架构示意图如下:



版图数据：

定义版图中的基本元素、层、版图场景的结构，一个版图就是一个版图场景对象。其他任何模块都是针对版图场景数据操作。

文件读写模块：

把各种格式的版图文件读入到通用的版图数据结构，或者把通用的版图数据结构写入到指定的格式文件。

运算模块：

根据需要把版图数据中的某个元素，或者某个层，甚至整个版图数据进行运算，诸如旋转平缩放等。

视图和UI交互：

显示版图场景，绘制版图中的各种元素如直线，圆等。包括各种控制的操作。

目前存在的问题：

1. 并没有一个通用的版图数据结构，数据只是简单实用QGraphicsItem子对象来表示并保存，这样明显是没法实际操作的。以后的一些运算仿真就会比较麻烦甚至没法做。
2. 对于版图中各种元素绘制，也是直接采用QGraphicsItem子对象来表示图元。这种方式的优点节省了代码量，很多细节不需要关心。缺点是资源耗费增大，有些特殊应用可能比较难以实现，例如也许修改会比较麻烦。我还要评估一下这样的效率和资源耗费。以及坐标转换后是否导致精度损失。
3. 目前实现的操作好像都是针对单个图元，不能针对一组甚至全部数据。例如undo。
4. 源代码目录结构不要随意变更

版图文件数据读写模块的代码逻辑结构



文件数据读取基类CGeoFileBase代码在GeoFile.h中，任何格式的文件实现代码必须派生与CGeoFileBase类，必须实现两个基本的虚函数ReadDataFromFile()和WriteDataToFile()，前者是把文件数据解析成场景数据CGeoScene，后者把场景数据保存为指定格式的文件。除了两个基本的虚函数，以后可能还要添加其他的基础函数，例如GetErrorInf()，用于读取或者写入文件失败后，获取更详细的错误信息。

一个派生类至少要实现一种文件格式的读取，当然也可以同时实现几种格式的读取。对于任何格式文件的文件解析，必须实现读取函数ReadDataFromFile()用于读取指定格式的文件数据。但是根据实际情况是否特别需要，可以有选择的实现WriteDataToFile ()函数。

任何派生类都必须有缺省构造函数，因为文件读写操作/遍历器根据不同格式生成对应派生类对象时调用的是缺省构造函数。

文件读写操作/遍历器子程序包括一系列的宏定义和两个读写函数，具体列举如下：

1. 全局函数ReadDataFromFile()和WriteDataToFile()
2. 宏定义DECLARE\_FILEFORMAT(class\_name)声明文件格式操作链接
3. BEGIN\_REGISTRATION(class\_name)和END\_REGISTRATION(class\_name) 宏定义用于实现文件格式操作链接
4. REGISTRATION\_EXT(class\_name,ext\_name)宏定义用于添加该类支持哪些格式解析。

有了这些宏定义后，以后扩展或者删除其他格式解析的类代码时，只需要针对具体的格式解析类做相关的操作，其他的代码都不需要修改。因为有了这些宏的定义，ReadDataFromFile()和WriteDataToFile()函数就跟自动遍历所支持的格式类并且生成相应的类对象然后操作这些对象进行读取或者写入操作。

如ReadDataFromFile("aaa.dxf")。则内部根类型名找到CGeoDXFFile类并生成CGeoDXFFile对象，然偶把文件名参数以及其他的必要参数传递进入后调用CGeoDXFFile::ReadDataFromFile()把文件数据读取到CGeoScene对象中。WriteDataToFile()也类似。

这些宏定义是自动遍历的基础，如果没有这些宏定义，每次增加新的格式解析类或者删除已有的格式解析类， ReadDataFromFile()和WriteDataToFile()函数代码都要相应的修改。

宏定义的使用方法：

1. DECLARE\_FILEFORMAT(class\_name)宏定义添加在每个派生的具体格式文件解析类的定义末尾，其中class\_name为派生类的类名。如下：

class CGeoDXFFile : public CGeoFileBase

{

…… //其他代码

DECLARE\_FILEFORMAT(CGeoDXFFile)

};

1. 在派生类的实现文件文件中的开头添加剩下的三个宏定义，BEGIN\_REGISTRATION，REGISTRATION\_EXT，END\_REGISTRATION。示例如下：

include "GeoDxfFile.h"

… //其他包含文件以及其他数据

BEGIN\_REGISTRATION(CGeoDXFFile)

REGISTRATION\_EXT(CGeoDXFFile, TEXT("dxf"))

REGISTRATION\_EXT(CGeoDXFFile, TEXT("dwg"))

END\_REGISTRATION(CGeoDXFFile)

1. 支持一种格式，就使用REGISTRATION\_EXT(class\_name,ext\_name) 添加。如果支持多种格式，则每种格式都调用一次，如上的示例代码同时支持dxf和dwg格式。class\_name表示派生类名，ext\_name表示支持的格式名称。

**注意：REGISTRATION\_EXT必须在BEGIN\_REGISTRATION和END\_REGISTRATION宏定义的中间。**