

Open Cascade DataExchange DXF

eryar@163.com

摘要 Abstract: 对 DXF 文本格式进行详细介绍, 并介绍了如何使用开源库 dxflib 对 DXF 文件进行读写操作, 并将 DXF 文件中图形导入到 OpenCascade。

关键字 Key Words: DXF、Open Cascade、Data Exchange, dxflib

一、引言 Introduction

目前市面上的 CAD 系统都有自己的数据文件, 各个系统之间的数据结构和格式各不相同, 这样极大影响了设计和制造部门之间或企业之间的数据传输和程序衔接的自动化, 同样给 CMM 和 CAD/CAM 的数据通信带来困难。因此, 迫切需要数据交换文件格式的标准化。

产品数据的正确交换可以使双方不必重建数据, 提高效率。产品数据交换的通常做法是使用数据交换接口, 即需要交换的每一系统与标准数据格式(中性格式)之间开发双向转换接口, 两系统通过中性格式进行交换。

DXF 是 Drawing eXchange File 的缩写, 意思为图形交换文件, 在工程制图中有广泛的应用, 掌握了 DXF 文件的读写对编写 CAD 软件时的图形信息的交换有重要意义。它有两种格式: 一种是 ASCII DXF 格式; 一种是二进制 DXF 格式。ASCII DXF 文件格式是 ASCII 文字格式的 AutoCAD 图形的完整表示, 这种文件格式易于被其它程序处理。二进制格式的 DXF 文件与 ASCII 格式的 DXF 文件包含的信息相同, 但格式上二进制格式比 ASCII 格式更精简, 能够节省百分之二十五的文件空间。AutoCAD 能够更快地对其执行读写操作(通常能快五倍)。这可能是对 ASCII 格式的 DXF 文件操作时有 ASCII 与二进制形式的转换, 因而花费时间较多。ASCII 格式的 DXF 文件可读性强。

OpenCascade 也有数据交换接口 DataExchange, 可对主流图形交换格式进行读写。

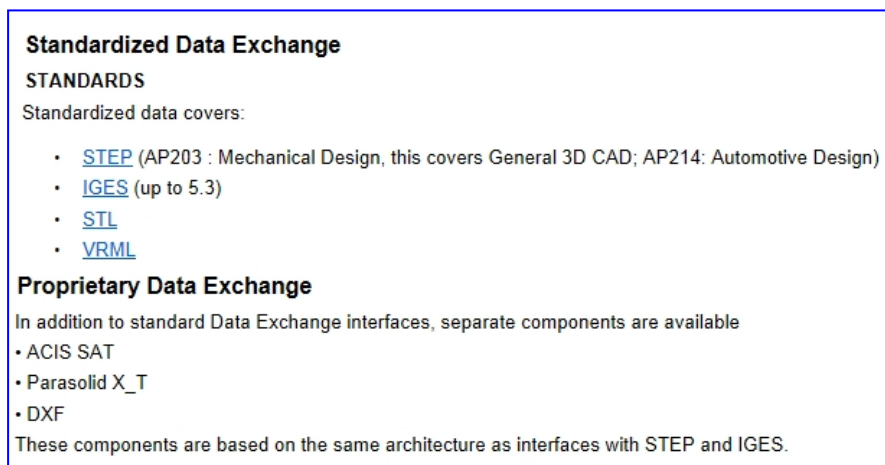


Figure 1.1 DataExchange Module of OpenCascade

其中对 STEP、IGES、STL、VRML 的程序是开源的, 对 ACIS 的 SAT、Parasolid 的 X_T 和 DXF 数据交换的程序是收费的, 对 DXF 读写的是 DXF-Import-Export, 详情可参考: <http://www.opencascade.org/support/products/dataex/>, 也可直接和 OpenCascade 进行联系。

二、DXF 格式简介 DXF Introduction

一个完整的 ASCII 格式的 DXF 文件结构如下：

- **HEADER** 段。它包含图形的基本信息。它由 AutoCAD 数据库版本号和一些系统变量组成。每个参数都包含一个变量名称及其关联的值。
- **CLASSES** 段。包含应用程序定义的类的信息，这些类的实例出现在数据库的 BLOCKS、ENTITIES 和 OBJECTS 段中。类定义在类的层次结构中是固定不变的。
- **TABLES** 段。包含以下符号表的定义：
 - APPID (应用程序标识表)
 - BLOCK_RECORD(块参照表)
 - DIMSTYLE(标注样式表)
 - LAYER(图层表)
 - LTYPE(线型表)
 - STYLE(文字样式表)
 - UCS(用户坐标系表)
 - VIEW(视图表)
 - VPORT(视口配置表)
- **BLOCKS** 段。包含构成图形中每个块参照的块定义和图形图元。
- **ENTITIES** 段。包含图形中的图形对象（图元），其中包括块参照（插入图元）。这里的信息很重要。
- **OBJECTS** 段。包含图形中的非图形对象。除图元、符号表记录以及符号表以外的所有对象都存储在此段。OBJECTS 段中的条目样例是包含多线样式和组的词典。
- **THUMBNAILIMAGE** 段。包含图形的预览图像数据。此段为可选。

每个段都以一个后跟字符串 SECTION 的组码 0 开始，其后是组码 2 和表示该段名称的字符串（例如，HEADER）。每个段都由定义其元素的组码和值组成。每个段都以一个后跟字符串 ENDSEC 的组码 0 结束。

关于 DXF 文件更详细内容可参考我 08 年写的一篇文章《[用 C 读取 DXF 文件](http://blog.csdn.net/eryar/article/details/2371755)》，可以从网上下载，网址：<http://blog.csdn.net/eryar/article/details/2371755>。在 11 年又写了一个 **Read DXF File**》，虽然使用了继承，但是读入时的很多判断的处理感觉还不是很满意，文章代码网址：<http://www.cppblog.com/eryar/archive/2013/03/26/162303.html> 参考这些代码，可以完全理解文本格式的 DXF 文件，并可自己动手来处理 DXF 文件了。

三、使用 dxflib 与 OpenCascade 交换数据 Wrap dxflib for OpenCascade DataExchange

如果程序中只用到 DXF 简单的图形，如直线、圆弧、圆等，可以自己写个处理的程序；如果要处理 DWG 格式的文件，有些库可以使用，如 openDWG Toolkit，虽然带有 Open 字样，但是这个工具并不是免费的，免费的有 LibreDWG；如果只处理文本格式的 DXF 文件，可以使用库 dxflib。当然 dxflib 也是有不足之处的，如只能处理文本格式的 DXF，不能处理二进制格式，不能处理包含的 ACIS 三维实体等。如果程序只处理简单的图形、三维面片等，dxflib 还是很好用的。

dxflib 的使用方法很简单，通过调用 DL_Dxf 对象的 in 函数并将指针传递给一个继承 DL_CreationAdapter 抽象类来打开此文件。in 函数运行时，会调用传递给它的类中的几个函数。如今，有数十个这样的函数（参阅 DXFLib 程序员手册），但是在大多数情况下，受关注最多的有如下这几个函数：addPoint、addLine、addCircle 和 addVertex。您只需实现您所关注的那些函数；剩下的您可以忽略。

下面给出一个使用 dxflib 读取形状并转换到 OpenCascade 中的具体例子。

四、程序示例 Example Code

这里给出具体的程序示例，当然目前只实现部分形状到 OpenCascade 的转换，如果有兴趣可以实现其他形状的实现。程序代码如下所示：

类声明文件：DxfReader.h

```
/*
 *   Copyright (c) 2013 eryar All Rights Reserved.
 *
 *   File : DxfReader.h
 *   Author : eryar@163.com
 *   Date : 2013-12-22 14:40
 *   Version : 1.0v
 *
 *   Description : Use dxflib to import entities from DXF to OpenCascade.
 *
 *   Key Words : OpenCascade, dxflib, DXF, DataExchange
 */

#pragma once

#define WNT
#include <TopoDS_Shape.hxx>
#include <BRep_Builder.hxx>

#include <memory>

#include "dl_dxf.h"
#include "dl_creationadapter.h"

/**
 * @brief Facade dxflib for OpenCascade DataExchange with DXF.
 */
class DxfReader : public DL_CreationAdapter
{
public:
    /**
     * @brief constructor.
     * @param fileName [in] dxf file name with path.
     */
    DxfReader(const std::string& fileName);
    ~DxfReader(void);

    /**
     * @brief Get the shape of the dxf.
     * @return OpenCascade topology shape.
     */
    const TopoDS_Shape& GetShape(void) const;

public:
    virtual void addPoint(const DL_PointData&);
    virtual void addLine(const DL_LineData& line);
    virtual void addArc(const DL_ArcData& arc);
    virtual void addCircle(const DL_CircleData& circle);
    virtual void addEllipse(const DL_EllipseData&);
};
```

```

virtual void addPolyline(const DL_PolylineData& polyline);
virtual void addVertex(const DL_VertexData&);

virtual void addSpline(const DL_SplineData&);
virtual void addKnot(const DL_KnotData&);
virtual void addControlPoint(const DL_ControlPointData&);

virtual void add3dFace(const DL_3dFaceData&);
virtual void addSolid(const DL_SolidData& solid);

private:
    std::auto_ptr<DL_Dxf> mDxf;

    TopoDS_Compound mShape;

    BRep_Builder mBuilder;
};

```

类定义文件: DxfReader.cpp

```

/*
 *   Copyright (c) 2013 eryl All Rights Reserved.
 *
 *   File : DxfReader.cpp
 *   Author : eryl@163.com
 *   Date : 2013-12-22 14:40
 *   Version : 1.0v
 *
 *   Description : Use dxflib to import entities from DXF to OpenCascade.
 *
 *   Key Words : OpenCascade, dxflib, DXF, DataExchange
 *
 */

#include "DxfReader.h"

#include <Degrees.hxx>

#include <gp_Circ.hxx>
#include <gp_Elips.hxx>

#include <GC_MakeSegment.hxx>
#include <GC_MakeCircle.hxx>
#include <GC_MakeArcOfCircle.hxx>
#include <GC_MakeEllipse.hxx>

#include <TopoDS.hxx>
#include <TopoDS_Edge.hxx>

#include <BRepBuilderAPI_MakeVertex.hxx>
#include <BRepBuilderAPI_MakeEdge.hxx>
#include <BRepBuilderAPI_MakeFace.hxx>
#include <BRepBuilderAPI_MakePolygon.hxx>

DxfReader::DxfReader(const std::string& fileName)

```

```

: mDxf(new DL_Dxf())
{
    mBuilder.MakeCompound(mShape);

    mDxf->in(fileName, this);
}

DxfReader::~DxfReader(void)
{
}

const TopoDS_Shape& DxfReader::GetShape() const
{
    return mShape;
}

void DxfReader::addPoint(const DL_PointData& point)
{
    mBuilder.Add(mShape, BRepBuilderAPI_MakeVertex(gp_Pnt(point.x, point.y,
point.z)));
}

void DxfReader::addLine(const DL_LineData& line)
{
    Handle_Geom_Curve theSegment = GC_MakeSegment(gp_Pnt(line.x1, line.y1, line.z1),
gp_Pnt(line.x2, line.y2, line.z2));

    mBuilder.Add(mShape, BRepBuilderAPI_MakeEdge(theSegment));
}

void DxfReader::addArc(const DL_ArcData &arc)
{
    gp_Circ theCircle;
    theCircle.SetRadius(arc.radius);
    theCircle.SetLocation(gp_Pnt(arc.cx, arc.cy, arc.cz));

    Handle_Geom_Curve theArc = GC_MakeArcOfCircle(theCircle, DEG(arc.angle1),
DEG(arc.angle2), false);

    mBuilder.Add(mShape, BRepBuilderAPI_MakeEdge(theArc));
}

void DxfReader::addCircle(const DL_CircleData& circle)
{
    gp_Circ aCircle;
    aCircle.SetRadius(circle.radius);
    aCircle.SetLocation(gp_Pnt(circle.cx, circle.cy, circle.cz));

    Handle_Geom_Curve theCircle = GC_MakeCircle(aCircle);

    mBuilder.Add(mShape, BRepBuilderAPI_MakeEdge(theCircle));
}

void DxfReader::addEllipse(const DL_EllipseData& ellipse)
{
}

```

```

}

void DxfReader::addPolyline(const DL_PolylineData& polyline)
{
}

void DxfReader::addVertex(const DL_VertexData& vertex)
{
}

void DxfReader::addSpline(const DL_SplineData& spline)
{
}

void DxfReader::addKnot(const DL_KnotData& knot)
{
}

void DxfReader::addControlPoint(const DL_ControlPointData& cp)
{
}

void DxfReader::add3dFace(const DL_3dFaceData& face)
{
    BRepBuilderAPI_MakePolygon polygon(
        gp_Pnt(face.x[0], face.y[0], face.z[0]),
        gp_Pnt(face.x[1], face.y[1], face.z[1]),
        gp_Pnt(face.x[2], face.y[2], face.z[2]),
        gp_Pnt(face.x[3], face.y[3], face.z[3]), true);

    BRepBuilderAPI_MakeFace makeFace(polygon.Wire());

    if (makeFace.IsDone())
    {
        mBuilder.Add(mShape, makeFace.Face());
    }

    //mBuilder.Add(mShape, polygon.Wire());
}

void DxfReader::addSolid(const DL_SolidData &solid)
{
    add3dFace(solid);
}

```

类 DxfReader 的使用也很简单，程序如下所示：

```

/*
 * Copyright (c) 2013 eryar All Rights Reserved.
 *
 * File : Main.cpp
 * Author : eryar@163.com
 * Date : 2013-12-22 15:00
 * Version : 1.0v
 *
 * Description : Use dxflib to import entities from DXF to OpenCascade.

```

```

*
*   Key Words : OpenCascade, dxflib, DXF, DataExchange
*
*/

#include "DxfReader.h"

#include <BRepTools.hxx>

#pragma comment(lib, "TKernel.lib")
#pragma comment(lib, "TKMath.lib")
#pragma comment(lib, "TKBRep.lib")
#pragma comment(lib, "TKGeomBase.lib")
#pragma comment(lib, "TKTopAlgo.lib")

int main(int argc, char* argv[])
{
    std::string strFile = ((argc > 1) ? argv[1] : "test.dxf");

    DxfReader dxfReader(strFile);

    ofstream brepFile("dxf.brep");

    BRepTools::Write(dxfReader.GetShape(), brepFile);

    return 0;
}

```

程序效果还不错，给出几个程序处理的例子如下图所示：

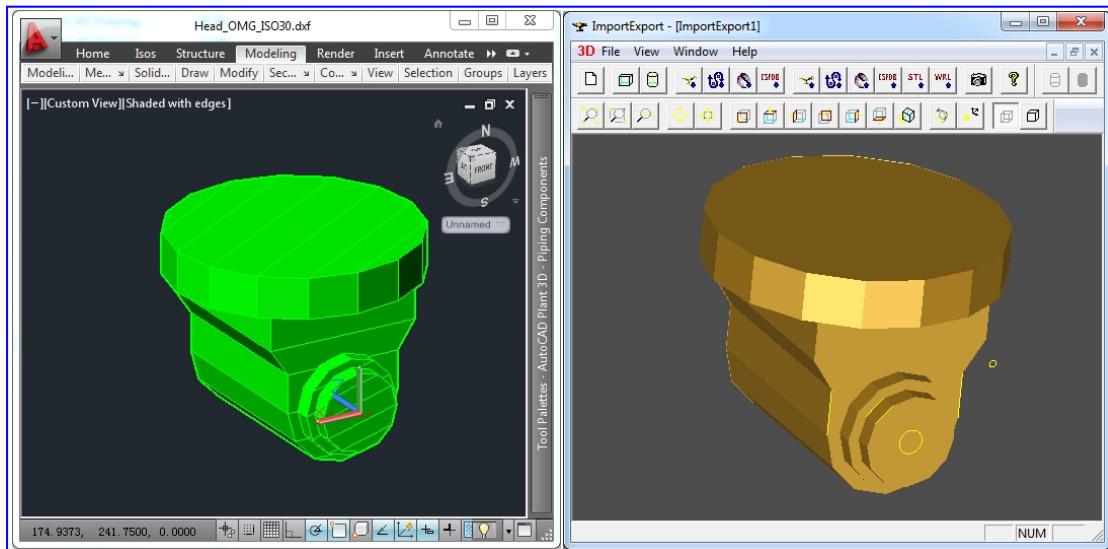


Figure 4.1 Use DxfReader import DXF to OpenCascade

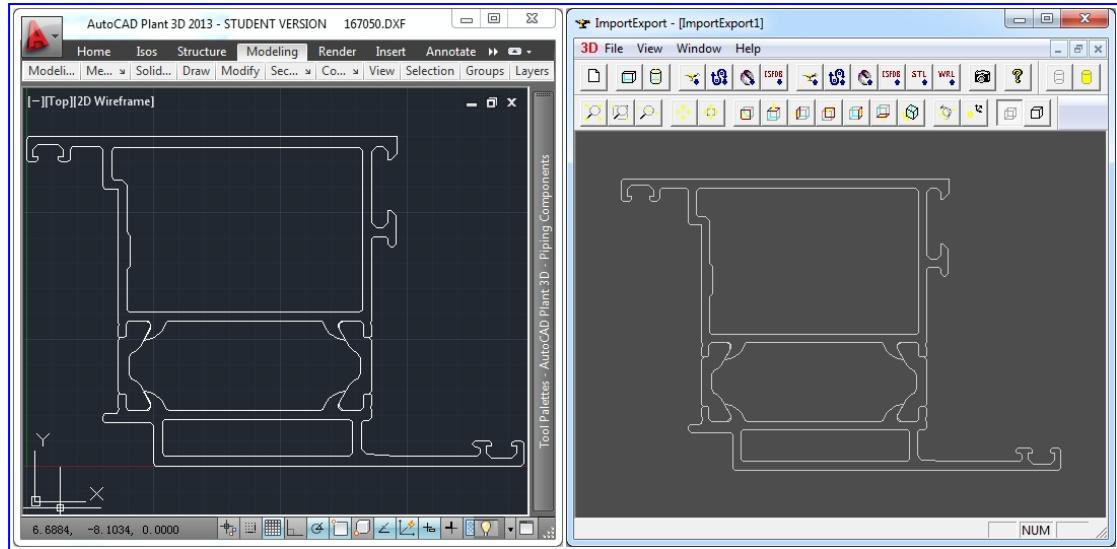


Figure 4.2 Use DxfReader import DXF to OpenCascade

五、结论 Conclusion

DXF 格式是图形数据交换的一种常见格式，DXF 文件的读写对 CAD/CAM 程序很有意义。本文通过把 dxflib 简单封装把 DXF 中图形转换为 OpenCascade 中的形状，为免费在 OpenCascade 中进行 DXF 数据交换提供了一个思路。希望本文对你有所帮助。

六、参考资料

1. 使用开源库读取 DWG 和 DXF 文件，
<http://www.ibm.com/developerworks/cn/opensource/os-autocad/>
2. eryar, 用 C 读取 DXF 文件, <http://blog.csdn.net/eryar/article/details/2371755>
3. eryar, Read DXF File, <http://www.cppblog.com/eryar/archive/2013/03/26/162303.html>
4. openDWG, <http://www.opendesign.com/>
5. OpenCascade, DXF Import-Export, <http://www.opencascade.org/support/products/dataex/dxf/>