

**麻辣精灵GIS概要设计**

院系：地球与空间科学学院

班级：2017级五班

组名：麻辣精灵GIS

组员：姜金廷 李法承 李子锦 马涵聪

二〇二〇年四月

目录

[一、概述 3](#_Toc40013264)

[二、术语表 3](#_Toc40013265)

[三、用例 4](#_Toc40013266)

[四、设计概述 6](#_Toc40013267)

[4.1 简述 6](#_Toc40013268)

[4.2 系统结构设计 6](#_Toc40013269)

[4.2.1数据存取子系统结构设计 7](#_Toc40013270)

[4.2.2 外部数据输入输出子系统设计 9](#_Toc40013271)

[4.3 系统界面 11](#_Toc40013272)

[4.4 约束和假定 15](#_Toc40013273)

[五、对象模型 15](#_Toc40013274)

[六、对象描述 18](#_Toc40013275)

[七、动态模型 20](#_Toc40013276)

[7.1 场景 20](#_Toc40013277)

[7.2 状态图 23](#_Toc40013278)

[八、非功能性需求 26](#_Toc40013279)

# 一、概述

麻辣精灵GIS项目为本组尝试开发的小型GIS应用软件，模仿市面上已有的ArcGIS，QGIS等软件，开发一款能够进行数据存取、数据编辑、地图操作、地图查询以及外部数据输入输出的应用软件。

系统主要为一个前端与用户交互的应用界面，存储数据采用数据库和文件管理结合的形式。用户通过键鼠操作与应用界面交互，进行数据的读取、保存；对地理数据进行在图层、图形进行编辑修改，对要素字段进行增删改等操作；对要素实现属性查询，暂不实现空间查询功能；对图层进行多种可视化渲染，增强显示效果；同时还可以从本系统中导出图片，或输入文件，实现与其他系统的兼容效果。

软件目标平台为Windows平台，开发环境为Visual Studio ，使用基于.Net Framework的C#语言，项目通过Github平台共享协作，内容部分参考《软件工程》课程内容，及其他地理信息系统软件的相关内容。

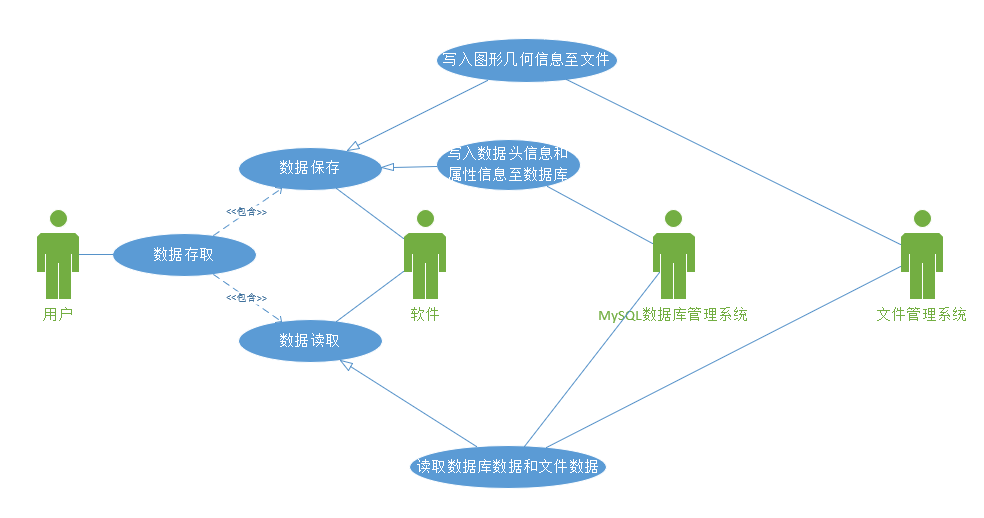
小组分工如下：

* 姜金廷——组长，负责数据存取，外部数据输入输出子系统
* 李法承——负责地图操作，数据编辑子系统
* 李子锦——负责专题地图子系统，及文档，软件的整饰设计
* 马涵聪——负责数据查询子系统，及文档，软件的整饰设计

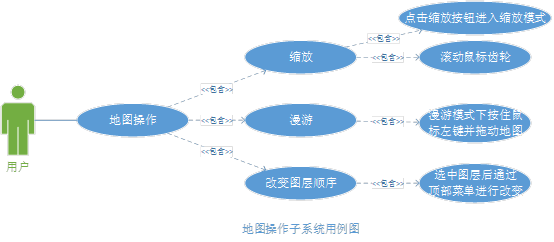
# 二、术语表

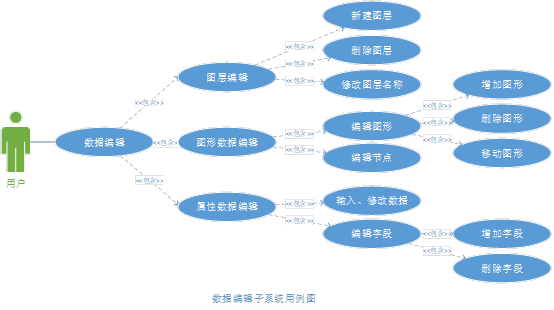
* Shp文件——shape文件由ESRI开发，一个ESRI（Environmental Systems Research Institute)的shape文件包括一个主文件，一个索引文件，和一个dBASE表。其中主文件的后缀就是.shp
* Shx文件——shape文件的索引文件
* Dbf文件——shape文件的属性表文件
* 要素——feature，指一个几何实体，在shp文件中存储于一条记录中
* 要素类——由相同类型的几何实体构成的集合，对应于一份shp文件
* 多段线——Polyline，指单个要素包含多段折线，每个折线由若干点连接形成，每个点只能连接1-2条线段
* 多边形——Polygon，指单个要素包含多个环形，每个环为闭合环，顺时针记录为外环，逆时针记录为内环。外环不能直接包含外环，内环不能直接包含内环。一个外环可以有多个相离内环（都包含于内环），一个内环只能有一个外环。
* 大端序——**高位**字节存入**低地址**，**低位**字节存入**高地址**
* 小端序——**低位**字节存入**低地址**，**高位**字节存入**高地址，一般的X86 PC系统存储格式**

# 三、用例



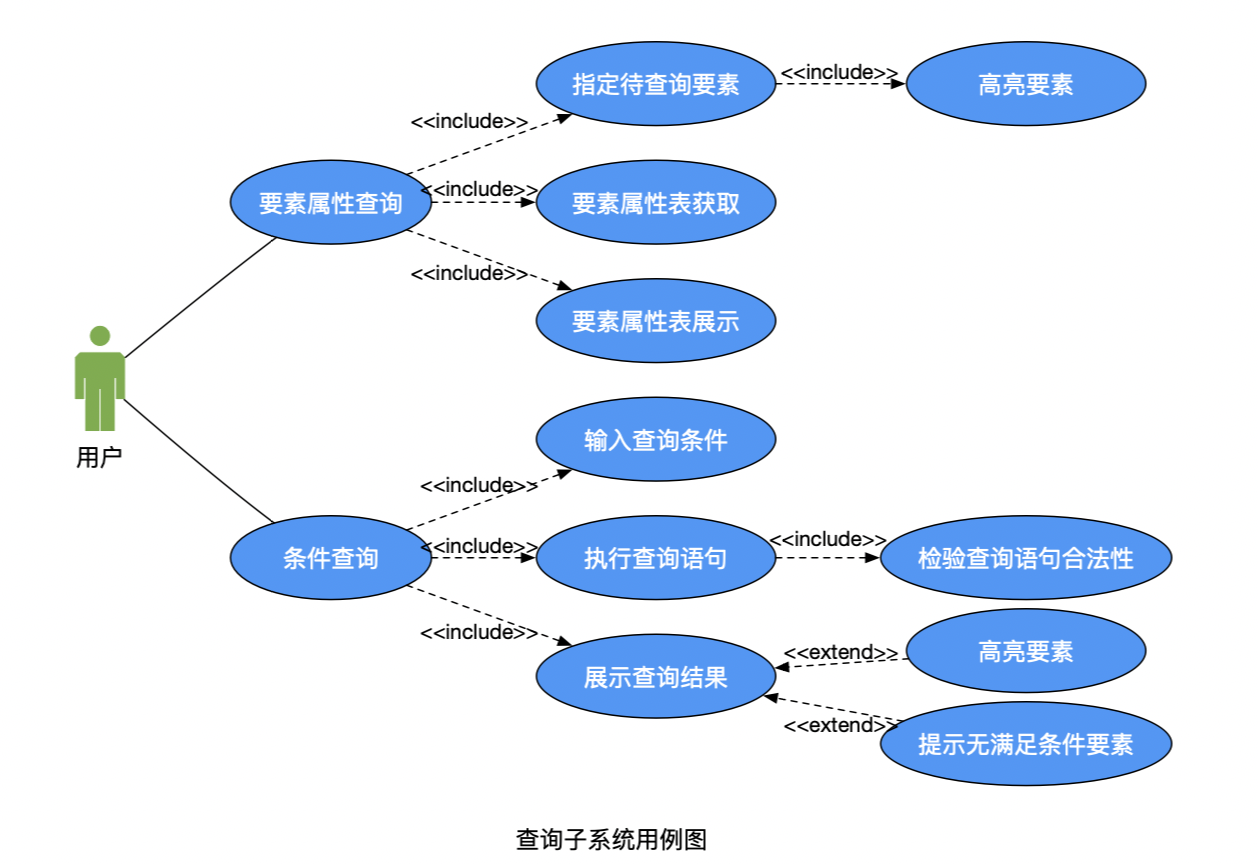
图表 1 数据存取用例图

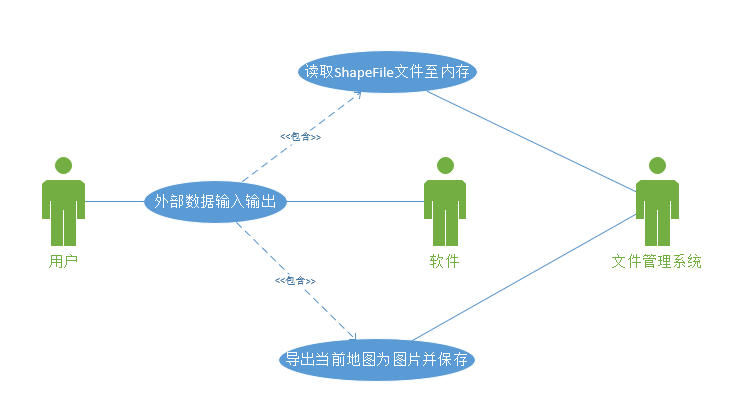






专题地图用例图





图表 2 外部数据输入输出用例图

# 四、设计概述

## 4.1 简述

软件总体采用面向对象的设计方法。采用客户端与MySQL数据库系统相连接，并与文件管理系统进行文件交互。数据库使用MySQL 8.0版本，数据库引擎为InnoDB，程序使用C#语言，Visual Studio进行项目管理，并且代码通过GitHub进行共享。

## 4.2 系统结构设计

本软件主要包括数据编辑、数据存取、专题制图、地图操作、数据查询、外部数据输入输出等子系统。

地图操作不改变数据本身，但是改变地图的显示内容。用户可通过缩放、漫游来观察重点内容，并可通过改变图层顺序来改变其叠加方式。地图制图的内容由窗口内容决定，因此也必须通过地图操作来进行调整。

数据编辑子系统与数据存取子系统同为GIS系统的数据来源，它不仅能修改所读取的数据，还能创造新的矢量数据与属性数据。数据编辑的对象既可以是要素，也可以是图层，它能灵活地对GIS数据进行调整，从而得到用户想要的数据。

专题地图子系统主要用于制作专题地图，将现有数据设置为自己喜欢的风格，根据数据特点制成专题地图，以便进行相关的研究。

### 4.2.1数据存取子系统结构设计

数据文件采用MySQL数据库管理系统和文件管理系统结合使用。

数据库建立名为’MalaSpiritGISDB’的模式（Schema），数据库引擎为InnoDB，字符集为UTF8MB4。模式中创建1+n张表格（Table），第一张表为头表header，记录了系统中n个需要存储的要素类信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 特性 | 备注 |
| ID | UNSIGNED INT | Not null; primary key; Auto Incremental; Unique | 唯一标识所有的要素类 |
| Type | ENUM(‘POINT’, ’POLYLINE’, ‘POLYGON’, ‘MULTIPOINT’) | Not null | 记录系统可以实现的类型 |
| Name | VARCHAR(45) | Not null | 要素类的名称，也用作图层显示时的名称 |
| Count | UNSIGNED INT | Not null | 要素类中的要素数量 |
| Xmin | DOUBLE | Not null | 要素类的MBR |
| Ymin | DOUBLE | Not null | 要素类的MBR |
| Xmax | DOUBLE | Not null | 要素类的MBR |
| Ymax | DOUBLE | Not null | 要素类的MBR |
| FilePath | VARCHAR(100) | Not null | 几何文件的路径（绝对） |

图表 3 头表的设计

其余n张表每个为记录一个要素类的属性信息的表格，表格名字为对应要素类的Name字段，表格的行数为对应要素类的Count字段。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 特性 | 备注 |
| ID | UNSIGNED INT | Not null; primary key; Auto Incremental; Unique | 唯一标识所有的要素 |
| FileBias | UNSIGNED INT | Not null | 该要素在几何文件中的偏移位置（from begin） |
| FileLength | UNSIGNED INT | Not null | 该要素在几何文件中的长度 |
| Other attribute | … | … | 其他需要记录的属性信息，类型由代码决定 |

图表 4 属性表的设计

每个要素类的几何文件存储于文件管理系统之中，为提高系统的兼容性，采用\*.shp文件格式进行存储，部分字段本系统默认不实现，用0代替。

\*.shp文件格式介绍如下。

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0-3 | 文件编号 | Int32 | 大端序 |
| 4-23 |  |  | 未被使用 |
| 24-27 | 文件长度 | Int32 | 大端序，包括文件头 |
| 28-31 | 版本 | Int32 | 小端序 |
| 32-35 | 图形类型 | Int32 | 见下表 |
| 36-67 | MBR | Double | Xmin ymin xmax ymax |
| 68-83 | Z坐标值范围 | Double | Zmin,zmax 不使用 |
| 94-99 | M坐标值范围 | Double | Mmin max 不使用 |

图表 5 shp文件头格式

|  |  |
| --- | --- |
| 图形类型 | 类型值 |
| 空图形 | 0 |
| Point 点 | 1 |
| Polyline 多段线 | 3 |
| Polygon 多边形 | 5 |
| MultiPoint 多点 | 8 |
| 其他类型不实现 |  |

图表 6 shp文件图形类型

文件头部分一共100个字节，接下来为若干个变长记录字段，每个字段代表一个要素，且不同要素的字段长度不同，详情见下方多表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0-3 | 编号 | Int32 | 大端序 |
| 4-7 | 记录长度 | Int32 | 大端序，不包括编号和长度本身 |
| 8-11 | 图形类型 | Int32 | 点为1 |
| 12-19 | 点x坐标 | Double |  |
| 19-26 | 点y坐标 | Double |  |

图表 7 点Point记录格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0-3 | 编号 | Int32 | 大端序 |
| 4-7 | 记录长度 | Int32 | 大端序，不包括编号和长度本身 |
| 8-11 | 图形类型 | Int32 | 多段线为3 |
| 12-43 | MBR | Double | Xmin ymin xmax ymax |
| 44-47 | PartNum | Int32 | 多段线的段数 |
| 48-51 | PointNum | Int32 | 多段线的点数 |
| 52- | Parts | Int32[] | 每段线的起始点下标，一共PartNum个下标，总长度为4\*PartNum |
| - | Points | PointD[] | 每个点的坐标，一个点为两个double值，x和y。总长度为16\*PointNum |

图表 8 多段线Polyline记录格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0-3 | 编号 | Int32 | 大端序 |
| 4-7 | 记录长度 | Int32 | 大端序，不包括编号和长度本身 |
| 8-11 | 图形类型 | Int32 | 多边形为5 |
| 12-43 | MBR | Double | Xmin ymin xmax ymax |
| 44-47 | PartNum | Int32 | 多边形的环数 |
| 48-51 | PointNum | Int32 | 多边形的点数 |
| 52- | Parts | Int32[] | 每个环的起始点下标，一共PartNum个下标，总长度为4\*PartNum。每段环的顺时针记录为外环，逆时针记录为内环。每个环的点记录为闭环，即一个三角形对应4个点记录。 |
| - | Points | PointD[] | 每个点的坐标，一个点为两个double值，x和y。总长度为16\*PointNum |

图表 9 多边形Polygon记录格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0-3 | 编号 | Int32 | 大端序 |
| 4-7 | 记录长度 | Int32 | 大端序，不包括编号和长度本身 |
| 8-11 | 图形类型 | Int32 | 多点为8 |
| 12-43 | MBR | Double | Xmin ymin xmax ymax |
| 44-47 | PointNum | Int32 | 多点要素的点数 |
| 48- | Points | PointD[] | 每个点的坐标。一个点为两个double值，x和y。总长度为16\*PointNum |

图表 10 多点MultiPoint记录格式

### 4.2.2 外部数据输入输出子系统设计

系统需要能够从外部读入shp文件，本系统需实现读入shp文件，shx文件，dbf文件以保证Shapefile文件内容足够齐全。上文已经介绍了shp文件的具体格式，此处继续介绍其余两份文件的格式。

Shx文件包含与shp文件相同的100个字节的文件头，及若干个8字节定长记录。每条记录为一个int32类型表示记录在shp文件中该要素的位移，一个int32类型表示记录在shp文件中该要素记录的长度。

Dbf文件结构如下，dbf文件分为两部分：文件结构说明区和数据区。文件结构说明区又分为数据库参数区和记录结构表区，各32个字节

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0 | 数据库版本 | Byte | 没看懂说明。。。但写0x03就行 |
| 1-3 | 文件修改日期 | Byte[3] | YY-MM-DD.YY=当前年份-1900 |
| 4-7 | 记录数count | Int32 | 大端序，即shp中的要素数量 |
| 8-9 | 第一条记录的位置 | Short | 等于文件结构说明区的长度 |
| 10-11 | 每条记录的长度length | Short | 包含记录的删除标志（delete flag） |
| 12-31 |  |  | 保留字节，不使用 |

图表 11 数据库参数区结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位置 | 字段 | 类型 | 备注 |
| 0-10 | 字段名 | Char[10] | 字段名不足10字节用0x00补齐 |
| 11 | 字段类型 | Byte | C    –    Character  N    –    Numeric  F    –    Float  D    –    Date  T    –    DateTime  B    –    Double  I    –    Integer L    –    Logical  以上为系统需要用到的类型 |
| 12-15 |  |  | 保留字节 |
| 16 | 字段记录的长度 | Byte |  |
| 17 | 字段记录精度 | Byte |  |
| 18-31 |  |  | 保留字节，不使用 |

图表 12 记录结构表区结构

文件结构说明区后为数据区，每个要素的记录按顺序用二进制记录。最终dbf的文件大小为32+m\*32+1+n\*length+1

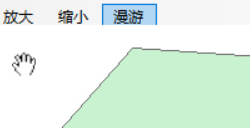
其中两个1分别对应这两个区域的结束标志0x0d和0x1a

## 4.3 系统界面

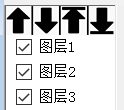
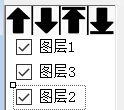
（1）地图操作界面：



点击菜单栏的“放大”或“缩小”按钮即可进入缩放模式，鼠标会变成缩放图标，此时在显示区域内点击鼠标左键会以点击点为中心放缩。另外也可通过滚动鼠标中键来实现类似的放缩效果。

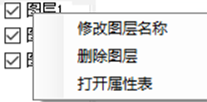


漫游状态下，按住鼠标左键时鼠标会自动变成漫游图标，此时鼠标进行移动即可实现漫游。

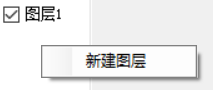


选中图层后，点击图层栏上方的“上移”、“下移”、“置顶”、“置底”按钮即可调整图层顺序。

（2）数据编辑界面



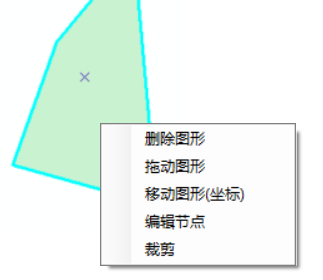
在图层所在区域点击鼠标右键，可打开属性表、修改图层名称或删除图层。



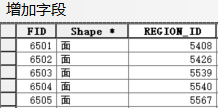
在图层空白处点击鼠标右键，可选择新建图层。



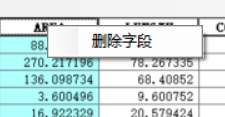
点击菜单栏“创建要素”即可增加图形。单击鼠标左键确定折点或创建点要素，双击左键创建线要素或面要素。之后可对要素进行操作，使其变成复合折线或复合面。



点击菜单栏“选择要素”即可开始选中要素。“选择要素”状态下，右键选中要素可删除图形、拖动图形、输入偏离值使图形移动、编辑线要素和面要素的节点，并可使线要素变为复合折线或使面要素裁剪为复合面。



打开属性表后，可选择增加字段。

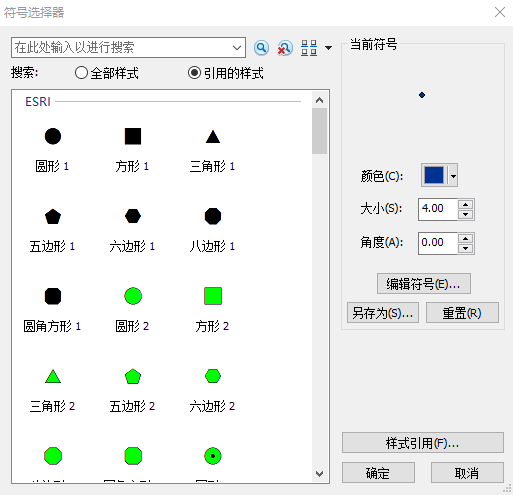


选中字段后右键，可删除字段。另单击某属性值后即可进行修改。

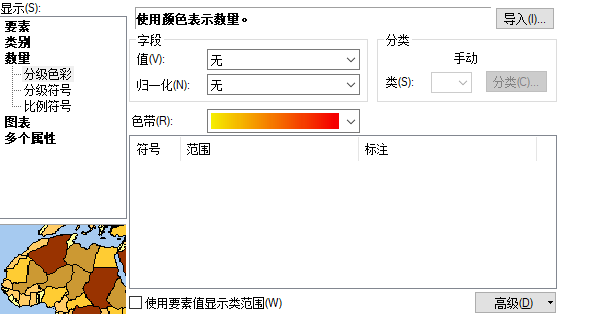
（3）专题地图界面：

双击图层下方的点/线/面图标，即可修改要素符号类型。

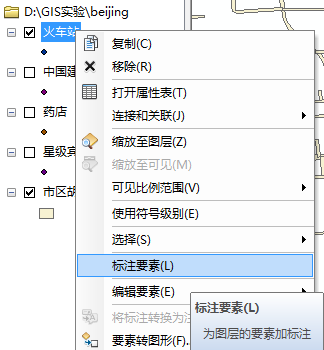
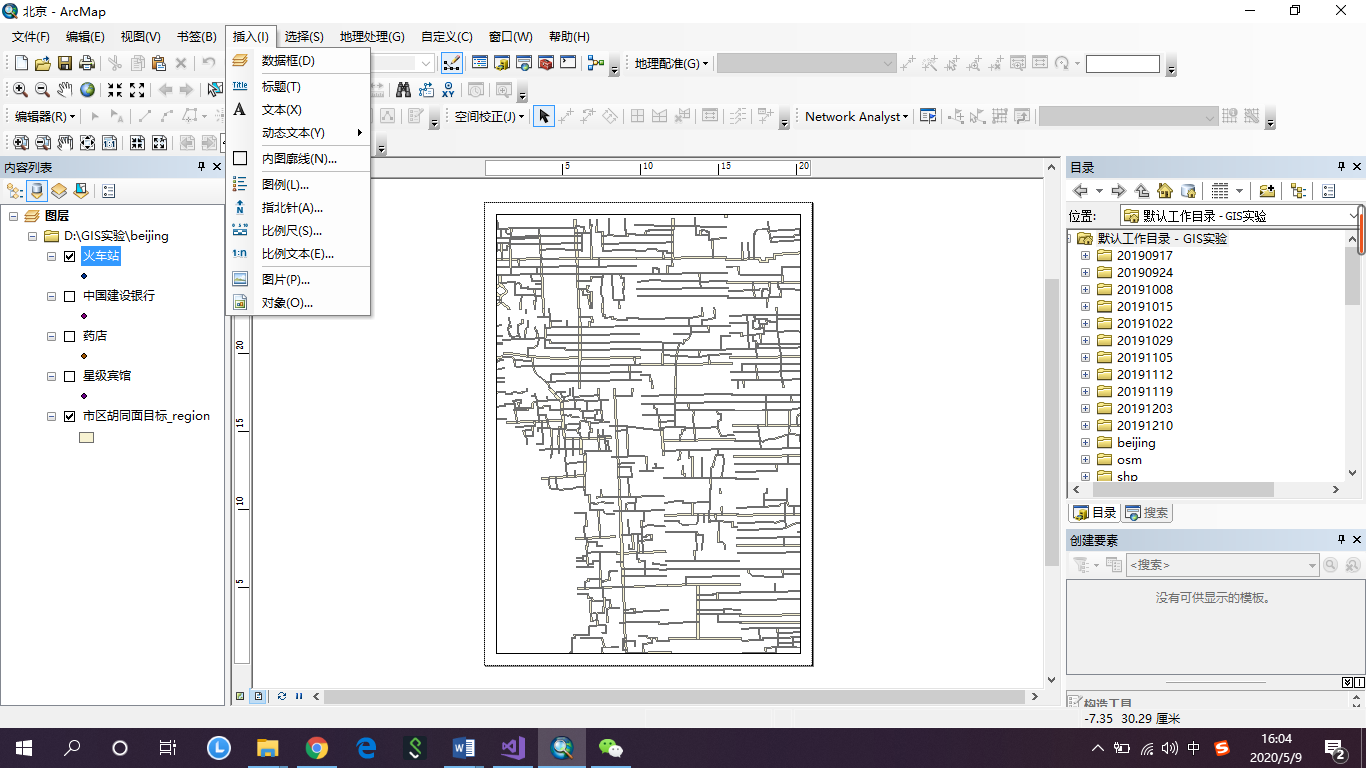




在图层列表中右击图层，选择要素，在显示系统里面，选择渲染字段和渲染方法，即可对图层进行渲染。

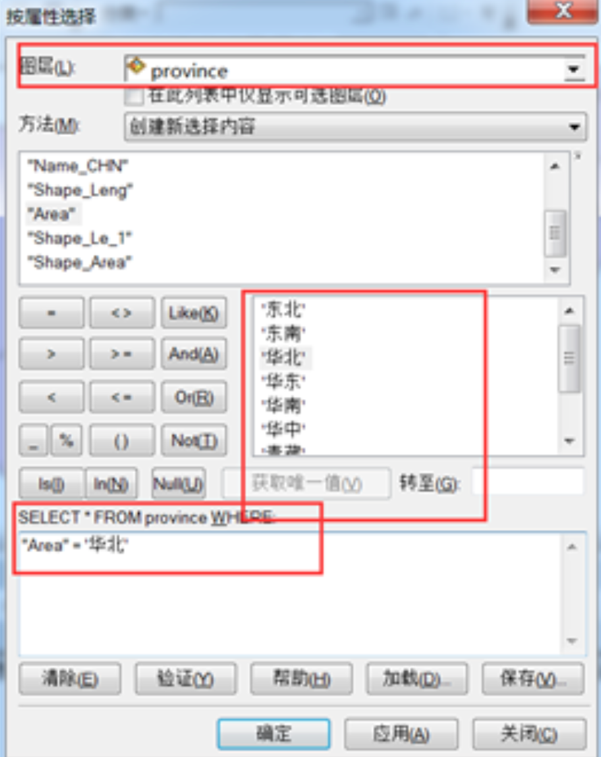


右击图层，选择标注要素，即可添加注记。或者在主菜单的插入栏中，选择添加文本等。

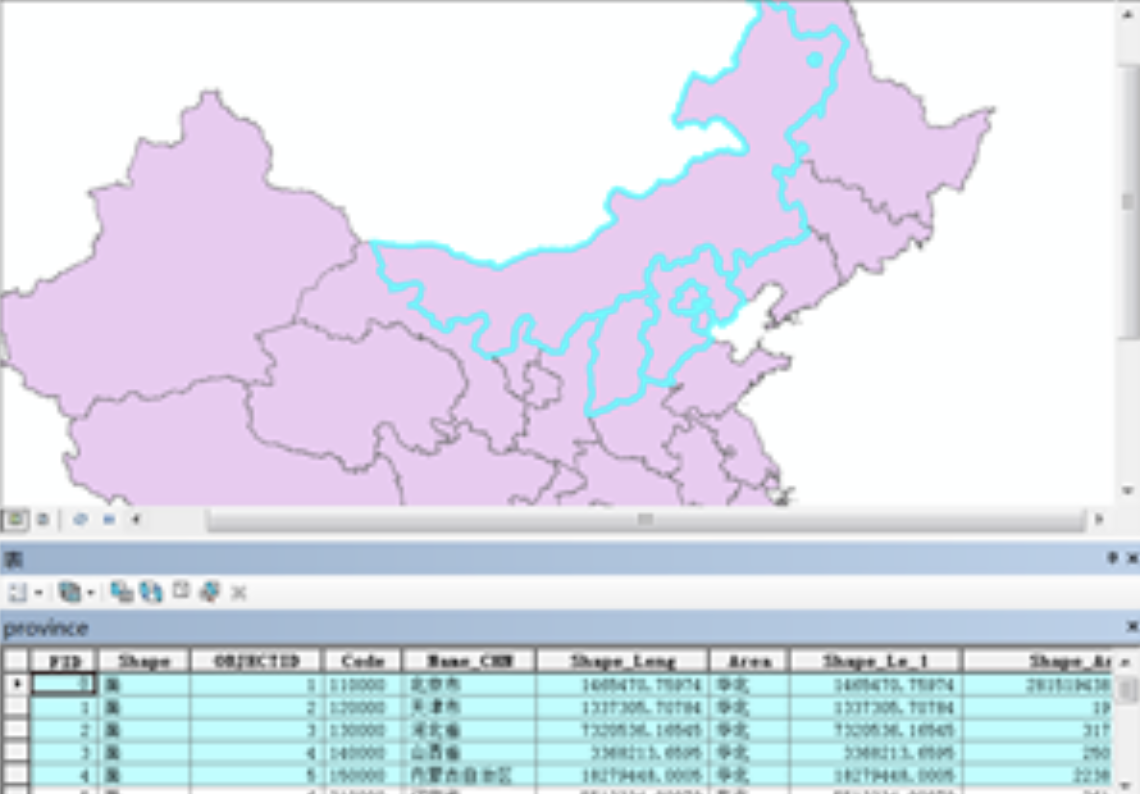
 

（4）条件查询界面

用户在查询窗口中下拉“图层”选择查询的目标图层。系统自动显示当前图层中对象的所有属性的字段名，用户双击字段名进行输入。选择字段名后系统自动显示该字段对应的唯一值。用户单击选择符号，再双击选择唯一值或输入查询目标。最后点击“应用”按键运行该表达式。



表达式运行完毕，若存在图形对象满足表达式，在画布中高亮对应的图形对象。若用户对属性表进行查看，图形对象的所有属性也会被高亮。



## 4.4 约束和假定

（1）输入坐标移动图形或输入属性值时，数值的精确度为10-6。这是受float浮点型的精确度所限制的。

（2）点符号类型只包括；线符号类型只包括实线和虚线；多边形符号类型只包括颜色填充。

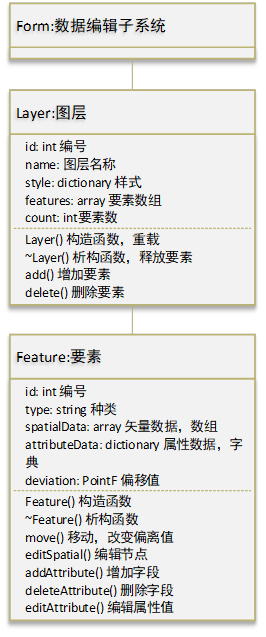
（3）Shapefile系列文件最大大小不应超过2GB，且内部的字段也有相应的长度规定。而本系统的数据库管理的长度限制相对宽松，能够进行数量较多的存储。

# 五、对象模型

1.地图操作子系统



2.数据编辑子系统



3.专题地图子系统



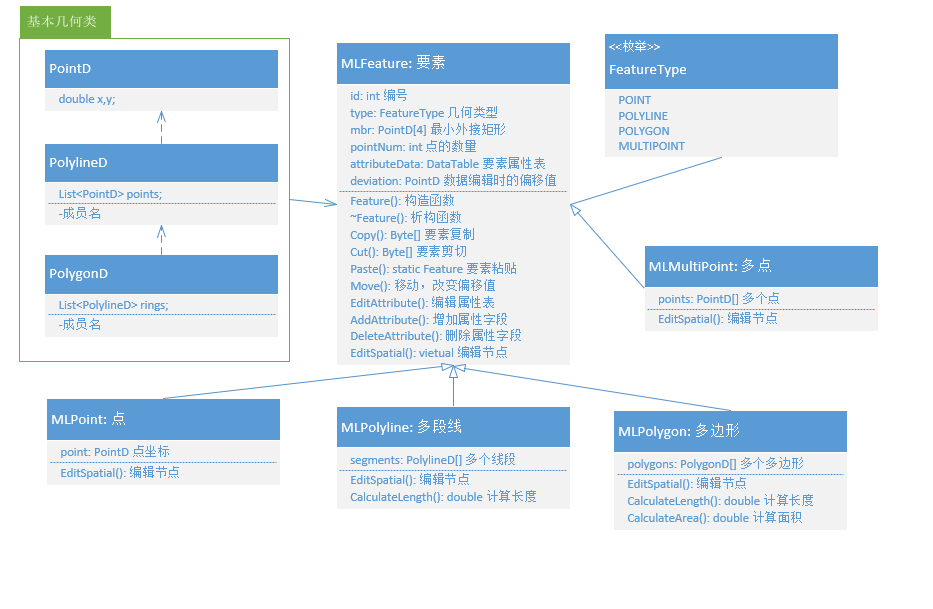


Figure 1 数据存取对象细化模型

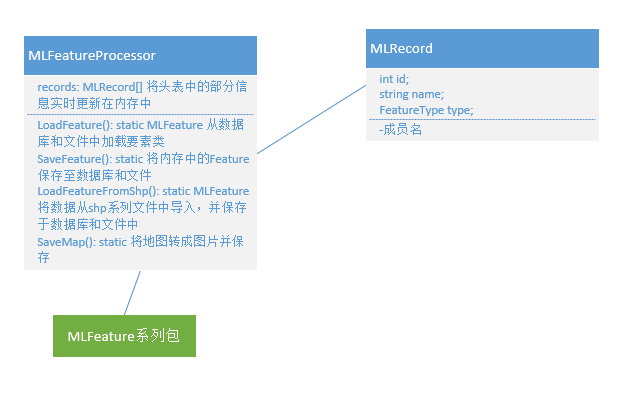


Figure 2 数据存取处理器对象模型

# 六、对象描述

1、名称：地图

属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 说明 |
| name | string | 地图的名称 |
| rate | int | 地图的放缩比例 |
| deviation | PointF | 地图的漫游偏移 |
| layers | array | 图层数组，即所绑定的图层 |

方法：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 参数 | 返回类型 | 返回值 | 用途 |
| Map | 无/文件地址、数据库地址 | 无 | 无 | 新建一个空白地图，或从图层创建新的地图 |
| ~Map | 无 | 无 | 无 | 释放资源 |
| move | PointF | 无 | 无 | 地图进行漫游时改变偏移值deviation |
| changeRate | int | 无 | 无 | 地图进行放缩时改变显示比例尺 |
| changeOrder | array | 无 | 无 | 改变图层的顺序 |

2、名称：图层

属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 说明 |
| id | int | 图层编号，用于唯一识别 |
| name | string | 图层名称，用于用户标注 |
| style | dictionary | 样式字典，用于地图制图 |
| features | array | 要素数组，即所绑定的要素 |
| count | int | 所包含的要素数量 |

方法：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 参数 | 返回类型 | 返回值 | 用途 |
| Layer | 无/文件地址、数据库地址 | 无 | 无 | 新建一个图层 |
| ~Layer | 无 | 无 | 无 | 释放资源 |
| add | Feature | 无 | 无 | 增加一个要素 |
| delete | int | 无 | 无 | 删除索引对应的要素 |

3、名称：要素

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 说明 |
| id | int | 要素编号，用于唯一识别 |
| type | FeatureType | 识别要素的种类 |
| mbr | PointD[4] | 最小外接矩形 |
| pointNum | int | 点的数量 |
| attributeData | DataTable | 要素的属性数据 |
| deviation | PointF | 要素矢量数据的偏移值 |

方法：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 参数 | 返回类型 | 返回值 | 用途 |
| Feature | spationData,attributeData | 无 | 无 | 生成要素 |
| ~Feature | 无 | 无 | 无 | 释放资源 |
| move | PointF | 无 | 无 | 改变偏离值 |
| editSpatial | PointF | 无 | 无 | 编辑节点 |
| addAttribute | string | 无 | 无 | 增加字段 |
| deleteAttribute | string | 无 | 无 | 删除字段 |
| editAttribute | string | 无 | 无 | 编辑属性值 |

4、名称：要素处理器

属性

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 说明 |
| Records | MLRecord[] | 将头表中的部分信息实时更新在内存中 |

方法

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 参数 | 返回类型 | 返回值 | 用途 |
| LoadFeature | Int | MLFeature | 加载的要素对象 | 从数据库和文件中加载要素类 |
| SaveFeature | Int | 无 | 无 | 将内存中的Feature保存至数据库和文件 |
| LoadFeatureFromShp | String | MLFeature | 加载的要素对象 | 将数据从shp系列文件中导入，并保存于数据库和文件中 |
| SaveMap | 无 | 无 | 无 | 将地图转成图片并保存 |

# 七、动态模型

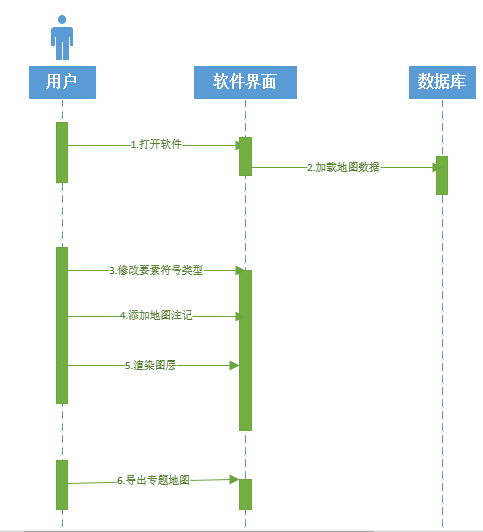
## 7.1 场景

场景一：用户新建了一个空白的面图层，然后点击“创建要素”，通过移动鼠标并点击鼠标左键绘制了一个多边形，双击鼠标左键完成绘制。之后用户点击“选择要素”，选中了多边形并右键打开菜单，选择“编辑节点”，移动了多边形一个节点的位置。再之后右键多边形选择“拖动图形”，将多边形向右上方挪动了一些位置。紧接着用户打开该图层的属性表，新建了一个字段“海拔”，点击对应位置输入了数据“50”，然后删除了字段。再之后用户右键多边形选择了“删除多边形”，右键图层选择了“删除图层”。最后退出程序。



场景二：

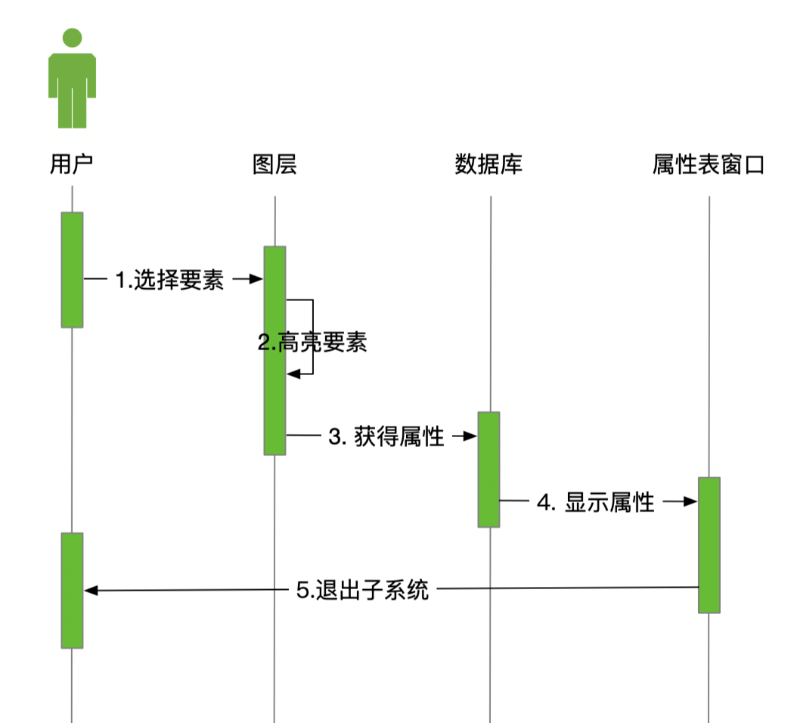
专题制图：用户可以修改要素符号类型，添加地图注记，并选择对图层进行渲染等操作，以制作专题地图。其顺序图如下：



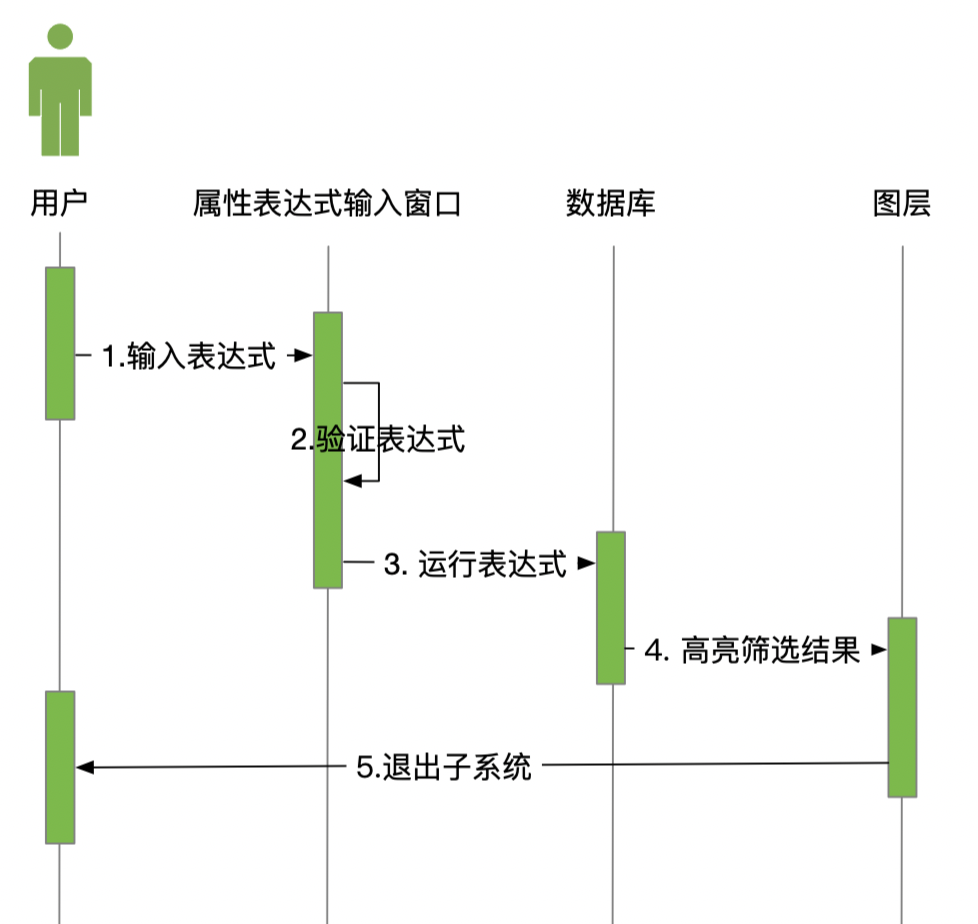
场景三：用户开始操作一个包含两个图层的地图。他先点击了“放大”按钮，然后对准某个要素点击鼠标左键，地图放大了。之后他点击了“漫游”按钮，然后按住鼠标左键拖动地图，使地图左下角的内容居中。最后他在图层栏点中第二个图层，点击“上移”键后使第二个图层位于最顶层，并退出了程序。



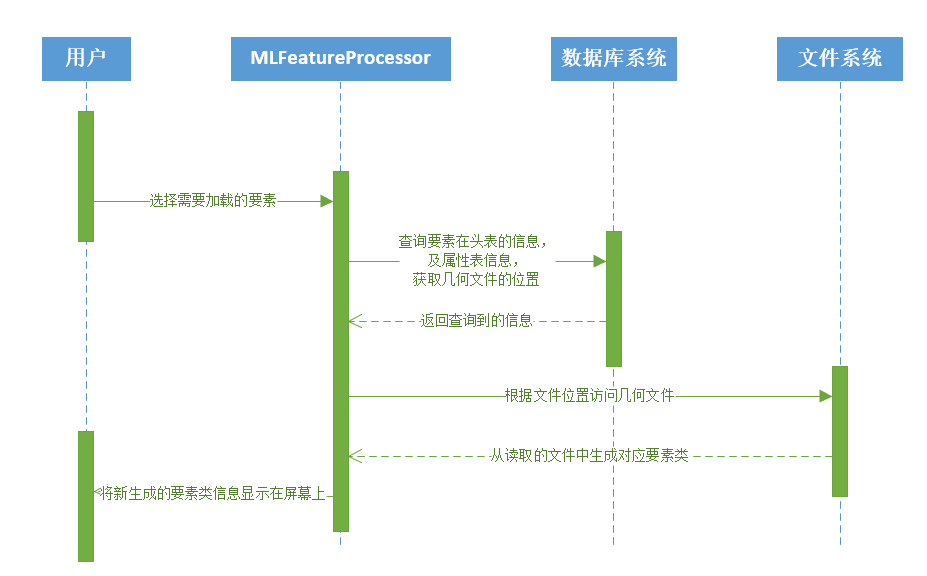
场景四：用户在图层中选择要素，选中后用户点击属性按键，弹出属性表窗口进行查看后用户关闭该窗口。



场景五：用户使用条件查询功能，在属性表达式输入窗口中输入正确的表达式并点击运行，查看图层中高亮的筛选结果后关闭属性表达式输入窗口。

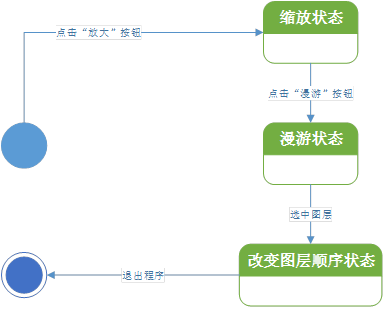


场景六：用户选择要素类，导入到系统内存中。

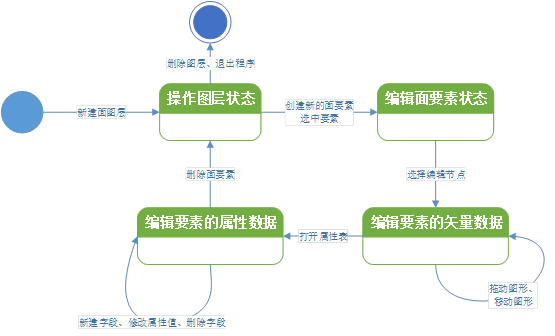


## 7.2 状态图

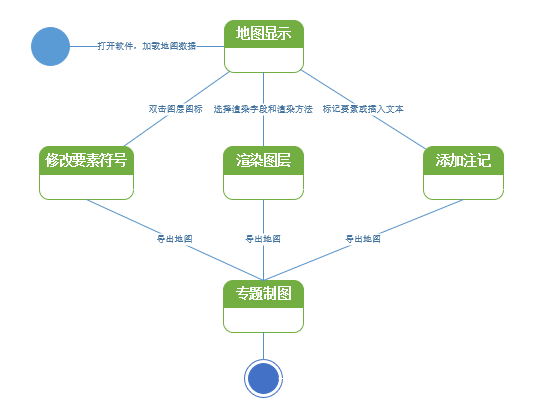
1.地图操作：



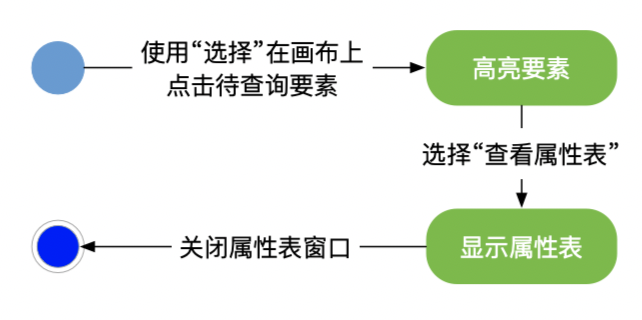
2.数据编辑：



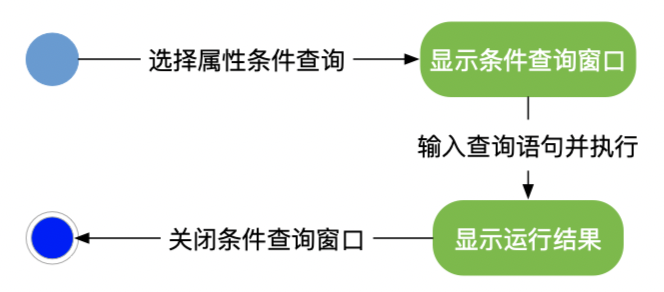
3.专题地图



4. 查看属性



5. 条件查询



# 八、非功能性需求

1.数据编辑的便捷性：提供更多途径实现完成绘制，如在双击之外新增一个同样功能的按钮。

2.地图操作的便捷性：提供更多途径实现完成放缩，如在放大模式下向左拖动仍然可以实现缩小。