、视口(ViewPort)、标注样式(Dimstyle)、视图(View)、注册应用程序(RegApp)和用户坐标系(ues)等 9 个容器表，以及 Modelspace(模型空间) 和 PaperSPace(图样空间) 两种基本类型的块表。

OdaHostApp 类表示整个应用程序，OdaHostApp 的对象中定义了相关面向对象的接口，用于处理应用程序。利用其 Application 属性来操作主应用程序。AcadApplication 类的对象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 AcadApplication 来与操作系统等外界的环境进行交互。但 AcadApplication 类的对象只能通过 OdaHostApp 对象的 Application 属性获得。

AcadDocument 对象的 Documents 属性返回一个 AcadDocument 对象集合，其中每个 AcadDocument 对象都代表一张图样。每张图样都有自己对应的模型空间，所以对应在内存中就会有一个 AcadModelSpace 集合。在 DWG 格式的图样文件中的点、线、文字、圆和圆弧等元素，都是以对应的实体形式存储在模型空间或图样空间中。每个实体都有一个索引，在模型空间中通过该索引我们可以找到该实体。

因工程图样格式相对固定，因此可利用图样中的“块”特性，预先定制不同图幅中标题栏与明细栏的属性名称与属性值。在图样提取过程中，重点对图样中的“块”属性值进行比对与提取，从而实现对图样标题栏与明细栏信息提取。

1.3 提取过程与实现代码

在进行软件开发之前需要先引入动态链接库的接口文件，即要复制 DWGDirectX 软件包中的 DWGDireetX.tlb、OdaToolkit.h 和 OdaX.h 等文件到所编程序的目录中，再将其头文件引入至程序中，代码如下：

```
#inelude”OdaToolkit.h” ;
#imPort’’DWGdirectX.tlb’’;
```

这样就完成了引入工作，同时还要定义一些操作 DWG 文件实例的变量：

```
IOdaHostAPP*m_iHost;
IAcadAPPlication*m_iAPPlieation;
IAeadDoeument *m_iCurrentDoc;
```

因以上所引用变量需要遵从 COM 规范，所以在引用时需要增加计数，同时在完成引用之后需要释放变量，利用 Release() 函数完成释放。

另外在程序开头需要引入接口变量，利用 COM 中的智能指针实现对变量的引入：

投稿邮箱 | cadcam@idnovo.com.cn
IMCHINA@idnovo.com.cn

```
CComPtr<IUnknown>PUnk;
CComQIPtr<IEnumVARIANT&IID_IEnumVARIANT>PNeWEnum;
CComQIPtr<IAcadEntity, &IID_IAeadEntity>PEnt;
CComQIPtr<IAeadText, &IID_IAcadText>PText;
CComQIPtr<IAeadLine, &IID_IAcadLine>PLine;
```

在图样信息提取之前，先对图样进行模板预定义，将标题栏和明细栏定义为属性块，利用 DWGdirectX 动态链接库，找到图样中相应的块名，依据属性标签中的名称进行比对，便可有针对性的提取明细栏、装配栏和标题栏中的数据，如图号、名称和数量等。相关实现代码如下：

```
void LoadDwg(CString FileName,int flag)
{
CoInitialize(NULL);
HRESULT  hr;
IClassFactory  *pCF;  // 处理整个类的对象
CoGetClassObject(__uuidof(OdaHostApp),CLSCTX_INPROC_SERVER,NULL,IID_IClassFactory,(void **)&pCF);// 找到接口
    hr=pCF->CreateInstance(NULL,IID_IOdaHostApp,(void **)&m_iHost);
// 创建新的非初始化接口对象
if (CLASS_E_NOTLICENSED == hr)
    {
    IClassFactory2  *pFactory;
    hr=pCF->QueryInterface(IID_IClassFactory2,(void **)&pFactory);
    if (S_OK == hr)
    {
            hr = pFactory->CreateInstanceLic(NULL,NULL,IID_IOdaHostApp,(BSTR)_bstr_t("some key"), (void **)&m_iHost);
            pFactory->Release();
    }
    }
    pCF->Release();
    if (S_OK != hr)
            return;
    hr=m_iHost->get_Application(&m_iApplication);  // 获得程序应用
    if (S_OK != hr)
    {
    m_iHost->Release();  // 程序初始化
    m_iHost = NULL;
            return ;
    }
    BindSink();
    IAcadDocuments* pDocuments;  // 定义操作图样指针
    IAcadDatabase*  pDb;
    _variant_t      vPassword;
  hr=m_iApplication->get_Documents(&pDocuments);  // 获得图样集合
    if (S_OK == hr)
    {
    vPassword.vt = VT_ERROR;
    hr=pDocuments->Open(FileName.AllocSysString(), _variant_t(true), vPassword, &m_pCurrentDoc);// 打开单个图样并，获取
    if (S_OK == hr)
    {
    hr = m_pCurrentDoc->get_Database(&pDb);
  // 获取图样数据信息
    if (S_OK == hr)
    {
            if(flag==0)          // 比对块名称
            ReadTitle(pDb);      // 读标题栏
            if(flag==1)          // 比对块名称
            ReadMxb(pDb);        // 读明细栏
    pDb->Release();
    }
    }
    pDocuments->Release();
    }
    m_iApplication->Quit();
```

```
    m_iApplication->Release();
    m_iApplication = NULL;    // 释放程序
    UnBindSink();
    m_iHost->Release();
    m_iHost = NULL;           // 释放句柄
    CoUninitialize();
}
```

2 图样结构关系关键算法

2.1 图样数据结构模型

将提取到的图样明细信息存入到一个名为 ProduceData 的嵌套结构体数组中，该数组中每一个元素代表一张图样的信息，该图样的名称和图号信息记入标题栏结构，它的各个子部件的名称、图号和装入数量等信息记入嵌套的明细栏结构数组中。该结构数组在文档类中定义，它完整记录了组成产品的所有零部件的信息，作为构建图样结构树与零部件信息统计的基础数据，其数据结构定义如下。

```
struct ProduceData
{
    CString filename;     // 文件名
    CString head-flag;    // 头标示参数
    CString tail;         // 尾部索引参数
    struct Title          // 标题栏结构体
    {
        CString name;     // 名称
        CString code;     // 图号
    }Title_info;
    struct List           // 明细栏结构体
    {
        CString name;     // 名称
        CString code;     // 图号
        CString number;   // 装入数量
    }List_info[];
}DWG_info[];
```

2.2 图样之间关联特点与关系构建

反映设备齐套性的设计文件主要由零件、部件、整件、关重件、外购件和辅助表格文件等组成。文件之间利用引用于装配形成映射关系，具有网状模型的特征。依据有无引用关系，整机图样又可分为零件图与装配图。其中零件图中只存在标题栏，用于记录零件图号、数量和名称等信息，零件图中无装配关系信息，所以不能向下构建子节点，是产品结构树的终端节点。装配图中即包含标题栏也包含装配栏，其中装配栏中记录了子节点的图号、名称及数量信息，可依次向下构建结构树子节点。

建立装配图样与零件图样之间的装配映射关系是构建结构树的关键。由于图样信息栏中图号的唯一性，因此可利用图号构建图样文件之间映射关系。由于无法预先判定顶层总装图图号，即不能实现自顶向下构建图样结构关系。因此仅将图样集合区分为“装配图”与“零件图”两级，利用结构体数组 ProduceData 中的 code 记录图号，head-flag 作为装配关系判断，tail 作为子集图样的索引，采用自底向上的方式构建图样之间的网状关系。方法如下：

（1）将所读的图样信息写入结构体数组 DWG_info[] 中，且读入的每一张图样对应结构体数组 DWG_info[i] 中唯一的序号 i，如图样为装配图，则明细栏中的图号也对应此数组 DWG_info[i].List_info[k] 中唯一的序号 K。

（2）开始嵌套循环对比图样集合 DWG_info[] 中的 code 图号信息，如果 DWG_info[i].code 值与 DWG_info[j].List_info[k].code 值相同，则利用尾部变量 tail 记录被装入图样的索引号，即将 DWG_info[j].List_info[k].tail 值记录为 i，并将被装入图样的头标示变量值记录为“done”表示此图样与其他图样存在装配关系。

（3）嵌套循环完成后图样之间的映射关系便记录在结构体数组中的 tail 变量中。在构建结构树时先判断 ProduceData 中的 head-flag 值是否为空，如果为空则说明本张图样不会装入其他图样中，可作为结构树的根节点，利用 tail 索引值可直接找到子节点的图样序号并读取其中信息，而对于 tail 值为 null 的装配图样，则说明图样集合中缺少对应此装配图号的零件图样，可进行图样缺漏提醒。

2.3 建立图样信息结构树

由于所检入图样可能存在多个总装图，即存在多个根节点，因此首先在基础数据中找到无装配关系的 ProduceData 结构体数组。由上一节中对无装配关系的 ProduceData 的 head-flag 参数标记为 null，因此只需循环比较便可找到根节点结构体 ProduceData 编号。根据之前所建立的图样装配映射关系，在找到根节点后便可依照深度优先策略，即先

投稿邮箱 | cadcam@idnovo.com.cn
IMCHINA@idnovo.com.cn

子后兄的方式从根节点开始建立结构树视图关系，步骤如下：

（1）插入结构树根节点。

（2）如果当前节点有子节点（含装配关系），用子节点变量替换，循环调用“构建结构树”递归函数，转步骤（1）。

（3）若该节点无子节点（无装配关系），返回。

具体代码如下：

```
ConstructTree(code,item)
{
FatherItem=Insert(code,item);    // 插入节点
If(HaveChild(code))          // 存在子节点情况
{
For(i=0；i<ChildNumber；i++)
{
ConstructTree(childnode[i],FatherItem) ；
// 替换变量递归循环
}
}
Return；
}
```

3 实例应用

以构建图样信息结构树与图样 BOM 信息统计显示为目的，利用以上所介绍算法，实现批量读取 DWG 格式图样文件信息，并同步构建图样间关系。利用树控件构建产品结构，统计图样数量，并利用 list 控件显示图样信息，实现图样结构树缺损信息提醒，最终将统计信息以 excel 格式导出，生成产品零部件统计清单。图样结构树与产品零部件信息统计工具软件如图 2 与图 3 所示。

图 2 图样结构树构建

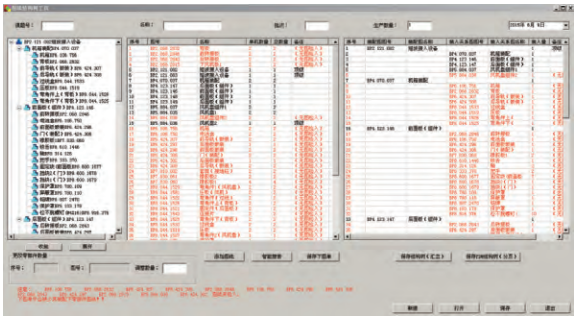

图 3 图样结构树软件应用

4 结论

本文重点介绍了基于 DWGDirectX 库技术的图样信息提取方法，实现了对图样标题栏与明细栏内容的快速读取，并以此为基础数据，提出了构建图样之间装配映射关系的方法。

针对样机研制阶段图样的管理要求，以快速采集图样信息，构建图样结构树与零部件数量统计为目标，利用 DWGDirectX 库与 VC++ 工具编制应用程序，读取图样信息，并采用自底向上的方式构建图样结构树与输出符合企业标准的零部件统计清单，实现了样机阶段图样信息的集中可视化管理与图样 BOM 信息高效纠错核对。

参考文献：

[1] 董爱先，王学军．第 5 代移动技术及发展趋势 [J]. 通信技术，2014，47(03)：235-240.

[2] Leong K K，Yu Kai Ming. Product data allocation or distributed product data management system [J].Computers in Industry，2002，(47)：289-298

[3] 仇巍，刑建国．基于 ObjectARX 提取 AutoCAD 中任意格式表格信息 [J]. 青岛大学学报，2013，02(06)：49-52.

[4] 董玉德，刘孙．面向工程图样离线式表格信息提取与识别方法研究 [J]. 工程图学学报，2009，01：17-25

[5] 黄维丰，李桥梁，吴洪涛．基于 OpenDWG Toolkit 访问 AutoCAD DWG 图形文件 [J]. 现代计算机，2003，01：69-72.

[6] Open Design Specification for .dwg files Version 5.2