## 1.创建一个GMF项目

New->Project->GMF Project

## 2.创建领域模型(Domain Model)

首先进行模型的设计，我们的模型里，有以下几种元素：

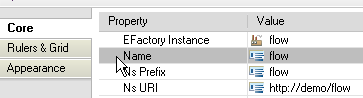
1. Flow 它代表一个流程
2. 检查节点：即分支，对某个条件进行检查，根据检查结果，可以有两种不同的后续节点，后续节点可以是除开始节点的任何节点。
3. 执行节点：表示这一步做某种操作。可以有一个后续节点。
4. 结束节点。

### 2.1 创建Ecore文件

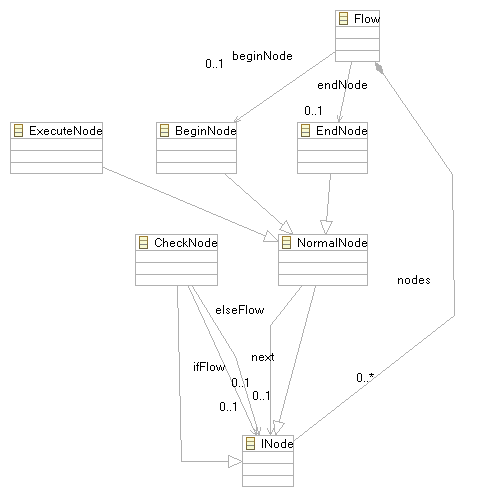
GMF框架使用EMF模型作为建模框架，EMF模型有几种来源，UML，XMI，XSD等。

这里使用ecore的方式做领域模型，方法New->Other->EMF Example Creation Wizard->Ecore Model, 输入名字flow.ecore，为方便，使用UML的方式编辑模型，在flow.ecore上面右键，'Initialize ecore\_diagram diagram file',会自动生成对应的图形化的文件flow.ecore\_diagram，双击打开flow.ecore\_diagram。

在打开的UML编辑器中，点击空白处，在属性视图中填写：



最终的模型UML如图：

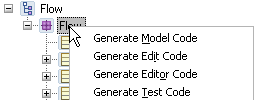


### 2.2 EMF模型

定义好ECore以后，创建EMF模型，New->Emf->EMF Model,填写名字，选择刚才创建的flow.ecore.

### 2.3 生成代码

打开上步生成的flow.genmodel，在Flow这个节点上右键，Generate Model Code和Generate Edit Code：

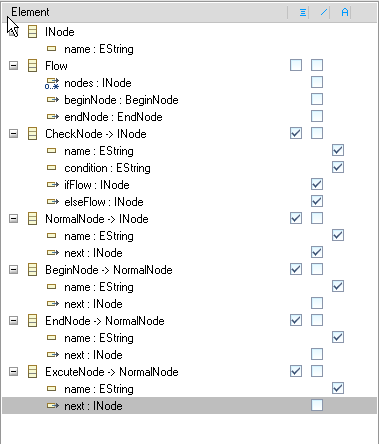


其中，生成EditCode会创建另外一个项目demo.flow.edit.

## 3.创建图形定义

New->Other->GMF->Simple Graphical Definition Model.

创建的时候，会让你选择Domain Model，然后选择模型对应的元素类型：

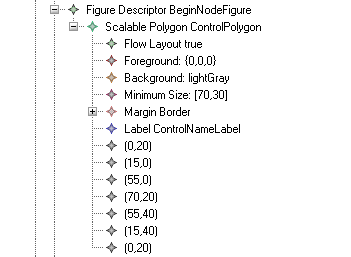


结束后，会生成一个flow.gmfgraph文件。

### 3.1自定义开始结束节点

这样会自动创建一些图形的定义，但是需要做一些修改，以满足需求。

首先，开始和结束节点都是特殊节点，需要自定义一下。找到Figure Descriptor BeginNodeFigure,调整原来默认的矩形为多边形：



并设置各节点的属性，对应代码为：

<descriptors

name="BeginNodeFigure">

<actualFigure

xsi:type="gmfgraph:ScalablePolygon"

name="ControlPolygon">

<layout

xsi:type="gmfgraph:FlowLayout"

vertical="true"

majorAlignment="CENTER"

minorAlignment="CENTER"/>

<foregroundColor

xsi:type="gmfgraph:RGBColor"/>

<backgroundColor

xsi:type="gmfgraph:ConstantColor"

value="lightGray"/>

<minimumSize

dx="70"

dy="30"/>

<border

xsi:type="gmfgraph:MarginBorder">

<insets

left="15"/>

</border>

<children

xsi:type="gmfgraph:Label"

name="ControlNameLabel"

text="Begin"/>

<template

y="20"/>

<template

x="15"/>

<template

x="55"/>

<template

x="70"

y="20"/>

<template

x="55"

y="40"/>

<template

x="15"

y="40"/>

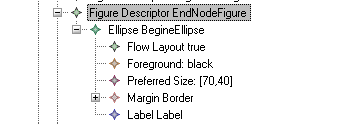
<template

y="20"/>

</actualFigure>

</descriptors>

然后，找到Figure Descriptor EndNodeFigure,调整形状为椭圆形：



对应的代码为：

<descriptors

name="EndNodeFigure">

<actualFigure

xsi:type="gmfgraph:Ellipse"

name="BegineEllipse">

<layout

xsi:type="gmfgraph:FlowLayout"

vertical="true"

majorAlignment="CENTER"

minorAlignment="CENTER"/>

<foregroundColor

xsi:type="gmfgraph:ConstantColor"

value="black"/>

<preferredSize

dx="70"

dy="40"/>

<border

xsi:type="gmfgraph:MarginBorder">

<insets

left="15"/>

</border>

<children

xsi:type="gmfgraph:Label"

name="Label"

text="End"/>

</actualFigure>

</descriptors>

### 3.2 为连接增加Label

Check节点可以有两个出口，一个是检查结果为true的时候，一个是检查结果为false的时候，为了区分，需要在连接线上增加一个Label表示区分。

GMF中增加连接的Label有些麻烦，需要按下面过程进行：

1. 首先要为Label定义Figure。

<descriptors

name="IfFlowLabelFigure">

<actualFigure

xsi:type="gmfgraph:Label"

name="IfLabel"

text="Yes"/>

</descriptors>

<descriptors

name="ElseFlowLabelFigure">

<actualFigure

xsi:type="gmfgraph:Label"

name="ElseLabel"

text="No"/>

</descriptors>

1. 然后，需要定义两个DiagramLabel，分别指向1中定义的Figure。

<labels

name="IfFlowFigureLabel"

figure="IfFlowLabelFigure"

resizeConstraint="NONE"/>

<labels

name="ElseFlowFigureLabel"

figure="ElseFlowLabelFigure"

resizeConstraint="NONE"/>

1. 在Mapping中，为Link定义两个LabelMapping。

<links>

<labelMappings

readOnly="true">

<diagramLabel

href="flow.gmfgraph#IfFlowFigureLabel"/>

</labelMappings>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.1/@tools.1"/>

<diagramLink

href="flow.gmfgraph#CheckNodeIfFlow"/>

<linkMetaFeature

xsi:type="ecore:EReference"

href="flow.ecore#//CheckNode/ifFlow"/>

<creationConstraints>

<sourceEnd

body="self &lt;> oppositeEnd"/>

</creationConstraints>

</links>

<links>

<labelMappings

readOnly="true">

<diagramLabel

href="flow.gmfgraph#ElseFlowFigureLabel"/>

</labelMappings>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.1/@tools.2"/>

<diagramLink

href="flow.gmfgraph#CheckNodeElseFlow"/>

<linkMetaFeature

xsi:type="ecore:EReference"

href="flow.ecore#//CheckNode/elseFlow"/>

<creationConstraints>

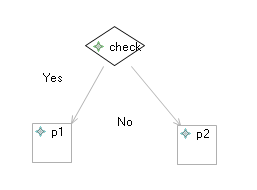
<sourceEnd

body="self &lt;> oppositeEnd"/>

</creationConstraints>

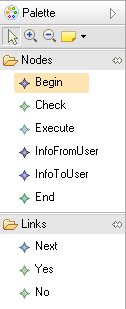
</links>

这样，连线上就会有Label，分别显示Yes，No，以区分两个出口。

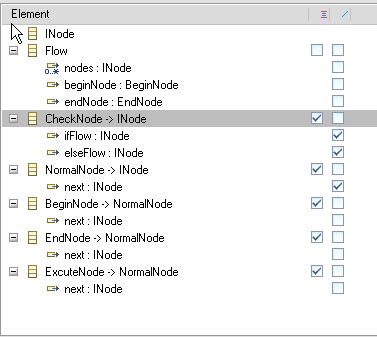


## 4.工具定义（Tooling Definition）

定义图形编辑器Palette上的工具，用户可以使用工具拖动图形到画布上，这里的定义就对应了：



与创建图形定义非常类似



去掉BeginNode, EndNode, ExcuteNode下的Next对应link的勾，因为不需要为这三个单独生成连接工具，只需利用父类NormalNode对应的工具即可。

## 5.定义映射（Mapping）

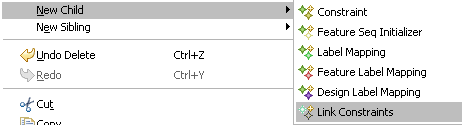
映射是从Domain Model到对应的图形、工具定义的映射；可以通过New🡪Graphical Modeling Framework🡪GMF Map model来创建一个映射文件。GMF可以自动地根据DomainModel,GraphicalDefinition,Tooling自动生成一个flow.gmfmp文件。

默认生成的映射文件并不能完全满足要求，可以进行简单定制，使其满足我们的要求。

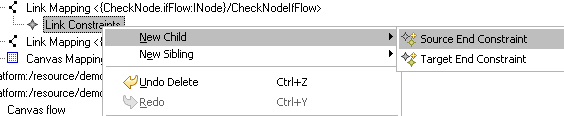
### 5.1连接的限制

首先是一个明显需要的限制，因为默认的连接，可以从一个图形连接到自身，我们需要限制这一点，使连接不能从图形连到自身。

找到生成的LinkMapping，右键->New Child->Link Constraints:



在新建的LinkConstraints上右键，New Child->Source End Constraint,



并且在新建的Source End Constraint的属性里输入，setf <> oppositeEnd，意思是连接的另一端和自身不同。

同样的，对其他两个Link Mapping节点做上述相同的操作，最终相应的代码为：

<links>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.1/@tools.0"/>

<diagramLink

href="flow.gmfgraph#NormalNodeNext"/>

<linkMetaFeature

xsi:type="ecore:EReference"

href="flow.ecore#//NormalNode/next"/>

<creationConstraints>

<sourceEnd

body="self &lt;> oppositeEnd"/>

</creationConstraints>

</links>

<links>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.1/@tools.1"/>

<diagramLink

href="flow.gmfgraph#CheckNodeIfFlow"/>

<linkMetaFeature

xsi:type="ecore:EReference"

href="flow.ecore#//CheckNode/ifFlow"/>

<creationConstraints>

<sourceEnd

body="self &lt;> oppositeEnd"/>

</creationConstraints>

</links>

<links>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.1/@tools.2"/>

<diagramLink

href="flow.gmfgraph#CheckNodeElseFlow"/>

<linkMetaFeature

xsi:type="ecore:EReference"

href="flow.ecore#//CheckNode/elseFlow"/>

<creationConstraints>

<sourceEnd

body="self &lt;> oppositeEnd"/>

</creationConstraints>

</links>

## 6.生成代码

首先要生成一个flow.gmfgen。在Mapping上右键，Create Generate Model…

然后就可以生成flow.gmfgen文件上右键，Generate Diagram Code,即可生成另外一个项目，即图形编辑器对应的项目。

此时，就可以运行，查看结果了。

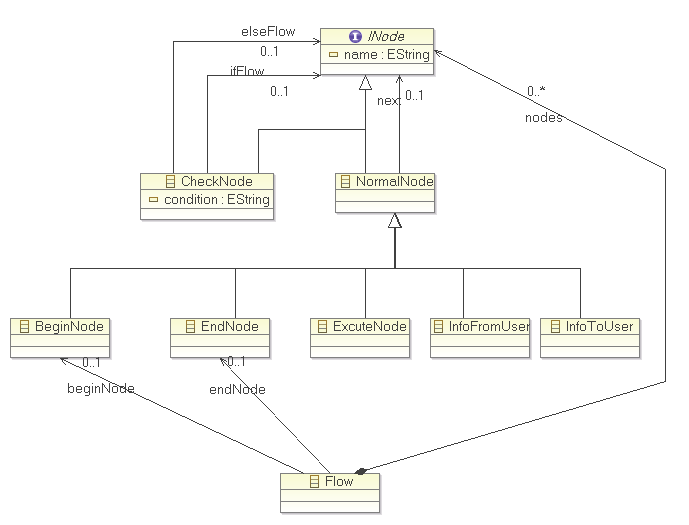
## 7.手动添加图形、工具和映射

下面为添加两个图元，即参考Visio中，“来自用户的信息”和“给用户的信息”。

### 7.1 增加模型定义

首先，需要在模型里添加，在UML中，增加两个Class，InfoFromUser和InfoToUser，分别代表“来自用户的信息”和“给用户的信息”。这两个模型添加好以后，需要更新flow.genmodel，在flow.genmodel上右键🡪reload；最后，打开flow.genmodel并重新generate model code和generate edit code。

新的UML图为：



### 7.2 调整/增加工具定义

打开flow.gmftool，在Tool Group Nodes下，增加两个Creation Tool，分别用来创建InfoFromUser和InfoToUser：



对应的代码为：

<tools

xsi:type="gmftool:CreationTool"

title="InfoFromUser"

description="">

<smallIcon

xsi:type="gmftool:DefaultImage"/>

<largeIcon

xsi:type="gmftool:DefaultImage"/>

</tools>

<tools

xsi:type="gmftool:CreationTool"

title="InfoToUser">

<smallIcon

xsi:type="gmftool:DefaultImage"/>

<largeIcon

xsi:type="gmftool:DefaultImage"/>

</tools>

### 7.3 增加图形定义

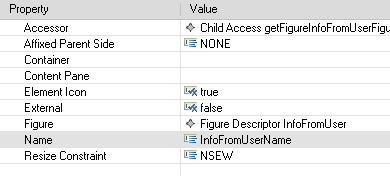
打开flow.gmfgraph，增加两个Figure Descriptor:



然后，参考BeginNode的图形定义的方法，为这两个元素设计自定义的图形元素，同时添加相应的ChildAccess。

添加两个Nodes Node节点，并对应刚建立的Figure，再添加两个DiagramLabel，对应刚建立的Figure和ChildAccess：





Figure对应的代码：

<descriptors

name="InfoFromUserFigure">

<actualFigure

xsi:type="gmfgraph:ScalablePolygon"

name="InfoFromUserPolygon">

<layout

xsi:type="gmfgraph:FlowLayout"

vertical="true"

majorAlignment="CENTER"

minorAlignment="CENTER"/>

<preferredSize

dx="70"

dy="40"/>

<children

xsi:type="gmfgraph:Label"

name="InfoFromUserFigureLabel"/>

<template/>

<template

x="70"/>

<template

x="55"

y="20"/>

<template

x="70"

y="40"/>

<template

y="40"/>

<template/>

</actualFigure>

<accessors

figure="//@figures.0/@descriptors.8/@actualFigure/@children.0"/>

</descriptors>

<descriptors

name="InfoToUserFigure">

<actualFigure

xsi:type="gmfgraph:ScalablePolygon"

name="InfoToUserPolygon">

<layout

xsi:type="gmfgraph:FlowLayout"

vertical="true"

majorAlignment="CENTER"

minorAlignment="CENTER"/>

<preferredSize

dx="70"

dy="40"/>

<children

xsi:type="gmfgraph:Label"

name="InfoToUserFigureLabel"/>

<template

x="15"/>

<template

x="70"/>

<template

x="70"

y="40"/>

<template

x="15"

y="40"/>

<template

y="20"/>

<template

x="15"/>

</actualFigure>

<accessors

figure="//@figures.0/@descriptors.9/@actualFigure/@children.0"/>

</descriptors>

Nodes对应的代码：

<nodes

name="InfoFromUser"

figure="InfoFromUserFigure"/>

<nodes

name="InfoToUser"

figure="InfoToUserFigure"/>

Diagram Label对应的代码：

<labels

name="InfoFromUserName"

figure="InfoFromUserFigure"

accessor="//@figures.0/@descriptors.8/@accessors.0"/>

<labels

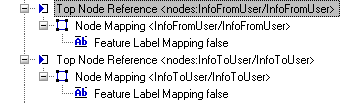
name="InfoToUserName"

figure="InfoToUserFigure"

accessor="//@figures.0/@descriptors.9/@accessors.0"/>

### 7.4增加映射定义：

完成了工具和图形定义后，打开flow.gmfmap，修改并增加相应的Mapping.仿照Check和Execute节点，增加两个Top Node Reference节点：



对应的代码：

<nodes>

<containmentFeature

href="flow.ecore#//Flow/nodes"/>

<ownedChild>

<domainMetaElement

href="flow.ecore#//InfoFromUser"/>

<labelMappings

xsi:type="gmfmap:FeatureLabelMapping">

<diagramLabel

href="flow.gmfgraph#InfoFromUserName"/>

<features

href="flow.ecore#//INode/name"/>

<editableFeatures

href="flow.ecore#//INode/name"/>

</labelMappings>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.0/@tools.3"/>

<diagramNode

href="flow.gmfgraph#InfoFromUser"/>

</ownedChild>

</nodes>

<nodes>

<containmentFeature

href="flow.ecore#//Flow/nodes"/>

<ownedChild>

<domainMetaElement

href="flow.ecore#//InfoToUser"/>

<labelMappings

xsi:type="gmfmap:FeatureLabelMapping">

<diagramLabel

href="flow.gmfgraph#InfoToUserName"/>

<features

href="flow.ecore#//INode/name"/>

<editableFeatures

href="flow.ecore#//INode/name"/>

</labelMappings>

<tool

xsi:type="gmftool:CreationTool"

href="flow.gmftool#//@palette/@tools.0/@tools.4"/>

<diagramNode

href="flow.gmfgraph#InfoToUser"/>

</ownedChild>

</nodes>

完成后，通过flow.gmfmap重新生成flow.gmfgen，并且重新生成Diagram Code，重新运行程序，可以看到相应的效果。

最终效果：

