

# 使用 FME 实现工业保护线入库

东莞市城建规划设计院 曾星

## 一、需求分析

### 1、工业保护线简述

为贯彻落实《广东省降低制造业企业成本支持实体经济发展的若干政策措施》文件精神，支持实体经济发展，保障产业用地发展空间，东莞市政府划定严格控制和保护的工业用地范围线。划定工业保护线分为两类：工业红线和工业蓝线。工业红线是为保障城市产业长远发展而确定的，符合城市规划的工业用地保护线；工业蓝线是为了稳定城市一定时期内工业用地总规模，将已改变规划用途的现状工业用地择优划定的保护线。同时为工业保护线划定一事一议项目的更新模式以打消市场对工业用地的不合理预期。

### 2、工业保护线入库的必要性

工业保护线规模大，全市共划定 420 平方公里工业保护线，其中工业红线 372 平方公里，工业蓝线 48 平方公里；一事一议项目 142 宗，面积 16 平方公里。工业保护线承载信息多，每宗工业保护线的属性数据应包含：图斑编号、所属镇街、所属类型、面积、基期工业用地比例、工业用地比例、新型产业用地比例、更新类型、更新时间、更新次数等。一事一议项目属性数据应包含：属地镇人民政府（街道办事处、园区管委会）承诺的改造模式及其他改造要求。东莞增量用地少，存量用地调整次数多，现状工业基数大，市场对工业用地存有不合理幻想等因素，潜在调整、管理问题较多。

为规范和细化全市工业保护线的管理、维护及调整工作，需对工业保护线进行标准化入库，建立工业保护线信息化数据库并进行动态跟踪与维护。

## 二、技术路线

### 1、技术需求

为方便管理和维护工业保护线数据库，库内需生成三类要素文件：工业

保护线（实时库）、一事一议项目、工业保护线（历史库）。

考虑到工业保护线调整历次范围会具备不确定性，利用空间的叠加来直接更新难以满足要求，需要设置图斑 ID、设置插入/修改/删除标识，利用 FME 的格式属性（fme\_db\_operation）来实现。

为避免入库人员的误操作，整体流程需增加重复录入流程终止，录入 SDE 数据库后支持回退操作。

规划设计人员习惯使用 CAD 制图，而普通 CAD 不具备扩展属性，因此只能使用传统方式规定 CAD 样式：图斑填充（存在镂空，填充更容易识别）+ 关键属性（图斑编号，唯一标识码；更新类型，方便 fme\_db\_operation 设置插入、修改、删除）。具体样式参见图 1。其他属性则使用 EXCEL 记录（所属镇街、所属类型、基期工业用地比例等），再基于图斑编号进行图属挂接。



图 1 工业保护线入库前 CAD 样式

## 2、技术路线

基于 CAD 成果样式，首先需对 CAD 填充和文本标注进行空间链接，因为文本标注是多行文本，包含 2 个属性，需对文本标注进行属性整理，整理好的属性就可以结合已入库数据情况对即将入库数据进行是否重复录入或者回退操作判断，以满足规避误操作需求。

在入库过程中，操作人员难免会存在不经意间重复录入情况，而自己并没有发现及时进行回退操作的话，数据库内相同要素会同时存在 2 份。为了避

免此种情况发生，在流程中增加此步骤让机器帮助自动终止，因为回退操作和误入都会存在相同文件输入的过程，故将判断重复录入放在判断是否需要回退之后以避免回退也被终止。

最后，就可以与 EXCEL 进行属性挂接，计算更新次数及更新时间并完成入库。

整个入库流程的具体思路如图 2 所示，实现的关键是利用 FME 的 FeatureReader 获取历史库、实时库内情况进行比对，并根据相应需求提取历史库中要素并设置 fme\_db\_operation 字段以完成插入、修改、删除操作达到维护数据库的效果。

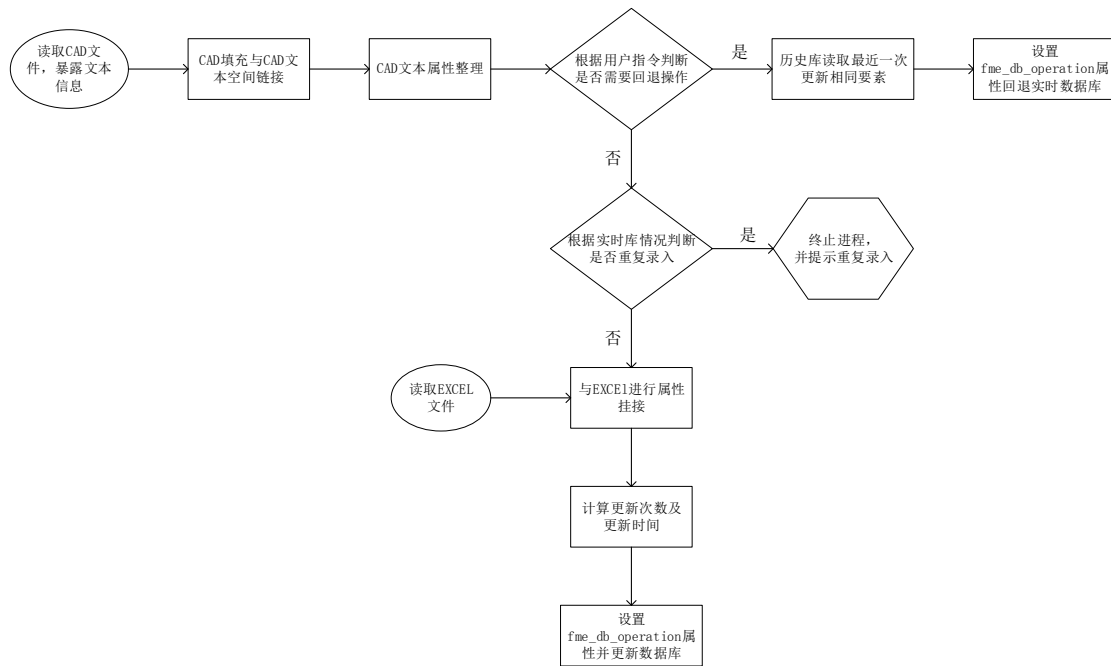


图 2 整体思路流程

### 三、具体流程

#### 1、CAD 填充与 CAD 文本空间链接

在 CAD 中，虽然文本标注在 CAD 填充之上，但文本标注的文本框中心点可能在相邻填充之上，而 FME 识别文本标注的填充为其文本框中心点。可能存在如图 3 所示情况：斑块 84-L-004 的填充上无文本压盖，而斑块 84-H-015 的填充上有 2 处文本压盖。

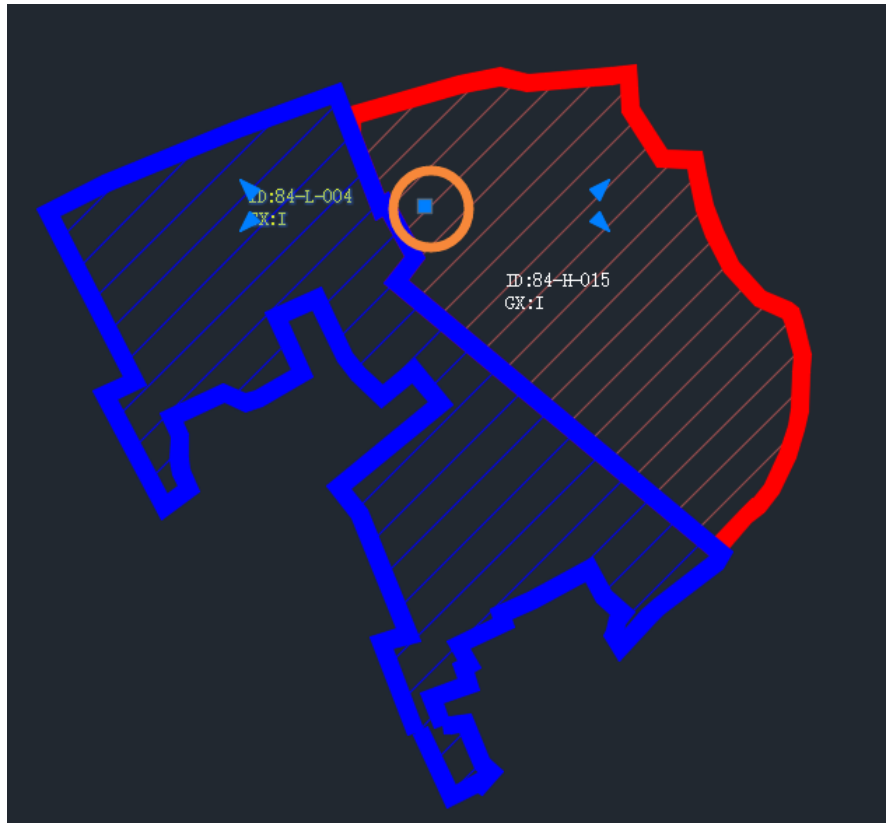


图 3 CAD 空间链接存在问题

解决思路为先对填充进行唯一码标识，以文本标注为参考寻找最近的填充，筛选出无异议的进入下一流程，对有冲突的进行空间链接 SpatialFilter。具体流程如图 4 所示。

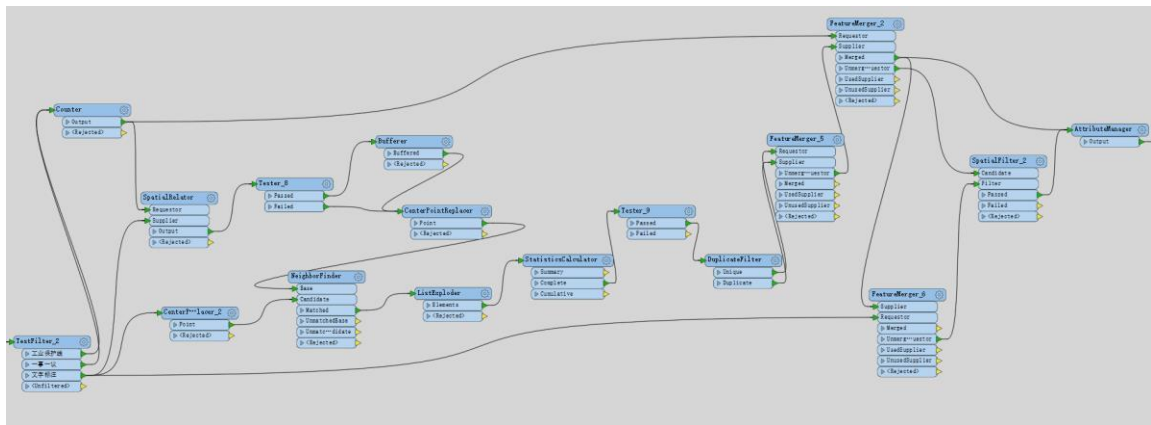


图 4 CAD 空间链接具体流程

## 2、CAD 文本属性整理

由于输入文本标注为多文本形式，整理文本数据时要注意读模块时要把实体选项中打散多文本实体勾选取消，不取消的话文本将被打散，而且读取

顺序会变成随机，text 扩展属性列表会存在有些 ID 在前更新类型在后，有些 ID 在后更新类型在前情况，不利于后期整理属性。

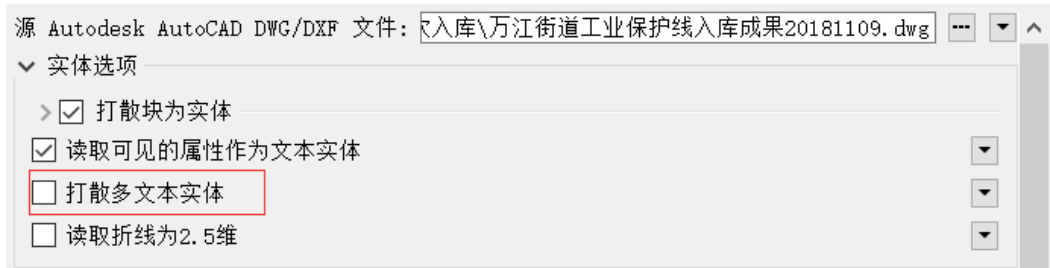


图 5 CAD 读模块设置注意取消勾选打散多文本实体

设置好读模块后只需要使用 AttributeSplitter 转换器依次对回车符、对冒号进行分割，再利用 AttributeCreator 新建对应属性并赋值即可，具体流程如图 6 所示。

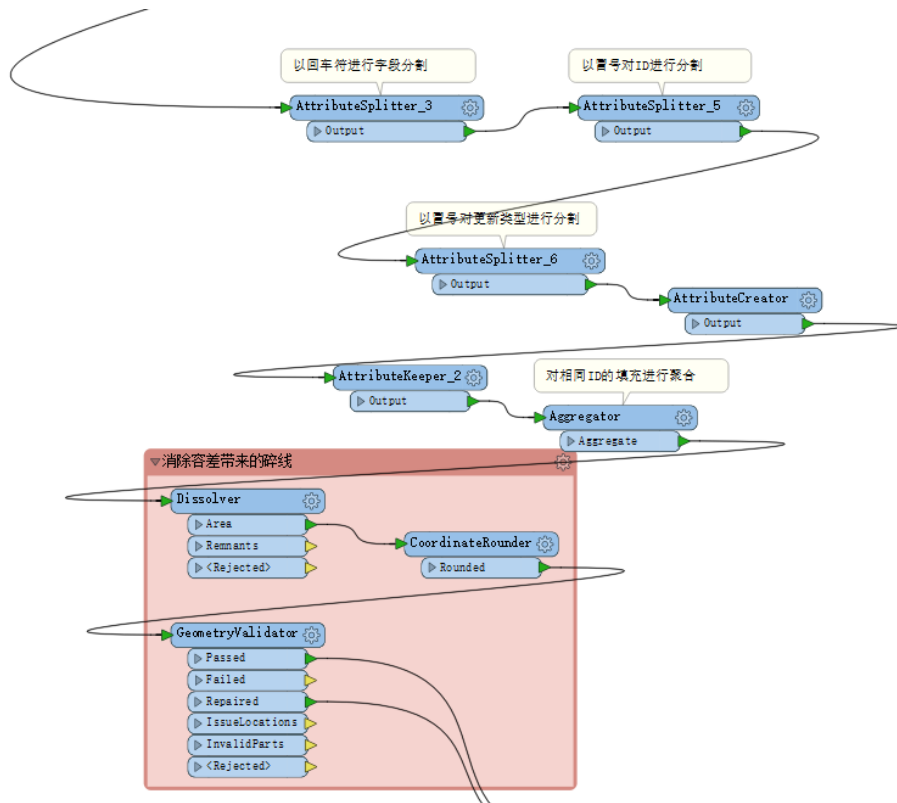


图 6 CAD 文本属性整理及图斑聚合清理

### 3、根据用户属性判断是否需要回退

判断是否回退主要利用 Tester 转换器，如果是的话执行回退操作。具体操作流程需结合 fme\_db\_operation 属性使用，在第 7 小点再详述，本节重点解决用户参数问题。

为了让用户在意识到录入错误数据到数据库后能够回退，需要设置用户参数给用户选择，用户参数类型为选择，可选配置为是%否，默认可设置为否，因为毕竟回退是少部分有意识的操作，要注意勾选发布，如此用户在运行转换器是可以调整该参数。Tester 转换器再根据此判断是否回退用户参数将处理情况进行分流。

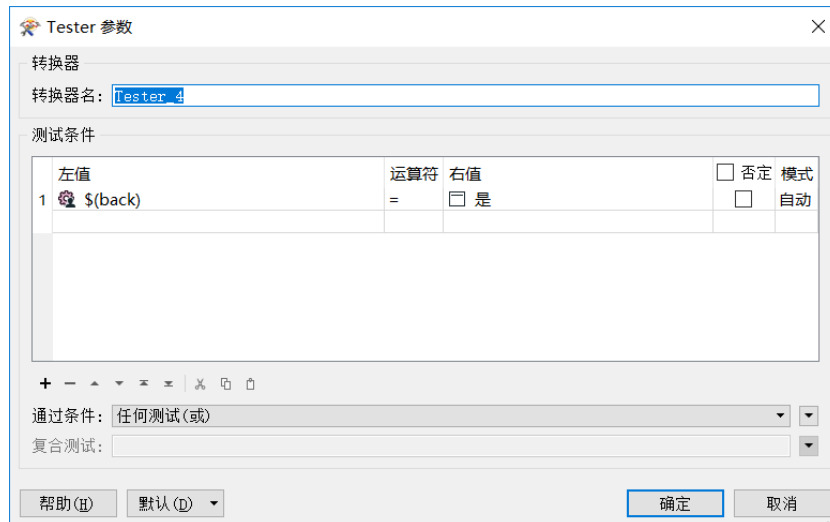


图 7 Tester 转换器设置

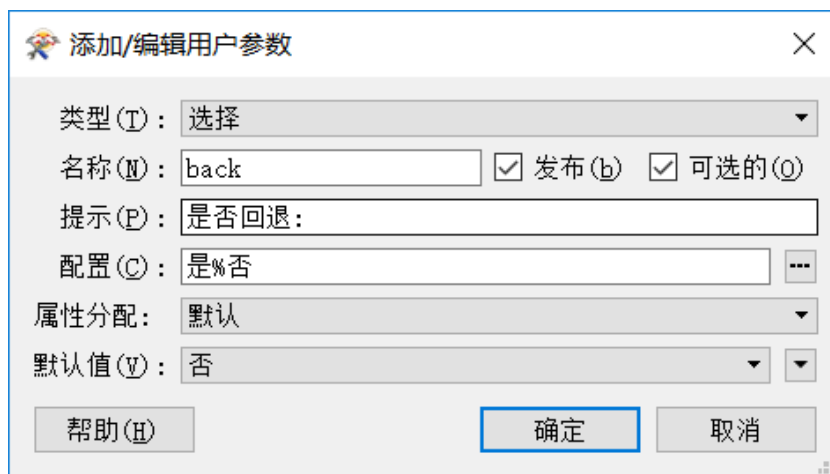


图 8 用户参数 back 设置

#### 4、判断是否重复录入

判断是否重复录入主要利用 FeatureReader 转换器，通过读取 SDE 中工业保护线和一事一议要素（实时库），提取 ID 和更新类型与 CAD 录入信息一直要素，并用 StatisticsCalculator 转换器进行总数统计。当总数大于 0 说明存在重复录入情况，再利用 Tester 和 Terminator 对此种情况进行终止

并抛出终止提示。



图 9 FeatureReader 转换器设置

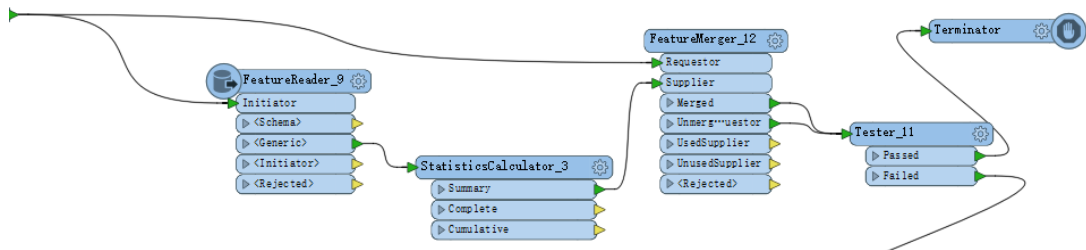


图 10 判断是否重复录入具体流程

## 5、CAD 与 excel 挂接

在 CAD 与 excel 挂接之前要先判断更新字段是否为 D (删除), 因为需要删除的图斑不再有对应的属性, 不需要进行挂接。另外, excel 中容易存在

前后空格，为了便于顺利与 CAD 图斑的 ID 顺利挂接，可以使用 AttributeTrimmer 清除前后空格再进行 FeatureMerger。

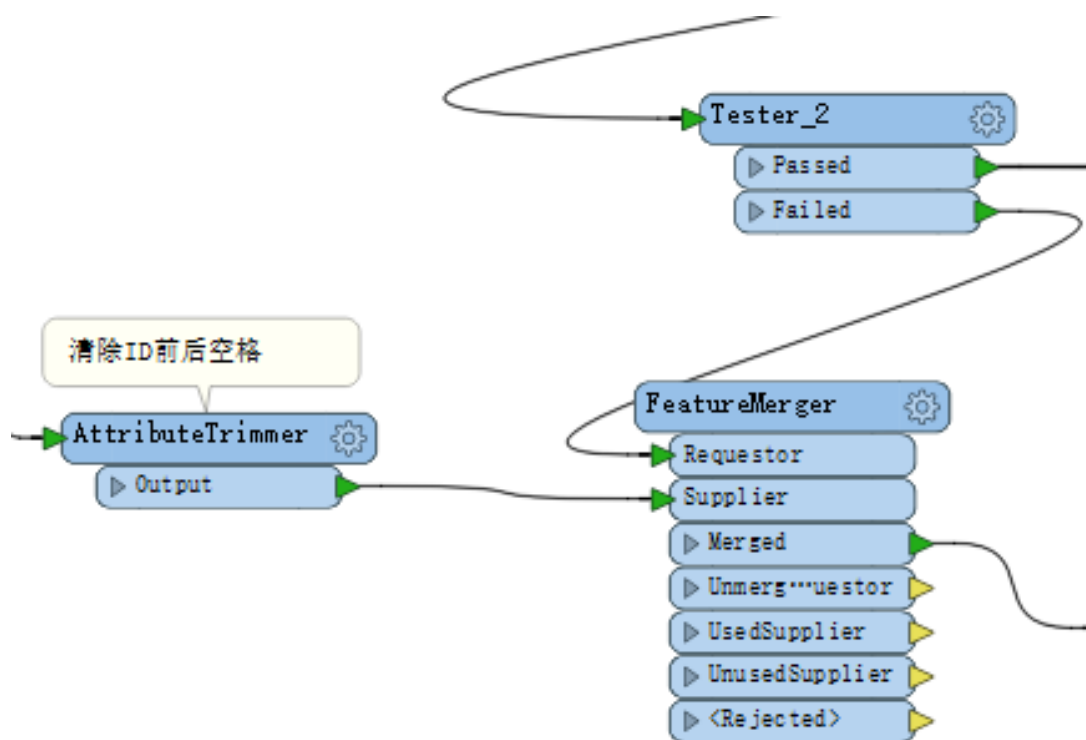


图 11 CAD 与 excel 挂接具体流程

## 6、计算更新次数及更新时间

为了方便数据库管理，实时库内数据需计算更新次数及更新时间。由于 CAD 与 excel 挂接流程已经将删除情形分流出去，目前数据流中仅包含 I(插入)和 UN(更新)两种情形。

其中，I 为首次录入数据库，更新次数直接赋值为 0 即可，更新时间直接利用 FME 文本编辑器内的时间函数@DateTimeNow()即可计算出当前时间；UN 为图斑更新，目前实时库内存在的相同图斑即为上一次更新次数，只需在此基础上+1 即可，而调用实时库内相应图斑也是利用 FeatureReader 转换器筛选图斑编号一致要素，要注意暴露更新次数属性，再利用 FeatureMerger 转换器连接回当前更新要素。最后再使用 DateTimeConverter 转换器规范化时间格式。



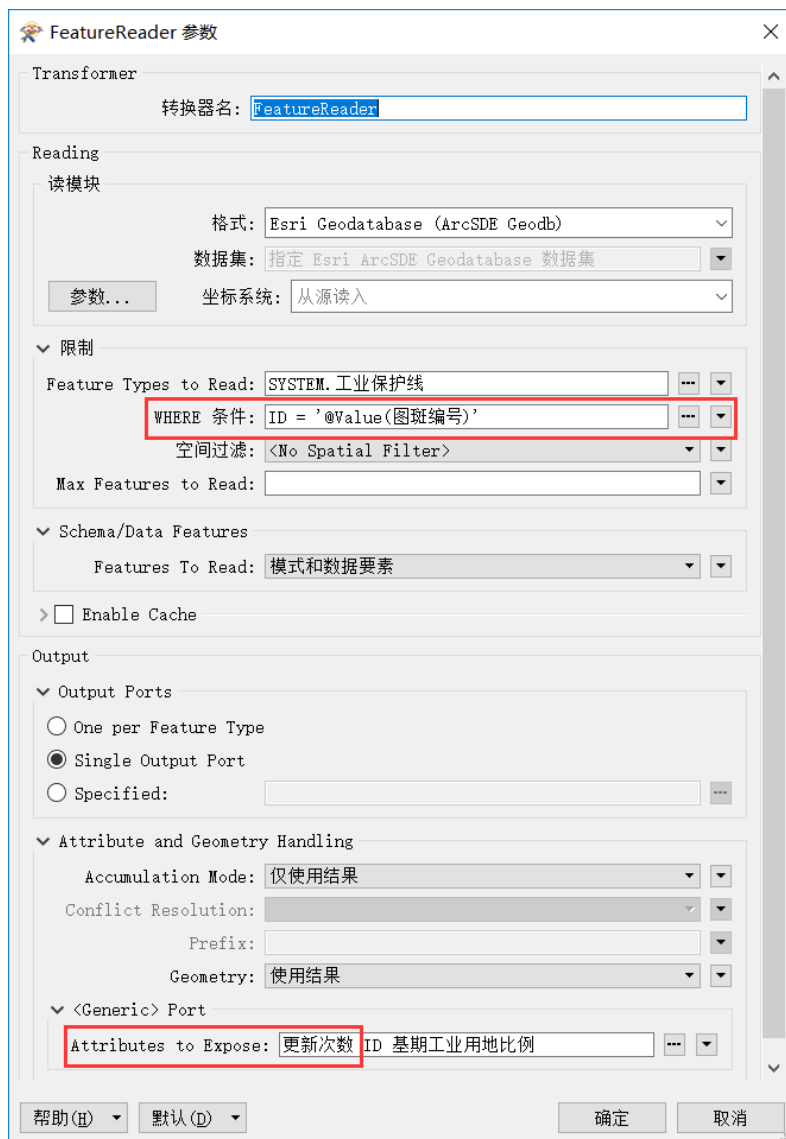


图 12 FeatureReader 转换器设置

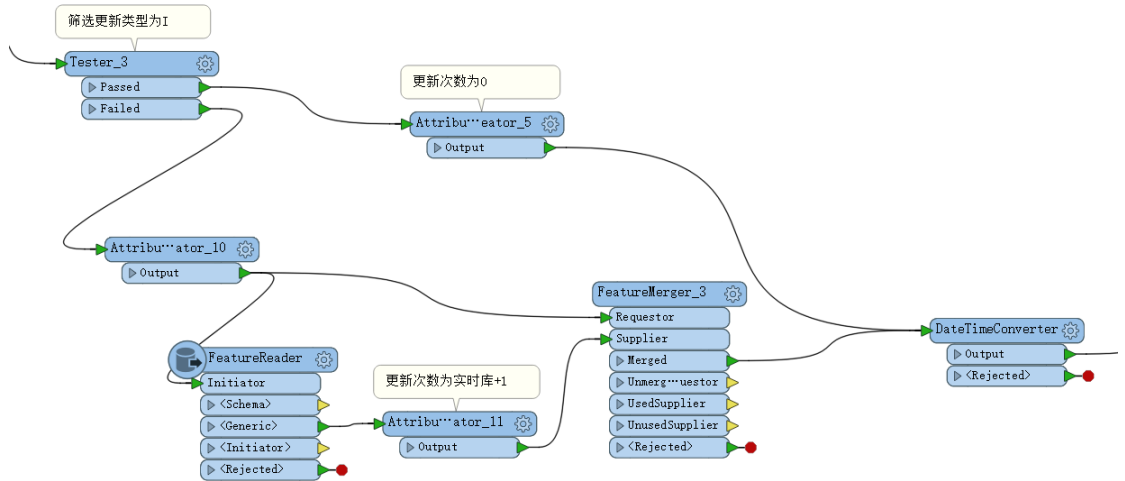


图 13 计算更新次数及更新时间具体流程

## 7、利用 fme\_db\_operation 属性更新数据库

### (1) 正常入库

#### ①实时库

只需将工业保护线和一事一议两类要素，分别基于更新类型用 TestFilter 分流，新建 fme\_db\_operation 属性赋值对应的 insert、update、delete，同时写模块注意设置要素操作为 fme\_db\_operation，同时匹配列为 ID(用户属性注意暴露 ID)。

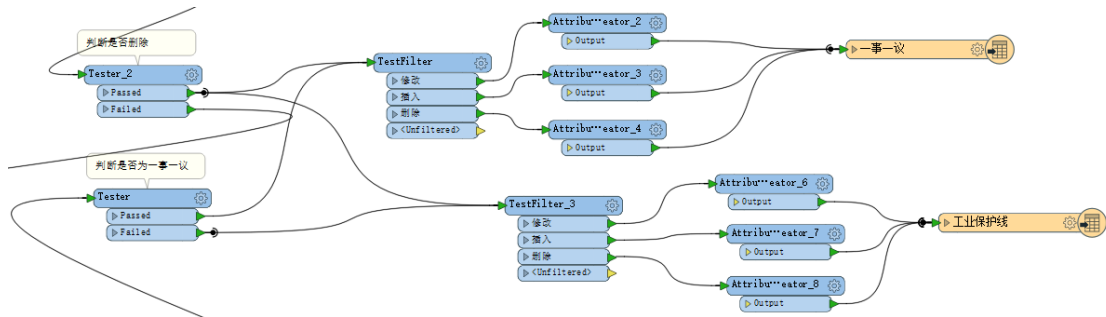


图 14 正常入库（实时库）流程

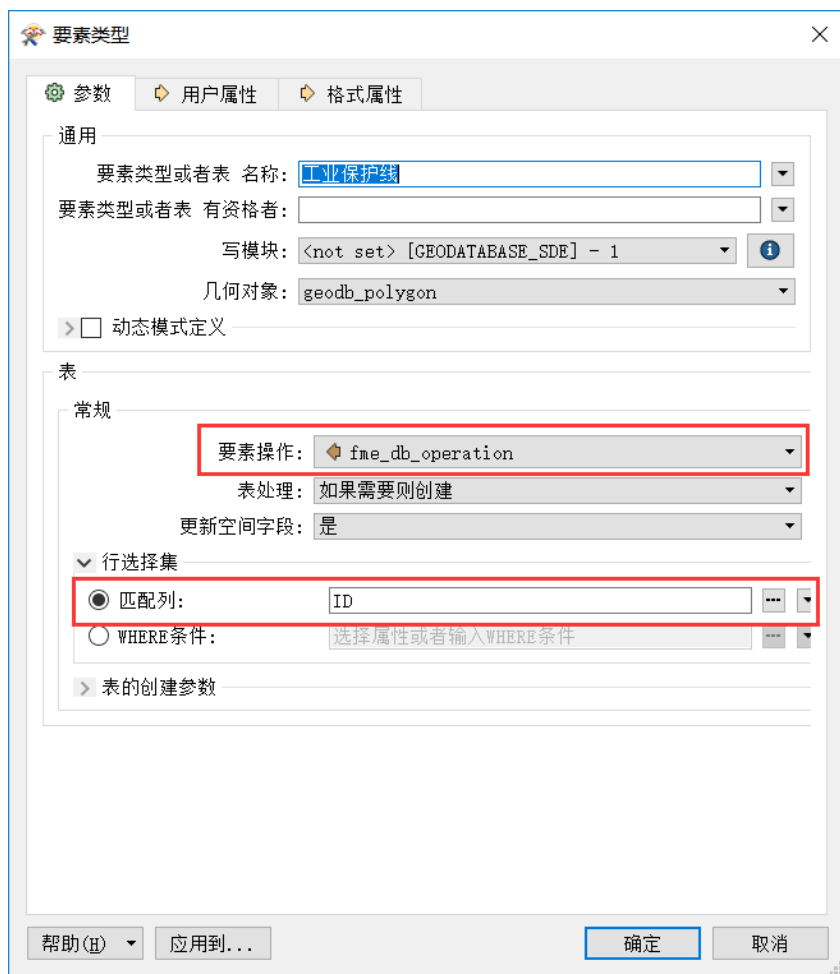


图 15 写模块设置

## ②历史库

历史库需要注意的是里面需要记录实时库每一次变动的过程。首先需要新增字段 STATTIME 和 ENDTIME, 还要格外注意即使是删除要素在历史库中也是有保留下来, 并要新增特殊标记字段[删除标记], 该字段的设置也是方便在回退操作中能找回被删除的要素。

对于插入要素, 入历史库 fme\_db\_operation 属性仍为 insert, 注意新增 STARTTIME; 对于更新要素, 历史库需包含 2 次要素调整, 1 是本次更新要素, 2 是该要素对应的上一次入库要素, 本次更新 fme\_db\_operation 属性仍为 insert, 该要素对应的上一次入库要素需利用 FeatureReader 读取出来, 方法为图斑编号一致且更新次数等于第 6 节计算的更新次数-1, 并暴露所有属性, 具体参数设置详见图 17, 然后记得计算 ENDTIME; 对于删除要素, 因为未参与 excel 属性挂接, 详细属性仍需要读会原始库, 故仍使用 FeatureReader 读取对应图斑, 但读取方法与更新要素不同, 因为删除要素没参与到计算更新次数中来, 在第 5 小节 CAD 与 excel 挂接中直接被筛选

出来参与本流程了，所以需要选择 ENDTIME 为空且图斑编号一致的要素，暴露所有属性并计算 ENDTIME，同时记得新增删除标记且赋值为“是”。

最后，写模块匹配列仅包含 ID 已经不具备唯一性了，此时需要再添加 STARTTIME。

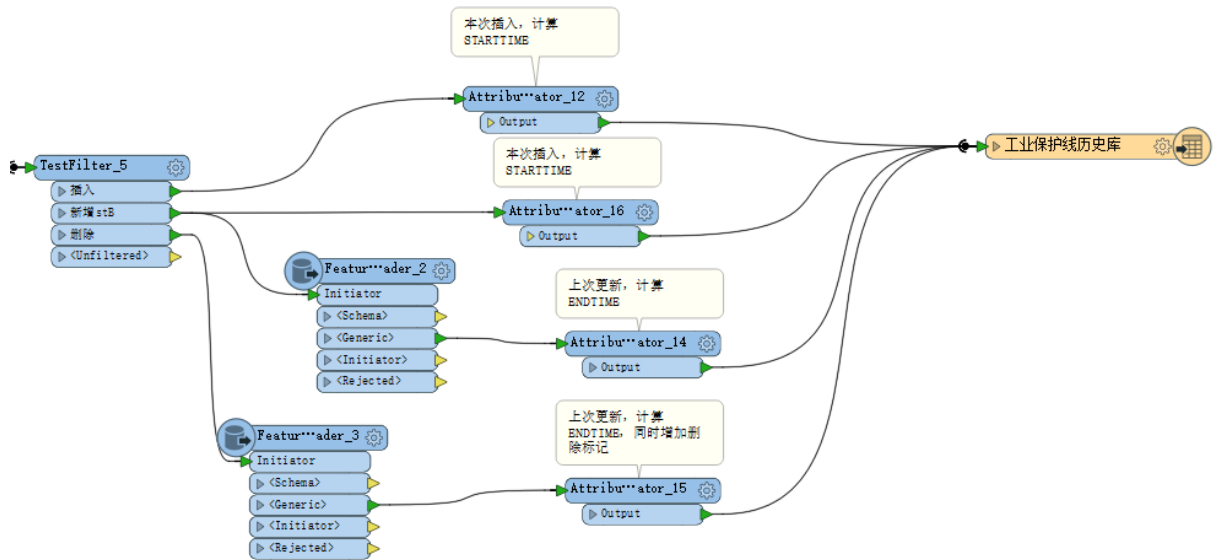


图 16 正常入库（历史库）流程



图 17 本次更新要素读取的上一次入库要素



图 18 本次删除要素读取的上一次入库要素

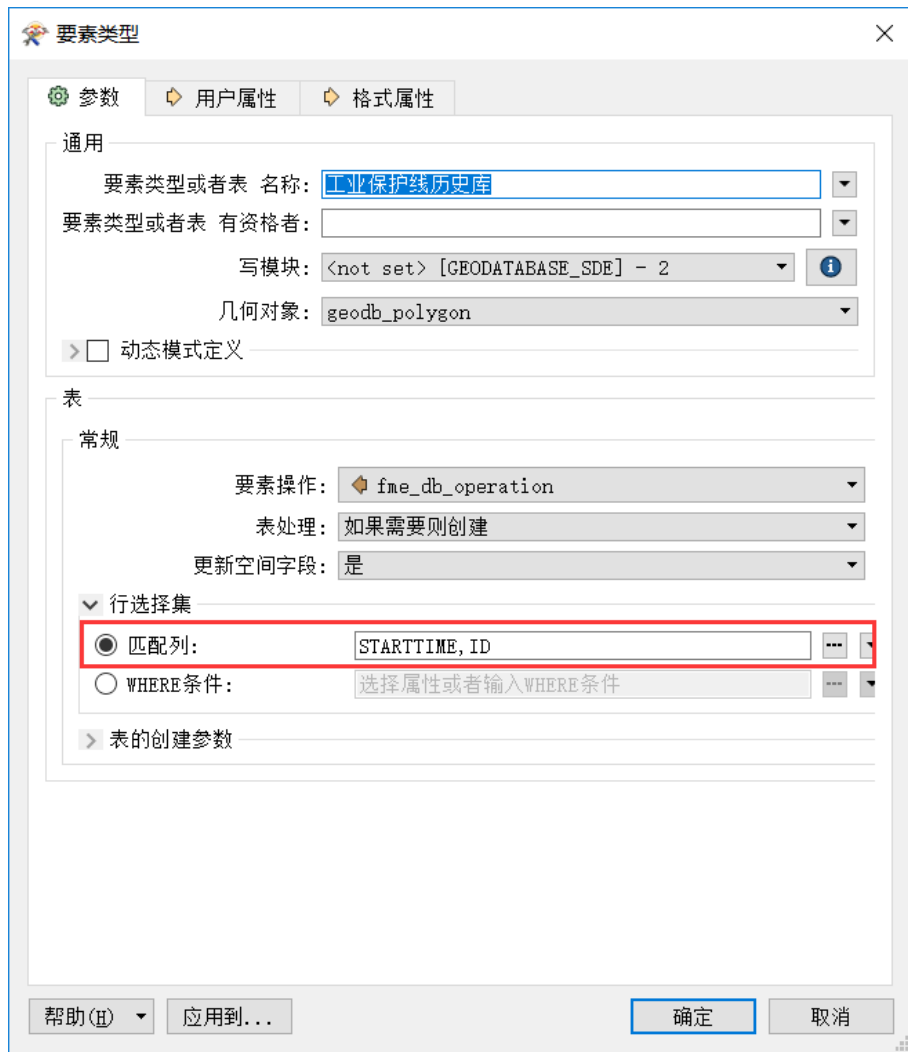


图 19 写模块设置

## (2) 回退操作

回退操作，其实只涉及新增和删除两类操作。

### ① 实时库

对于上次删除的需要补回，对于上次新增的需要删除，对于上次更新的需要删除实时库中的图斑并找回上上次的对应图斑。

基于此，可以先使用 Tester 筛选出删除的图斑，读取历史库中对应图斑补回实时库，由于在历史库入库过程中有专门对删除类要素进行标记，故 FeatureReader 转换器只需找到 ID 一致且删除标记为‘是’的图斑，具体设置参见图 21。剩余要素通过 FeatureReader 读出实时库中对应要素，要注意暴露出更新次数（图 22），因为这是判断上次操作是更新还是插入的关键字段，而这两种类型需采取不同方式回退。再次使用 Tester 筛选出更新次数为 0 的要素（上次新增要素）fme\_db\_operation 直接定义为删除即可，由于

一事一议项目仅划定工业保护线时存在，故一事一议项目的写模块回退只需连此一条数据流。此次 Tester 失败要素为上次更新的要素，需要产生 2 条数据流，1 条增加 fme\_db\_operation 赋值 delete 用来删除实时库中需要回退的要素，另 1 条用来补回这些要素，只能从历史库中找回，仍是使用 FeatureReader 读取历史库中 ID 一致且更新次数 = @Value(更新次数)-1 的要素，fme\_db\_operation 赋值 insert。完整流程如图 20 所示。

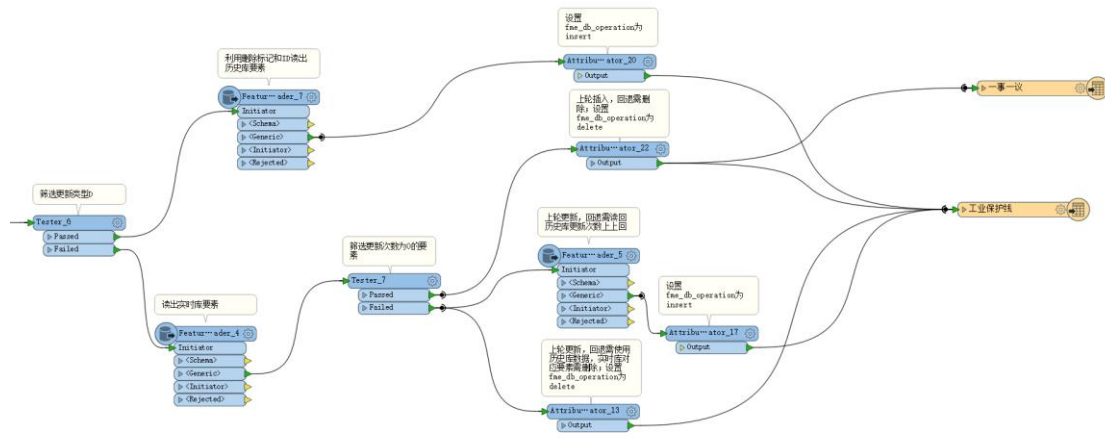


图 20 回退操作（实时库）流程





图 21 读取需回退的删除要素



图 22 读取上次操作非删除类要素 FeatureReader 转化器设置

## ②历史库

历史库回退操作流程与实时库基本一致，主要是针对实时库中 fme\_db\_operation 赋值为 insert 的要素更新并对删除标记、ENDTIME 字段进行处理。

对于上次删除的需要更新，主要操作是将删除标记、ENDTIME 清零；对

于上次新增的需要删除；对于上次更新的需要删除历史库中的图斑并将上上次的对应图斑 ENDTIME 清零。

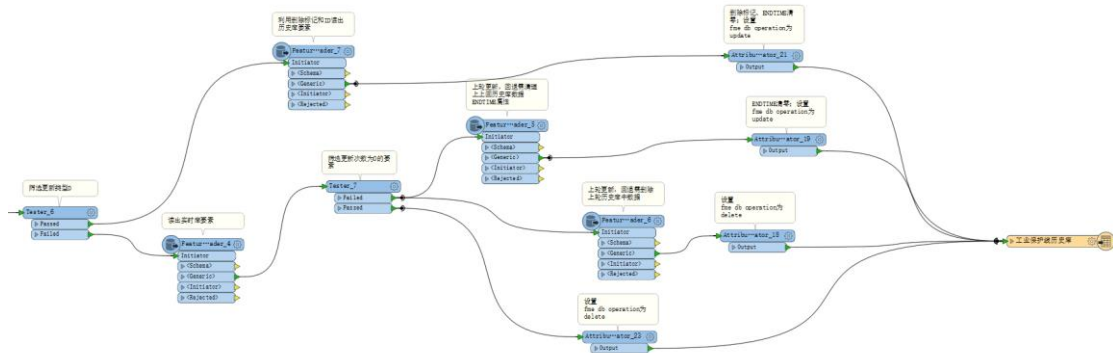


图 23 回退操作（历史库）流程

## 8、输出数据属性设置

对于输出数据库属性类型可以在用户属性处进行设置，主要更新时间、STARTTIME、ENDTIME 设置为时间格式，更新次数设置为整数型等，具体可参考图 24。尤其注意 STARTTIME 做为历史库行选择集的匹配列一定要输出规范，不然可能会导致更新出错。

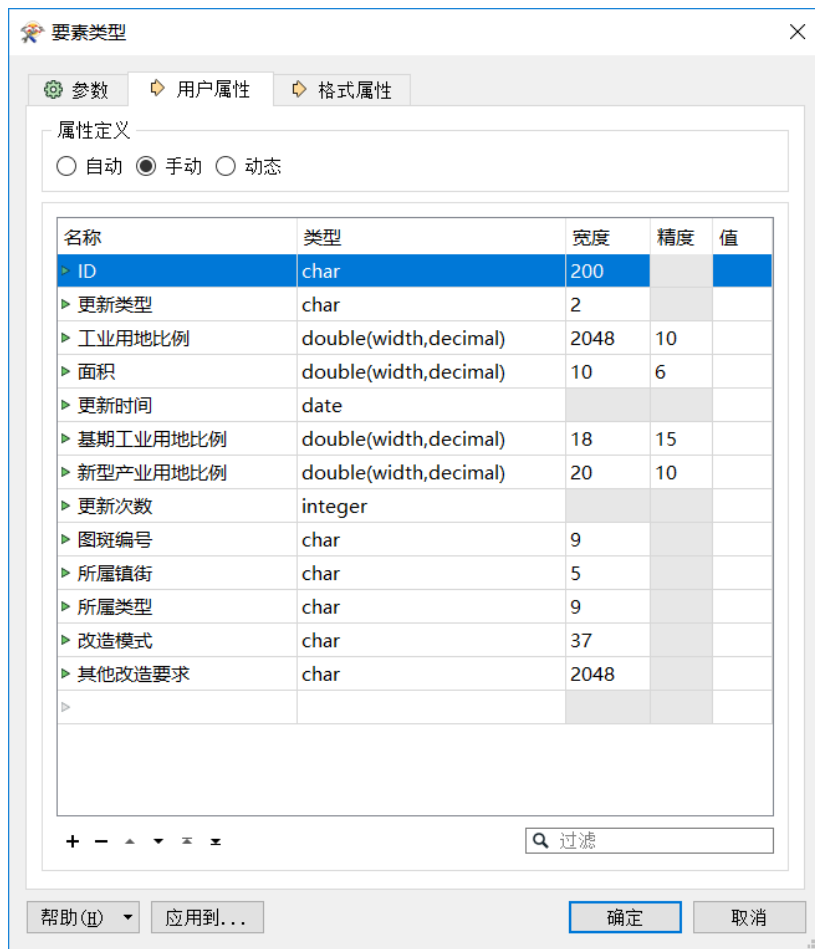


图 24 数据属性设置

#### 四、结语

在整个模板 FME 全程零代码操作，仅涉及简单的几句 SQL 语言以实现对数据库内的数据提取。可以让流程设计者倾注更多的心力关注流程本身，而非编程语言中各种对象的定义。以一种非常高效的流程可视化编程方式实现一些复杂功能。

结合 FMESERVER 的文件检测和收发邮件功能，后期还可进一步优化流程，指定特定文件夹作为数据库入库文件夹，发布 Directory Watch 协议监控文件的创建和删除，发布到特定主题，主题检测到创建文件执行入库命令，检测到删除文件使用 Tester 将判断是否回退用户参数设为是，执行回退命令。而每一次 Job Submitter 之后自动发送邮件给用户，基本实现全程自动化操作。