##### 循环组件设计

一、For循环

1、概述

For循环是针对某一个变量的初值进行不断的改变，同过相同组件对某一输入值进行计算得出不同结果的过程，通过设置该值的初值和终值来获得计算区间，然后再对步长值进行设置，得到一系列的输入数据进行计算。

2、循环建模参数定义



1. 参数：参数是指定该循环内的一个组件的输入参数进行循环计算
2. 初值：初值就是刚才所指定的参数的初始值，这个值可以是一个常量，也可以是一个变量，但是变量必须是循环以外的某一个组件的输出值。
3. 终值：终值就是所设定的参数值的最大范围区间，这个值可以是一个常量，也可以是一个变量，如果是一个变量的话必须是循环以外的殁偶一个组件的输出值。
4. 步长： 步长是指在区间范围内每一次执行的值的增量，这个值可以是一个常量，也可以只之前某组件执行的输出值。

1. for循环xpdl定义

3.1数据扩展定义

for...step循环需要四个固定的参数，就是需要关联的具体参数，初值，终值和步长值，需要在xpdl中的<DataFields></DataFields>中声明这四个参数，具体如下：

<DataField Id="f652766" IsArray="FALSE" Name="d7"><DataType><BasicType Type="INTEGER"/></DataType>

<ExtendedAttributes>

<ExtendedAttribute Name="dataType" Value="101"/><!-- 新增扩展属性，101for step循环参数-->

<ExtendedAttribute Name="startDataId" Value="f657233"/><!-- 新增扩展属性，标识参数和初值终值步长的关系-->

<ExtendedAttribute Name="endDataId" Value="f800405"/>

<ExtendedAttribute Name="stepDataId" Value="f806622"/>

</ExtendedAttributes>

</DataField>

<DataField Id="f657233" IsArray="FALSE" Name="d8"><DataType><BasicType Type="INTEGER"/></DataType>

<InitialValue>1</InitialValue>

<ExtendedAttributes>

<ExtendedAttribute Name="dataType" Value="102"/><!-- 新增扩展属性，102for step循环参数，初值-->

</ExtendedAttributes>

</DataField>

<DataField Id="f800405" IsArray="FALSE" Name="d9"><DataType><BasicType Type="INTEGER"/></DataType>

<InitialValue>10</InitialValue>

<ExtendedAttributes>

<ExtendedAttribute Name="dataType" Value="103"/><!-- 新增扩展属性，103for step循环参数，终值-->

</ExtendedAttributes>

</DataField>

<DataField Id="f806622" IsArray="FALSE" Name="d10"><DataType><BasicType Type="INTEGER"/></DataType>

<InitialValue>2</InitialValue>

<ExtendedAttributes>

<ExtendedAttribute Name="dataType" Value="104"/><!-- 新增扩展属性，104for step循环参数，步长-->

</ExtendedAttributes>

</DataField>

参数d7是关联参数，扩展属性中标识的dataType为参数类型，101为关联参数，并且他的扩展属性还有三个值，分别是初值，终值和步长，方便在使用的时候进行关联。

参数d8是初值也有相对的扩展属性，其中102为初值类型。

参数d9是终值，103为终值的扩展属性。

参数d10是步长值，104是步长值的扩展属性

3.2转移线扩展定义

循环节点的次数和条件判定放在<Transition></Transition>标签内，在<Transition>内定义了循环的具体属性，在扩展属性内按照自定义的方式定义，具体内容如下：

<ExtendedAttributes>

<ExtendedAttribute Name="StartOfWorkflow" Value="default\_participant\_id;F409315;90;102;NOROUTING;STARTF409315"/>

<ExtendedAttribute Name="EndOfWorkflow" Value="default\_participant\_id;F409315;194;102;NOROUTING;ENDNDF409315"/>

<ExtendedAttribute Name="processType" Value="static"/>

<ExtendedAttribute Name="SubType" Value="11"/><!--新增流程扩展属性，流程类型-->

<ExtendedAttribute Name="VariableToProcess\_INOUT" Value="f652766"/><!--新增流程扩展属性，流程参数，参数7-->

<ExtendedAttribute Name="VariableToProcess\_START" Value="f657233"/><!--新增流程扩展属性，流程参数，参数8-->

<ExtendedAttribute Name="VariableToProcess\_END" Value="f800405"/><!--新增流程扩展属性，流程参数，参数9-->

<ExtendedAttribute Name="VariableToProcess\_STEP" Value="f806622"/><!--新增流程扩展属性，流程参数，参数10-->

<ExtendedAttribute Name="robotRunMode" Value="1"/>

</ExtendedAttributes>

其中VariableToProcess\_INOUT参数为关联参数，VariableToProcess\_START为开始参数，VariableToProcess\_END为结束参数，VariableToProcess\_STEP为步长值，在引擎运转的过程中需要获得这VariableToProcess\_INOUT参数关联组件。

循环新增参数如下

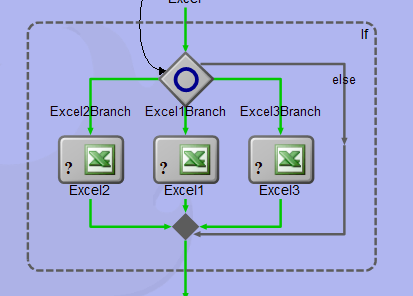
|  |  |  |
| --- | --- | --- |
| 扩展属性 | 名称 | 说明 |
| SubType | 循环类型 | 0：普通子流程  11：for...step  12：for...array  14: while  15: do...while  21：if  22：if...else |
| loopType | 循环参数类型 | 101:for...step关联参数  101：for...step初值  102：for...step终值  103：for...step步长值 |
| VariableToProcess\_INOUT | 关联参数 |  |
| VariableToProcess\_START | 初值 |  |
| VariableToProcess\_END | 终值 |  |
| VariableToProcess\_STEP | 步长 |  |
| maxRunNum | 最大执行次数 | 必须是整形数字 |

##### IF…ELSE组件设计

1. 概述

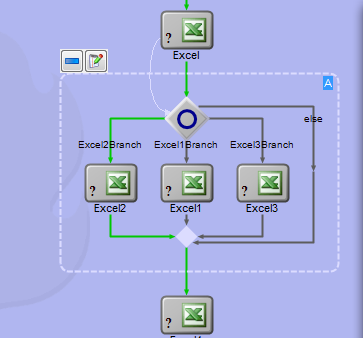
If…Else活动用条件表达式来控制工作流中流程的运行。工作流将根据条件表达式的结果来决定执行条件分支（IfElseBranch）中的哪一个活动。

If…Else分支分为两种执行方式，第一种执行方式是每一个分支都进行条件判断，而且每一个符合条件的分支都将被顺序执行，如果每一个条件都不符合则执行ELSE分支的组件，如下图所示：



（符合判断条件的每一个分支都进行执行的情况if… if…else）

另外一种情况是顺序判断每一个分支的条件，当遇到符合条件的分支时，执行该分支的组件，与上面一种状况不同的是，在执行完当前的组件以后，如果后面还有后续的分支节点则不再执行后面的节点，并且退出当前的If…Else执行后续任务节点，具体执行图例如下：



（符合条件的执行完退出整个分支if…else if…else）

1. 分支条件执行说明

|  |  |  |
| --- | --- | --- |
|  | if… if…else | if…else if…else |
| 输入条件 | 表达式、常量 | 表达式、常量 |
| 表达式判断条件 | >,<,=,!= | >,<,=,!= |
| 输出条件 | 表达式、常量 | 表达式、常量 |
| 执行逻辑 | 符合条件继续执行 | 符合条件跳出分支 |

1. 分支条件执行说明

根据两种分支执行特点不同，执行逻辑和xpdl描述也有不同的地方，首先把分支条件作为子流程处理，具体标识如下：

<WorkflowProcess AccessLevel="PUBLIC" Id="subprocess148328" Name="子2">

每个子流程标识有自己的type，type标识在子流程的扩展属性中，

<ExtendedAttributes> </ExtendedAttributes>

判定条件如下：

加在当前转移线的扩展属性内

<ExtendedAttribute Name="branchCondition" Value="![CDATA[(f652211.equals("1"))]]"/>

具体设置如下：

|  |  |
| --- | --- |
| 名称 | Xpdl扩展属性描述 |
| 普通子流程（subflow） | <ExtendedAttribute Name="SubType" Value="0"/> |
| 循环（for） | <ExtendedAttribute Name="SubType" Value="1"/> |
| 分支（if） | <ExtendedAttribute Name="SubType" Value="2"/> |

所有的组件都是先判断执行条件，如果符合条件发起相应的运行实例，如果不符合则进行下一个运行实例节点的筛选

* if… if…else

if… if…else在执行时需要将每个组件节点都设置为开始节点，在对结束节点的处理上，将else条件的节点设置为结束节点，在整个分支执行中，在执行else之前判断当前子流程是否有组件实例，如果有组件实例则不执行else分支，直接提交当前分支进行后续组件的操作，如果没有组件实例则执行else分支，然后再提交整个分支，具体分支xpdl扩展属性事例如下：

<ExtendedAttributes>

<ExtendedAttribute Name="SubType" Value="2"/>

<ExtendedAttribute Name="StartOfWorkflow" Value="default\_participant\_id;F140203; NOROUTING;STARTF140203"/>

<ExtendedAttribute Name="StartOfWorkflow" Value="default\_participant\_id;NOROUTING;STARTF142437"/>

<ExtendedAttribute Name="EndOfWorkflow" Value="default\_participant\_id;F142437; NOROUTING;ENDNDF142437"/>

<ExtendedAttribute Name="processType" Value="static"/>

<ExtendedAttribute Name="robotRunMode" Value="1"/>

</ExtendedAttributes>

分支条件中有两个组件，都为整个分支的开始节点，但是只有else分支中的节点为结束节点。

* if…else if…else

if…else if…else在执行是也是将每个节点都设置为开始节点，与if… if…else不同的是if…else if…else每个节点即使开始节点也是结束节点，如果根据条件判断执行符合条件的组件是，当前组件在执行过后则调用结束当前子流程的方法，提交整个分支，具体xpdl描述如下：

<ExtendedAttributes>

<ExtendedAttribute Name="SubType" Value="2"/>

<ExtendedAttribute Name="StartOfWorkflow" Value="default\_participant\_id;F140203; NOROUTING;STARTF140203"/>

<ExtendedAttribute Name="StartOfWorkflow" Value="default\_participant\_id;NOROUTING;STARTF142437"/>

<ExtendedAttribute Name="EndOfWorkflow" Value="default\_participant\_id;F140203;NOROUTING;ENDNDF140203"/>

<ExtendedAttribute Name="EndOfWorkflow" Value="default\_participant\_id;F142437; NOROUTING;ENDNDF142437"/>

<ExtendedAttribute Name="processType" Value="static"/>

<ExtendedAttribute Name="robotRunMode" Value="1"/>

</ExtendedAttributes>

分支条件中有两个组件，都为整个分支的开始节点，同时这两个开始节点是该分支的结束节点。