

利用 FME 进行 CAD 数据转换及属性赋值

—— FME 模板兴趣班 2020 年第一期心得总结

摘要：FME 是数据处理尤其是 GIS 数据处理的利器。本文以 FME 模板兴趣班 2020 年第一期作业内容为例，并加入了一些限制条件，讲述了如何利用 FME 对 CAD 数据进行提取，并转入 GIS 数据库；如何对表格数据进行行列转置并动态输出；以及如何利用转换器生成箭头方向的三角符号等。从数据分析到模板实现，综合利用了 FME 原生的各种转换器，在保证结果的前提下，也保证了运行效率。受限于条件，某些流程会相对复杂，模板不是最佳的，但整体的设计思路依然可以给广大的 FME 使用者特别是初学者提供一些帮助和灵感。

1. 引言

本期模板兴趣班出了个好题，给了我本文的思路。主题还是 CAD 数据提取、属性挂接和转入 GIS 数据库等，但是考查的内容却有不少，主要有如下几点。

- 1) 行列转置
- 2) CAD 读模块参数设置
- 3) CAD 面域转多边形
- 4) 邻近分析、空间分析及属性挂接
- 5) RGB 颜色转换、空值转换
- 6) 三角符号生成
- 7) 文件数据库 (*.gdb) 读写模块参数设置

2. 数据分析

2.1 CAD 图形测试数据分析

CAD 数据主要包含 a 地块面、b 地块编码、c 地块指标表、d 配套设施符号和 f 三角符号 5 个图层。

1) a 地块面图层：主要包含了常规多边形 (polygon) 和面域 (region) 两类数据，其中面域数据需要转换为多边形。在设置 CAD 读模块参数时，不能选择“Read Region Entities As Areas”，否则输出奇怪的多边形。

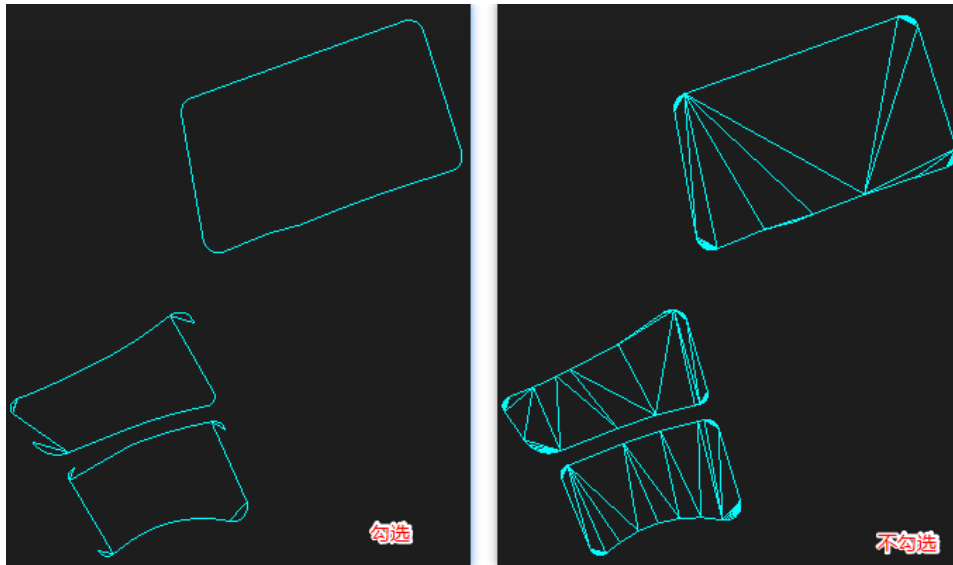


图 1 面域输出情况

2) b 地块编码：文本标注，分两种情况，一种是在地块里，可直接通过空间关系赋予地块编码属性；另一种是在地块外，要通过其指引标识来判断与地块的空间关联，间接赋予地块属性。

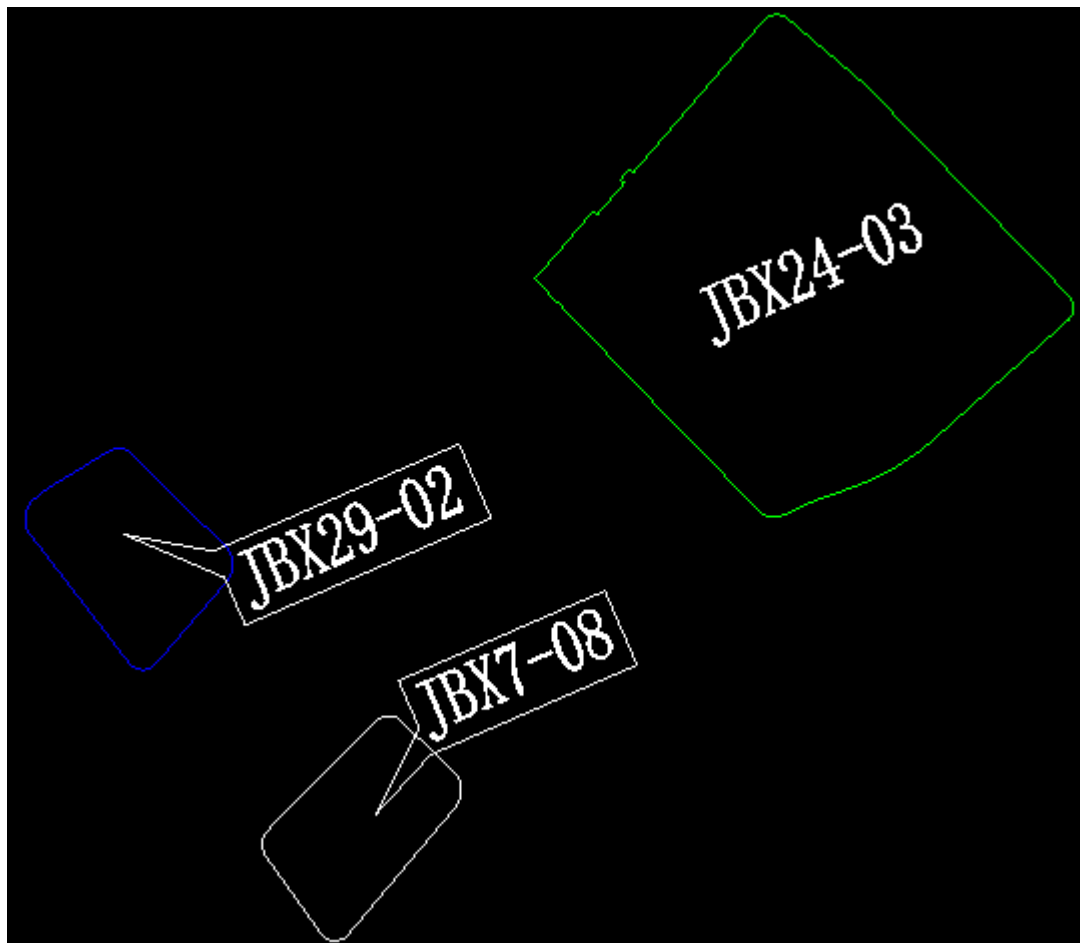


图 2 地块编码在地块内、外情况

3) c 地块指标表：可直接读取地块指标，处理方法与 b 地块编码图层相似但又不同。



图 3 指标表在地块内、外情况

4) f 配套设施符号：有些地块是组合在一起了，需要先打散；同时还有些标注文本框嵌到地块里了，需要移除。

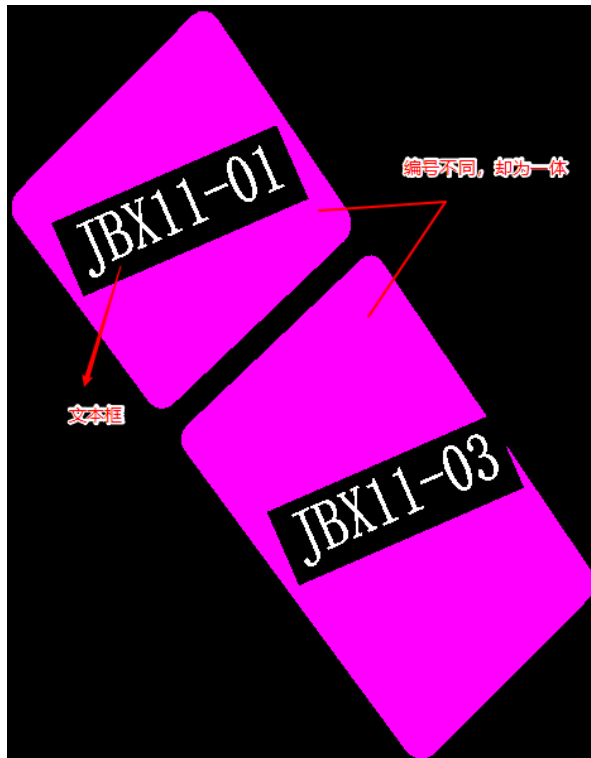


图 4 配套设施符号情况

5) f 三角符号：在 FME 里是看不出效果的，需要借助其他软件（如 CAD 迷你看图）来查看其箭头指向。

2.2 地块指标表分析

罗列了部分地块的详细指标，类似于 CAD 数据里的“c 地块指标表”，需要跟 CAD 里的图层进行挂接。

2.3 业务环节表分析

按作业要求，需要先做行列转置后，再与 CAD 里的图层挂接。

2.4 已有 GDB 数据分析

只有一个“a 地块面”图层数据，但需要注意几个问题：

1) 数据库的图层坐标跟原 CAD 里的坐标不一致，分析后，发现是 CAD 里采用了用户坐标系（UCS）。这里以 GDB 数据库里的坐标为准，在设置 CAD 读取参数时，需要勾选忽略用户坐标系选项。

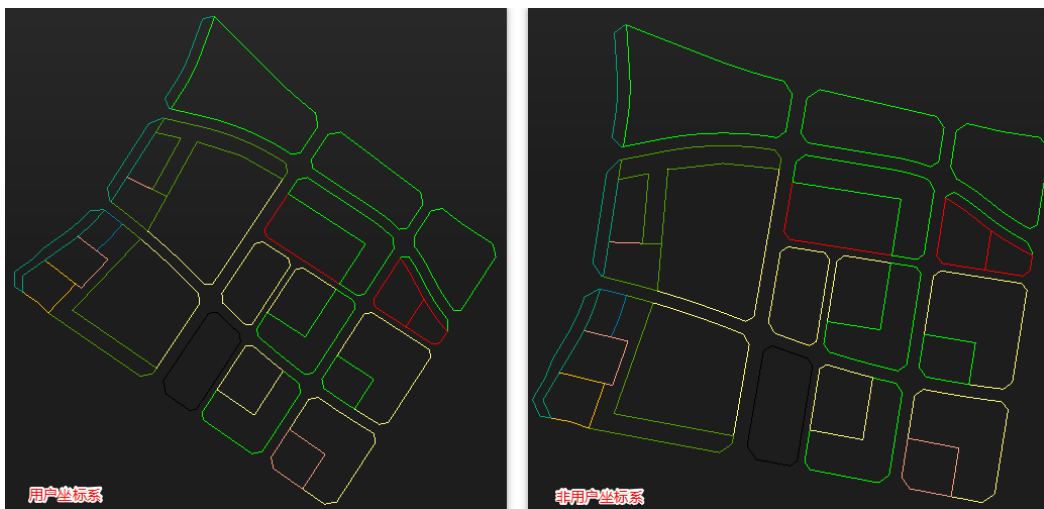


图 5 用户坐标&非用户坐标

2) 地块编号与跟地块的对应关系与 CAD 里的不一致，须以 CAD 里的为准。

3) 某些字段名称与 CAD 的“c 地块指标表”及 Excel 地块指标表的名称不一致，因要保证 GDB 里的图层结构，则需要对两个指标表的字段重命名。

4) 某些字段类型和字段长度可能存在问题。如绿地率，设定的是文本类型，并且长度为 1，而“地块指标表”里的绿地率是两位数，会导致数据截断，输出不准确，需要更改为整型或浮点型，其他字段类似。值得注意的是，GDB 数据库图层在拥有数据的情况下，是不允许直接修改字段类型的，而作业要求保证 OBJECTID 不做变化来进行更新，那么也不能使用先清空数据后再修改字段类型的方式。目前采用的是先删除问题列，再新建列的方式。

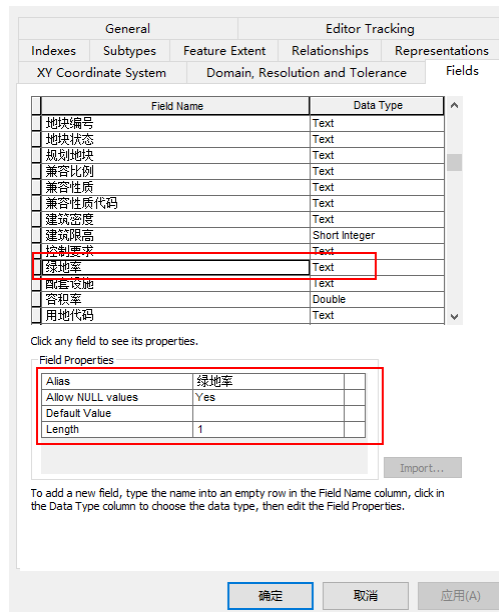


图 6 字段类型及长度问题

5) 长度与面积字段是内建字段，不能删除。而由于地块编号与地块对应错误，在根据地块编号更新数据时，需要先计算地块的长度和面积，再进行更新。

3. 总体流程

设计该模板的初衷是想要把自己掌握的转换器尽量用上，同时又能保证质量和效率。同时，为了最大程度的使用 FME 的各种转换器，也给自己定下了几个约束(说白了就是自虐...):

- 1) 不能使用代码编辑 (Caller) 类转换器，虽然有些流程使用 Caller 会简化很多。
- 2) 尽可能使用不同的转换器。换句话说，即同一个转换器尽量不要多次使用。
- 3) 在满足前两个条件的情况下，尽可能使用更少的转换器。
- 4) 只使用 FME 原生的转换器，不借助第三方 (Hub、MyFME 等) 的转换器。
- 5) 在保质保量的前提下，尽可能提高运行效率。

在模板设计的过程中，逐渐形成了本文思路，并且通过对测试数据的分析和按作业要求，最终设计了整个流程模板。

除邻近分析 (NeighborFinder) 转换器使用两次外，其余转换器均只使用了一次。

同时，还借助了一些节点连接器 (Junction) 和调整一些转换器输出端口的上下位置，来梳理整个流程布局，使得流程更加清晰。

具体模板可参照提交的模板作业 (懒人 GIS)。

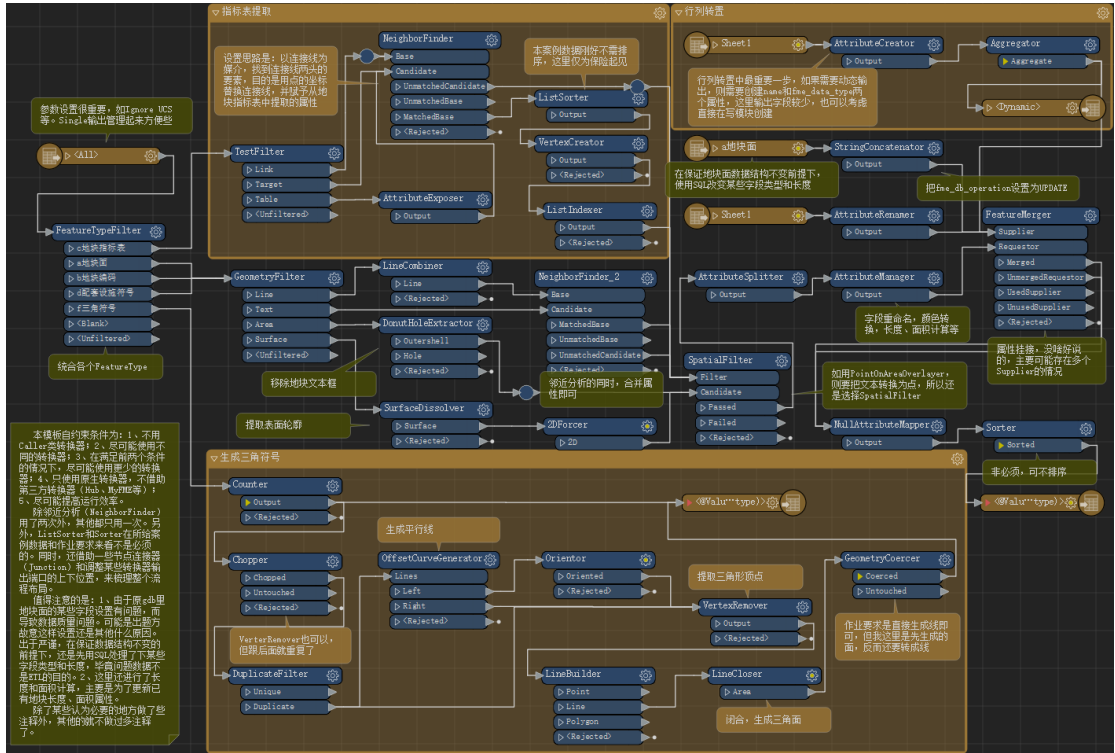


图7 总体流程

4. 流程分解

接下来，将对整个流程主要步骤进行详细说明。

4.1 业务环节表行列转置

关于 FME 行列转置，有好几种方法，在[《FME 处理表格转置的 3 种方式》](#)一文中提到，而我更倾向于使用 AttributeCreator + Aggregator 组合。这个组合的好处是局限性更小一些，可以处理多种数据类型，适配性更好。但在处理技术交流群里“BORG (PAN)”提供的 pivot_source.csv 数据时，却出现数据丢失的奇怪现象，现已找到了问题所在，打算后面再起一文详细讨论，敬请关注。

AttributeCreator 是可以通过 Attribute Value 创建列的，这是最重要的一步，而 AttributeManager 则不行。同时，如果想要动态模式输出的话，还可以顺便创建 Schema 结构，fme_data_type 可以设置为 date、char 和 varchar 三种，按作业要求，这里选用了 varchar。

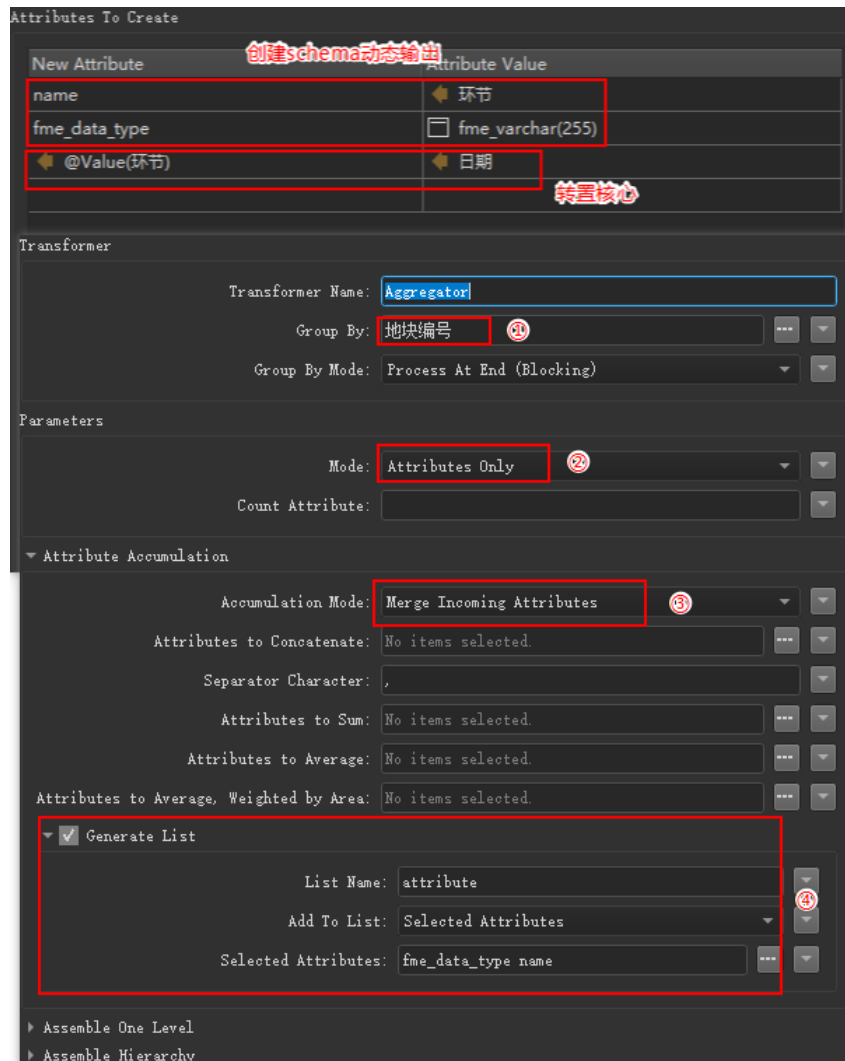


图8 行列转置

4.2 GDB 读/写模块参数设置

本来是打算把字段（主要字段）处理的 SQL 放在写模块的 SQL To Run Before Write 里的，但发现输出的字段内容全为空，暂时也没去探究原因，只好先把处理的 SQL 放到读模块里。另外，由于要利用“地块编号”属性更新以保证“OBJECTID”不便，写模块的数据库操作可以使用 fme_db_operation，同时为已有地块设置 fme_db_operation 属性为 UPDATE。

```

FME_SQL_DELIMITER ;
alter table "a地铁面" drop "备注" ;
alter table "a地铁面" drop "规划地块" ;
alter table "a地铁面" drop "兼容比例" ;
alter table "a地铁面" drop "兼容性质" ;
alter table "a地铁面" drop "建筑密度" ;
alter table "a地铁面" drop "建筑限高" ;
alter table "a地铁面" drop "控制要求" ;
alter table "a地铁面" drop "绿地率" ;
alter table "a地铁面" drop "配套设施" ;
alter table "a地铁面" add "备注" varchar(256) ;
alter table "a地铁面" add "规划地块" varchar(50) ;
alter table "a地铁面" add "兼容比例" varchar(50) ;
alter table "a地铁面" add "兼容性质" varchar(50) ;
alter table "a地铁面" add "建筑密度" decimal(18,6) ;
alter table "a地铁面" add "建筑限高" decimal(18,6) ;
alter table "a地铁面" add "控制要求" varchar(50) ;
alter table "a地铁面" add "绿地率" decimal(18,6) ;
alter table "a地铁面" add "配套设施" varchar(256) ;

```

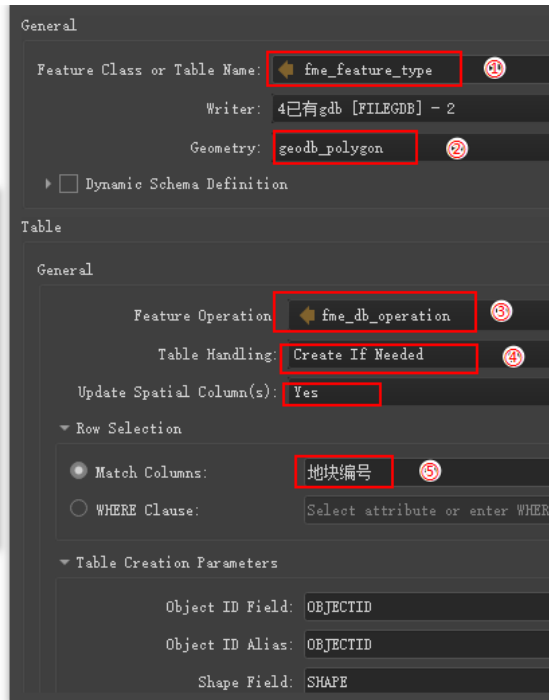


图 9 GDB 读写模块设置

4.3 CAD 读模块参数设置

CAD 读模块主要是对用户坐标系和面域的设置和暴露一些 Format Attributes，其他设置则是为了减少一些不必要的输出，不影响结果。

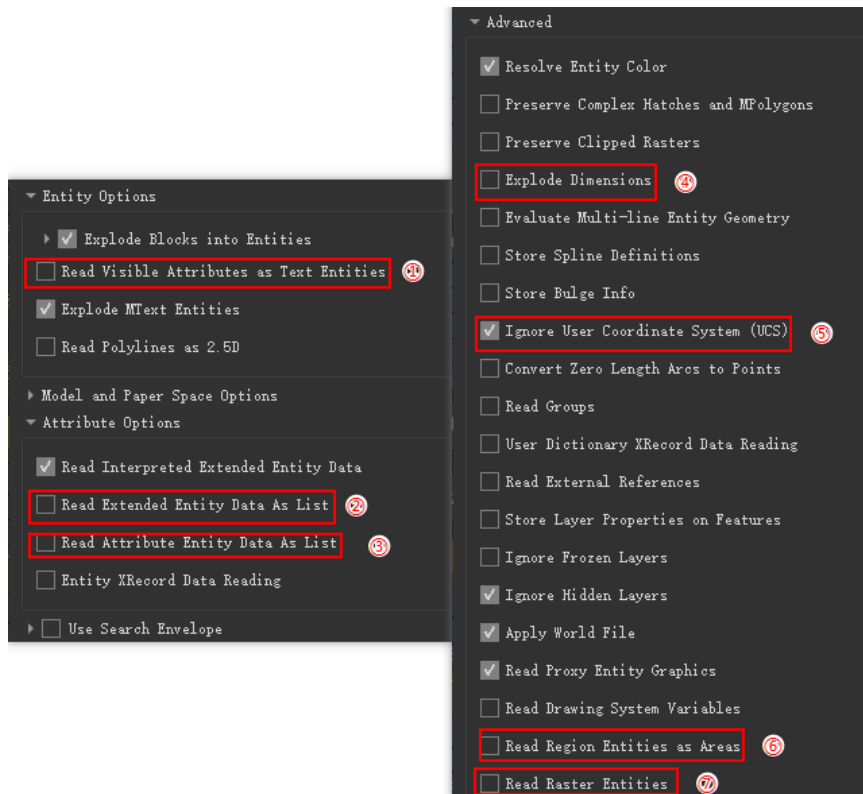


图 10 CAD 读模块设置

4.4 三角符号生成

三角符号生成的方法也有多种。最简单的，我认为是以线为轴，端点为原点分别向左右旋转（Rotator）相同角度，再连线，但形成的三角形边长会短一些，未能到达我的预期。我采用的是先在左右两侧生成平行线，然后分别取它们一侧的端点和原线段的一个端点，最后连成三角形的方式，这样使用的转换器会多一些，但更符合预期。另外，也可以考虑使用 TINGenerator 的方式，具体不详说。

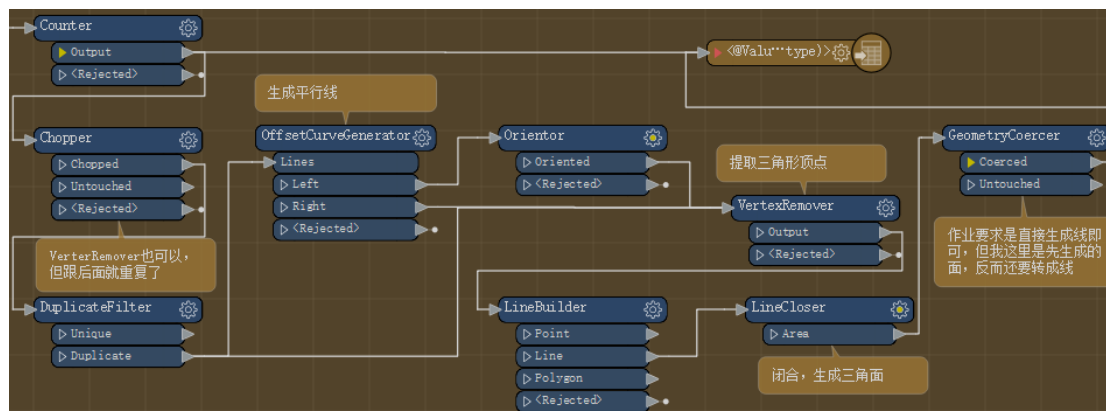


图 11 三角符号生成流程

4.5 地块面提取

这里的地块面主要有三种情况，一种是普通的多边形，一种是带文本框（带洞，有些可能需要打散）的多边形，还有一种是面域。

普通和带文本框的多边形都可以通过 DonutHoleExtractor 来进行提取和打散。

而面域的提取则分三步走：

- 1) CAD 读模块不能勾选“Read Region Entities As Areas”。
- 2) 使用 SurfaceDissolver 进行表面融合处理。
- 3) 强制转换为 2D

经过以上三步处理，就能完整地把面域轮廓提取出来。

最后，把地块面作为 SpatialFilter 的 Candidate 端输入。

4.6 地块编码关联

如前面数据分析，地块编码要分两种情况考虑。观察数据，所有落在地块外的编码都是落在指引标识里的，而指引标识是由指向符和外框组成，所以第一步是利用 LineCombiner 把其指向符和外框连接起来，再通过邻近分析把地块编码赋予指引标识。

最后把 MatchedBase 和 UnmatchedCandidate 作为 SpatialFilter 的 Filter 端输入。



图 12 地块编码邻近分析

4.7 地块指标提取及关联

也是两种情况。以连接线作为媒介，进行邻近分析，其中连接线是作为 Base 端输入，目标节点和地块指标由 Candidate 端输入，然后把连接线关联的目标节点坐标替换连接线，再赋予指标表属性，最后再和 UnmatchedCandidate 一起作为 SpatialFilter 的 Filter 端输入。

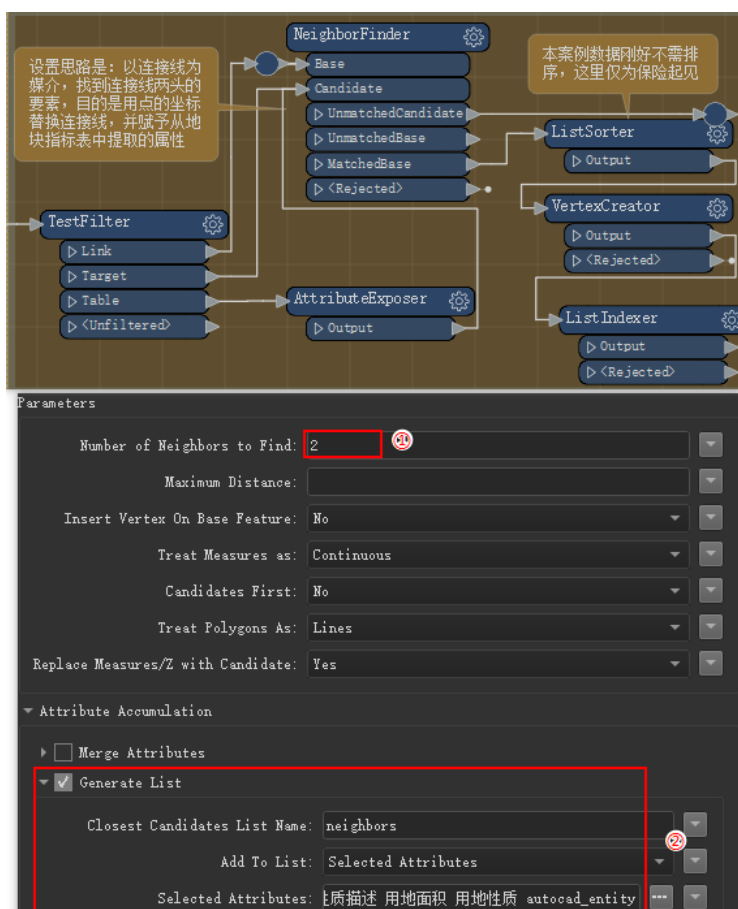


图 13 地块指标提取流程及邻近分析

4.8 空间过滤

较为简单，只需要做一个相交过滤即可，可直接使用默认参数设置。

4.9 RGB 颜色转换及空值转换

RGB 颜色转换是把各个颜色通道由[0,1]范围扩展至[0,255]，用 AttributeSplitter 分开再分别计算组合即可。

空值转换主要是针对某些字段以“-”表示空值的情况，直接输出会报错，需要提前转换为 Null 或 Missing。

5. 结语

在作业要求前提下，加入了自约束条件，最终完成了本模板。基本上把一些常用的转换器都用了个遍，对主要的流程步骤都进行了详细说明，把平时使用 FME 的一些经验都应用上了，也算是一次完整的自我检查。

运行效率上，消耗时间约 2.5 秒，峰值消耗内存约 100M，总体还是比较满意。至于运行结果，虽然个人觉得没什么问题，至于否准确还有待兴趣班来检验和公布结果。

```
385 =====
386 a地块面 90
387 d配套设施符号 11
388 f三角符号 32
389 转置结果表 7
390 =====
391 Total Features Written 140
392 =====
393 Translation was SUCCESSFUL with 0 warning(s) (140 feature(s) output)
394 FME Session Duration: 2.5 seconds. (CPU: 1.4s user, 0.7s system)
395 END - ProcessID: 24480, peak process memory usage: 104400 kB, current process memory usage: 84576 kB
396 Translation was SUCCESSFUL
```

图 14 模板运行日志

最后，感谢这期模板兴趣班，给了我本文思路。也希望本文对大家使用 FME 有所帮助。