1 模型配置设计文档

**1.1功能描述**

针对算子、数据源、连接线的增删改操作，引发的算子配置信息及状态变化问题，设计多种情况下算子状态切换方案。

**1.2设计目的**

配置局部修改时，尽量减少客户的不必要操作；同时配置的设计也尽量简化代码实现。

**1.2多场景算子状态切换方案设计**

1. 删除数据源①、删除算子情况②或删除连接线③，如下图所示：



删除数据源：

1. 删除数据源，所有邻近第一层子节点算子状态变为Null，后续链路中子节点状态降级（Degrade），降级规则为：**Done状态变为Ready状态，其他状态保持不变。**
2. 重新拖入新数据源，算子中上次保存的数据源信息与本次拖入的数据源信息进行比较。
3. 无论数据源路径FullFilePath是否一致，当新数据源表头与旧数据源表头字段顺序一致(SequenceEqual)且新数据源表头Count>=旧数据源表头Count，邻近第一层子节点算子状态升级(Upgrade), 升级规则为：**原来为Null状态，就变回Null状态，原来为Ready或者Done状态，就变回Ready状态。**后续链路中子节点状态不做变化。
4. 若不满足条件三，后续整个链路所有算子的状态变为Null，但所有算子配置设置保留。

删除算子：

1. 删除算子，与算子相连的结果控件同时删除，右侧最近的算子状态变为Null. 后续链路中Done状态变为Ready状态。
2. 重新拖入算子，只连接左侧数据源，模型无任何影响；重新拖入算子，并配置完毕，连接右侧算子，右侧算子将上次保存的数据源信息与新生成的结果算子的数据源信息进行比较。
3. 若数据源路径一致、新数据源表头包含旧数据源表头且表头字段顺序一致，右侧最近的算子原来为Null状态，就变回Null状态，右侧最近的算子原来为Ready或者Done状态，就变回Ready状态。
4. 若数据源路径不一致或新数据源表头不完全包含旧数据源表头或表头字段顺序一致，后续整个链路所有算子的状态变为Null。

删除连接线：

1. 删除连接线，右侧最近的算子变为Null状态
2. 重新连接线，右侧最近的算子原来为Null状态，就变回Null状态，右侧最近的算子原来为Ready或者Done状态，就变回Ready状态。
3. 连线：

进行环路检测，不允许模型中环路的存在。

1. 某个算子配置信息修改，如下图所示：



输出字段修改：

1. 修改的配置输出字段名称一致且选择的索引先后位置不变（例如，旧输出选择的字段索引为2、3，新选择的字段索引为1、2、3，这样认为索引先后位置改变；若新选择字段索引为2、3、4，则认为索引位置不变），后续所有算子配置状态不变化，否则后续算子状态变为Null状态，与其连接的结果控件对应的BCP文件内容清空，并写入新的表头，仅刷新与其相连的结果算子表头
2. 否则后续所有算子配置状态变为Null状态，与其连接的结果控件对应的BCP文件内容清空，并写入新的表头。
3. 若打开ⅱ中结果控件相连接的算子，自动加载算子上次配置的索引对应的新bcp字段数据，若索引值在新BCP文件中不存在则不加载该信息。
4. 步骤ⅲ加载完毕，用户点击取消，用户操作不保存，模型无任何影响；用户操作完点击添加，该算子状态变为Ready，此外本次输出字段内容仍与上次配置中保存的输出字段内容进行比较，重复步骤ⅰ、ⅱ。



仅配置条件修改：

仅配置条件修改，配置输出字段不修改，无任何影响。

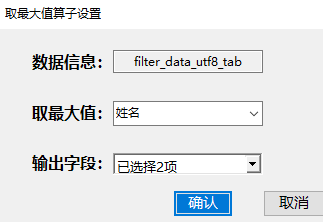
2配置信息持久化

**2.1配置信息保存**

配置信息内存存储格式：

算子配置信息保存在算子的OptionDict字典中，字典键为配置窗口选项名称，值为配置窗口的用户输入结果。对于配置窗口中控件有下拉列表的字段选项保存问题，在配置信息持久化中只保持字段的索引，以下进行详细说明：

最大值算子：



取最大值算子的“数据信息”、“取最大值”、“输出字段”信息保存在配置字典，存储格式：

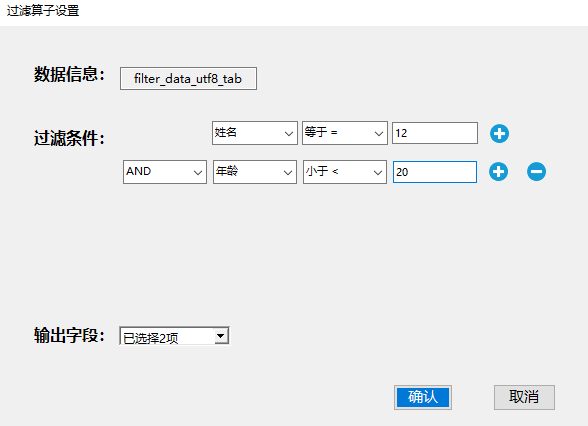
OptionDict ["columnname0"]=”姓名 年龄 手机号码 身份证”

OptionDict ["maxfield "]=”0”

OptionDict ["outfield "]=”1,3”

其中”columnname0”保存配置窗口“数据信息”的表头，”Maxfield”是配置窗口“取最大值”选择的字段“姓名”的索引“0”，”outfield”是配置窗口“输出字段”选择的字段的索引。其中，最大值算子为单输入算子，如果为双输入算子，会有两个数据源，则”columnname0”存储左表数据源表头字段，”columnname1”存储右表数据源表头字段。

过滤算子：



过滤算子的数据信息、过滤条件、输出字段信息保存在配置字典，存储格式：

OptionDict ["columnname0"]=”姓名 年龄 手机号码 身份证”

OptionDict ["factor1"]=”0,2,12”

OptionDict ["factor2"]=”1,1,3,20”

OptionDict ["outfield "]=”2,3”

其中“factor1”存放第一行过滤条件，“0，2，12”分别对应“姓名”的索引，“等于=”的索引，输入的过滤条件”12”。

配置信息Xml存储格式：

1. 配置信息属于算子的一部分，因而在Xml文档中将配置信息作为一个子节点<option>保存在算子节点< ModelElement >下面，以最大值算子的配置信息持久化为例，说明配置信息xml文件存储格式：

< ModelElement >

<type>Relation</type>

<start>元素标号</ start >

< end >元素标号</ end >

<end\_pin>pin序号<end\_pin>

< start \_location>{1,2}</ start \_location>

< end \_location>{3,4}</end\_location>

<option>

<columnname0>年龄 访问次数</columnname0>

<maxfield>1<\maxfield>

<outfield>1,2<outfield>

<option>

</ ModelElement >

其中 <columnname0>节点存储数据源表头，表头字段用“\t”分割， <maxfield>节点保存最大值字段的索引，<outfield>节点保存输出字段的索引。

**2.2配置信息加载**

用户再次登录，从模型Xml文件中读取算子option节点内容，将节点的属性和属性值分别作为字典的键和值赋值到对应算子的Option类的OptionDict字典中，以2.1最大值算子的加载为例：

<option>

<columnname0>年龄 访问次数</columnname0>

<maxfield>1<\maxfield>

<outfield>1,2<outfield>

<option>

读取最大值算子Xml中配置信息，并转化为字典OptionDict{“columnname0”:”年龄 访问次数”,” maxfield” : ”1”,”outfield”:”1,2” }形式存到算子的配置属性中。