

3. XLab 用户使用手册 V2. 1

XLab 用户使用手册 V2. 1

目录

一、 整体描述.....	548
二、 功能模块.....	549
1. 项目管理.....	549
1.1. 新建项目.....	407
1.2. 展开项目.....	407
1.3. 新建包.....	408
1.4. 新建连续类.....	408
1.5. 打开连续类.....	409
1.6. 定义图.....	410
1.7. 定义图-类属性弹窗	410
1.8. 方程图.....	414
1.9. 编辑方程图.....	414
1.10. 编辑方程图-方程组	416
1.11. 编辑方程图-条件方程	417
1.12. 编辑方程图-循环方程	418
1.13. 编辑方程图-内部事件方程	419
1.14. 编辑方程图-外部事件方程	420
1.15. 编辑方程图-确定	421
1.16. 连续类-查看文本（图形-->文本）	422
1.17. 连续类-保存文本（文本-->图形）	423
1.18. 新建离散类.....	424
1.19. 离散类-状态机图	425
1.20. 离散类-添加初始状态	426
1.21. 编辑状态机图-接收事件	429
1.22. 编辑状态机图-时间事件	430
1.23. 编辑状态机图-输出	431
1.24. 编辑状态机图-确定	431
1.25. 离散类-添加普通状态	432
1.26. 离散类-添加复合状态	434
1.27. 离散类-添加结束状态	437
1.28. 离散类-状态机图之前跳转.....	437
1.29. 离散类-查看文本（图形转文本）	440
1.30. 离散类-保存文本（文本转图形）	441
1.31. 新建智能体类.....	443
1.32. 打开智能体类.....	445

其他绩效评价材料

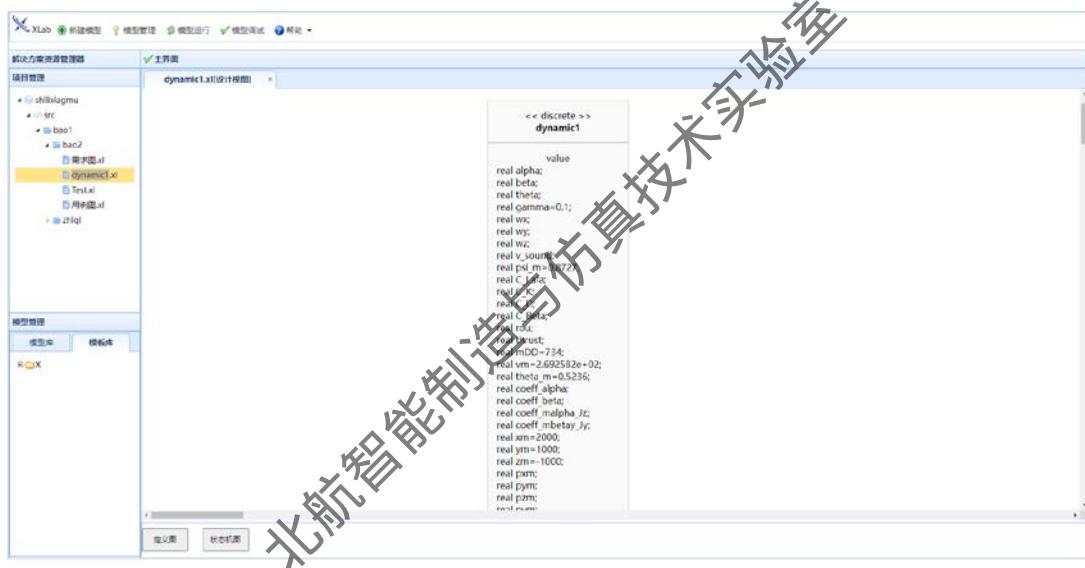
1.33. 智能体类-活动图	447
1.34. 智能体类-打开 plan	452
1.35. 智能体类-查看文本（图形到文本）	454
1.36. 智能体类-文本保存（文本到图形）	455
1.37. 新建耦合类.....	457
1.38. 打开耦合类.....	458
1.39. 耦合类-定义图	458
1.40. 耦合类-连接图	461
1.41. 耦合类-查看文本（图形到文本）	463
1.42. 耦合类-文本保存（文本到图形）	464
1.43. 新建连接器类.....	467
1.44. 打开连接器类.....	468
1.45. 新建记录类.....	469
1.46. 新建函数类.....	471
1.47. 打开函数类.....	472
1.48. 函数类-活动图	473
1.49. 函数类-查看文本（图形到文本）	475
1.50. 函数类-文本保存（文本到图形）	476
1.51. 新建用例图.....	478
1.52. 打开用例图.....	479
1.53. 编辑用例图.....	480
1.54. 用例图-查看文本（图形到文本）	484
2. 模型管理.....	645
2.1. 模型库.....	499
2.2. 模板库.....	502
3. 帮助.....	648
3.1. 文件下载.....	502

一、整体描述

访问地址: <http://localhost:9888/xYuYanKaiFaGongJu/index.html#>

温馨提示: 请使用谷歌浏览器访问, 分辨率为 1920*1080

整体界面如图所示:



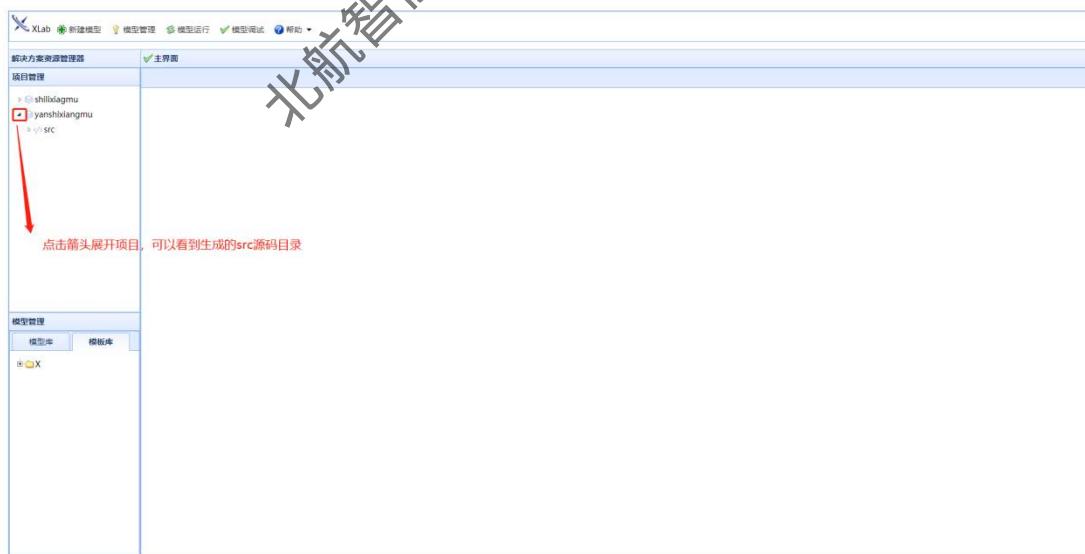
二、功能模块

1. 项目管理

1.1. 新建项目

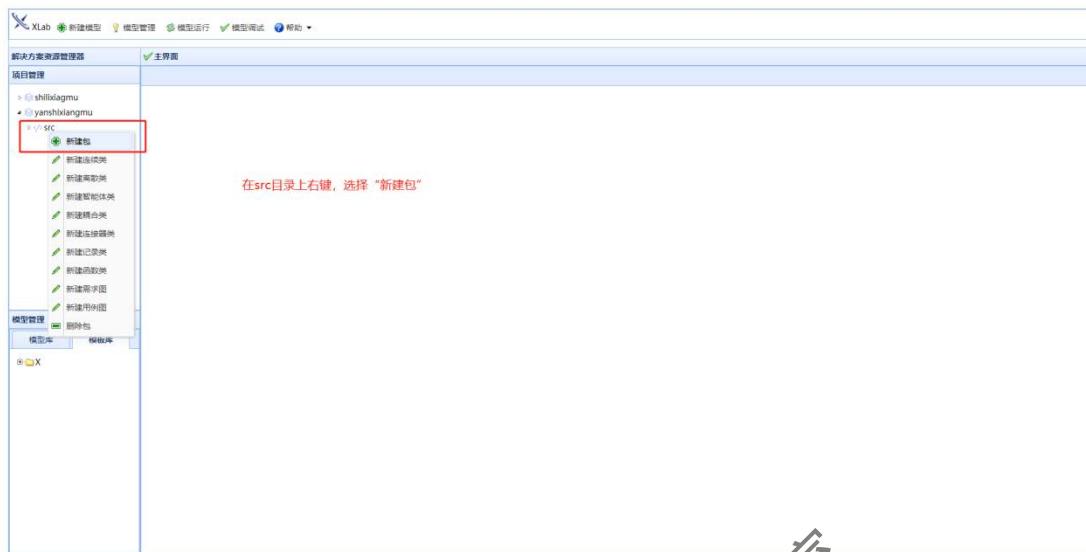


1.2. 展开项目



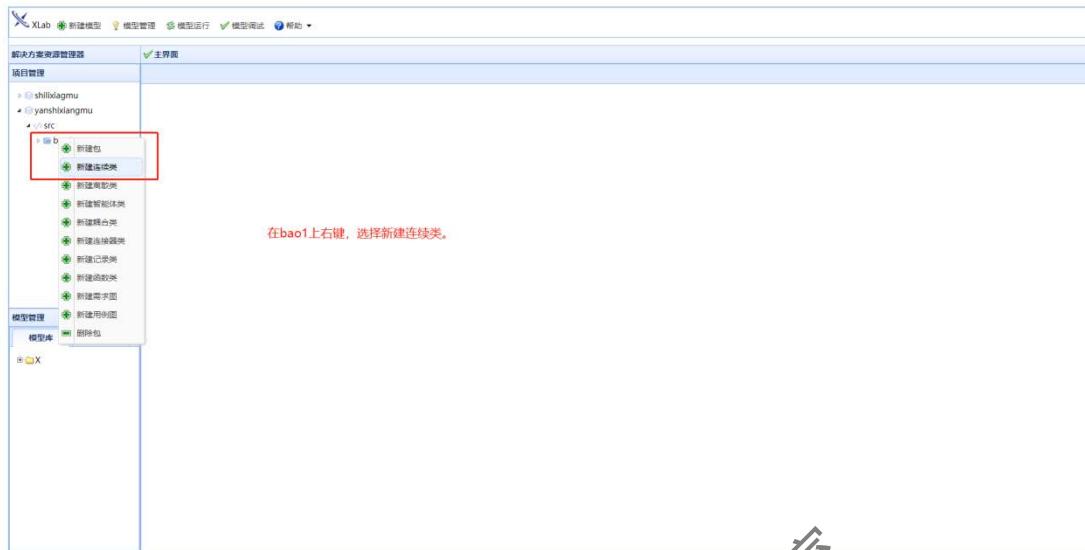
其他绩效评价材料

1.3. 新建包



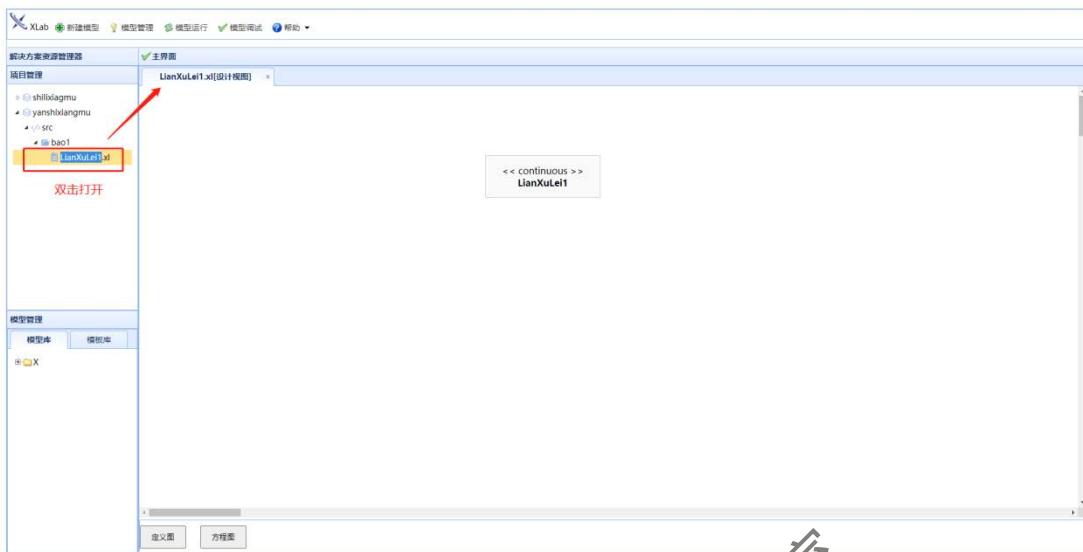
其他绩效评价材料

1.4. 新建连续类



其他绩效评价材料

1.5. 打开连续类

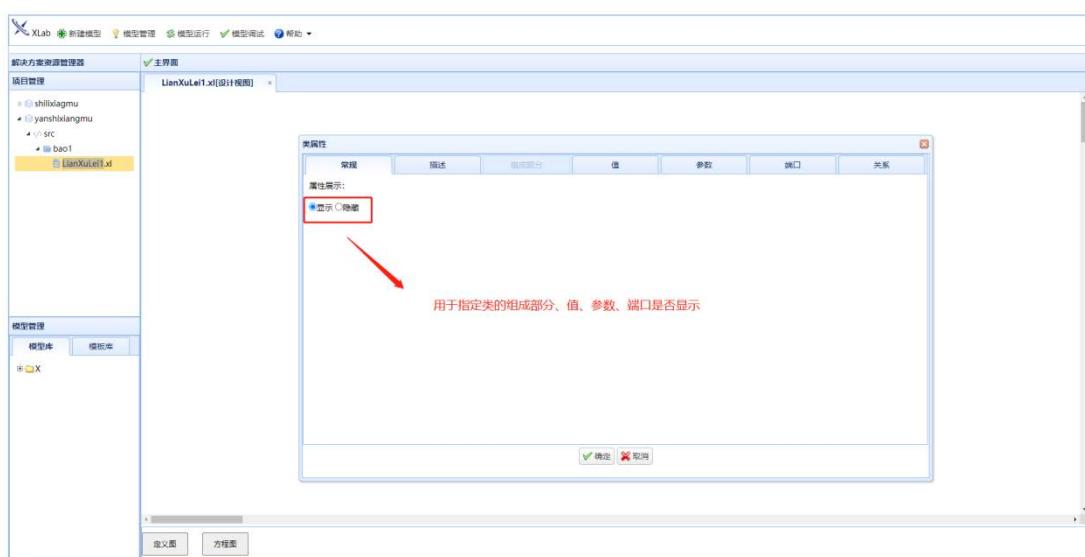
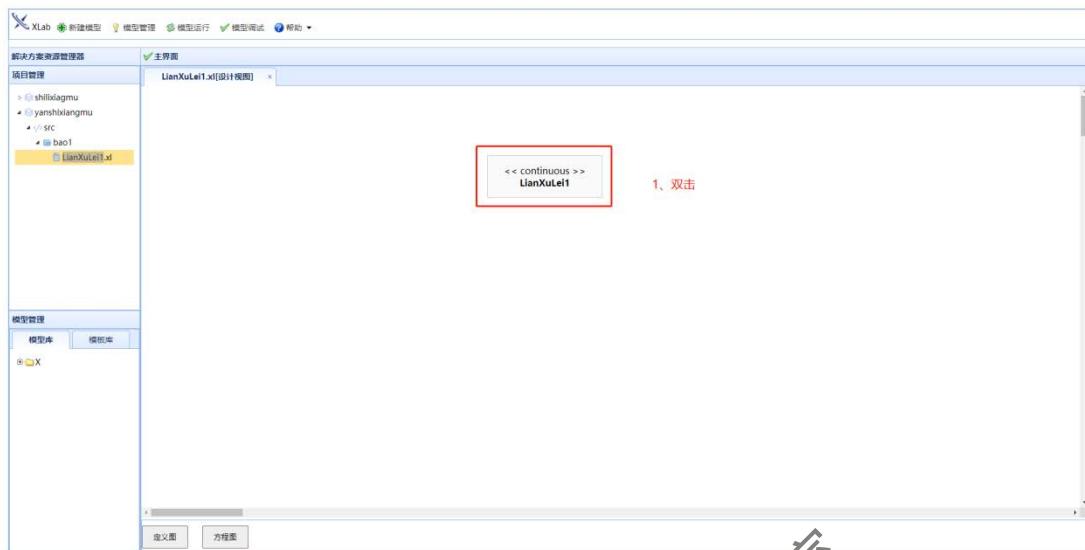


1.6. 定义图

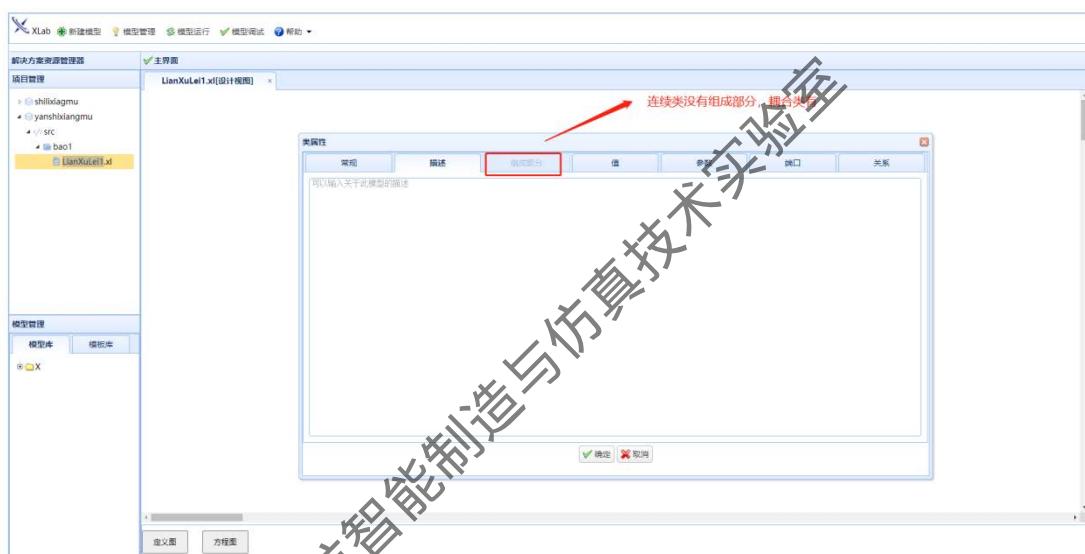
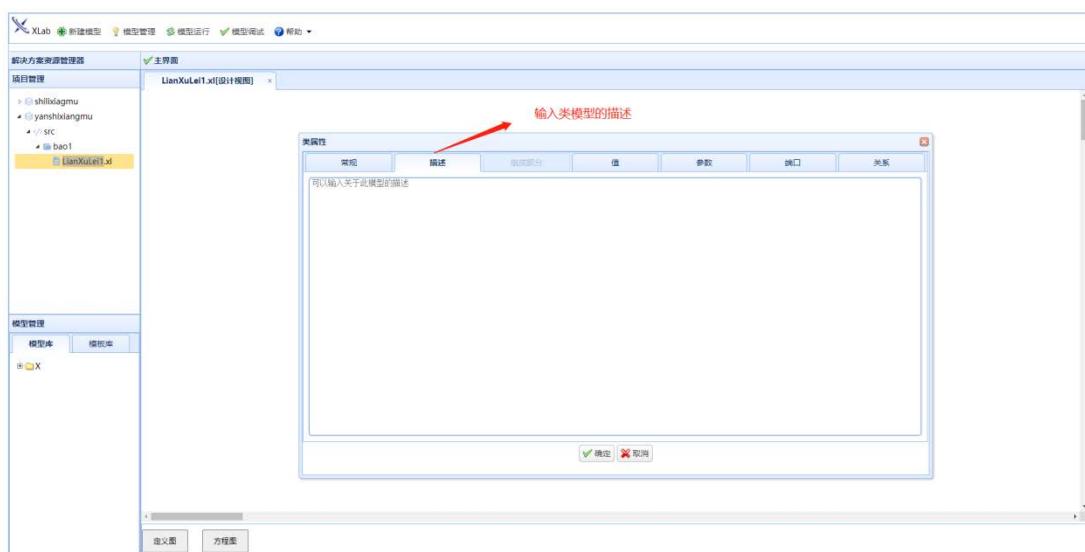


其他绩效评价材料

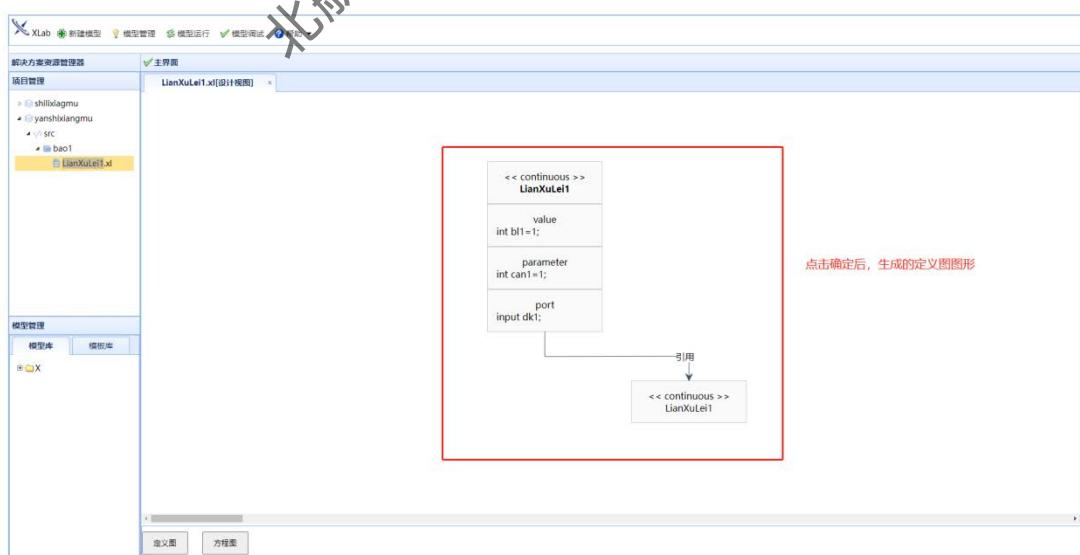
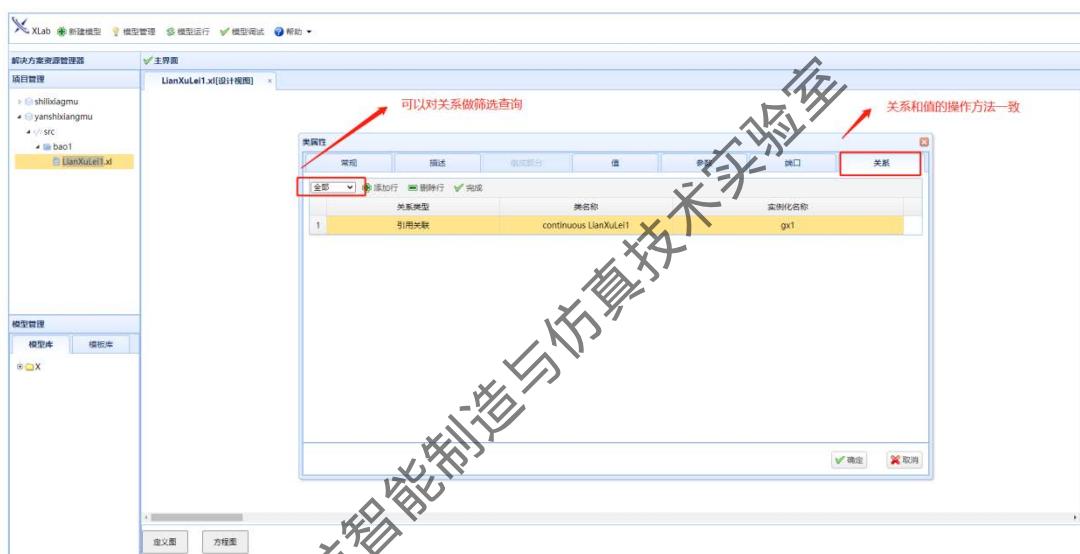
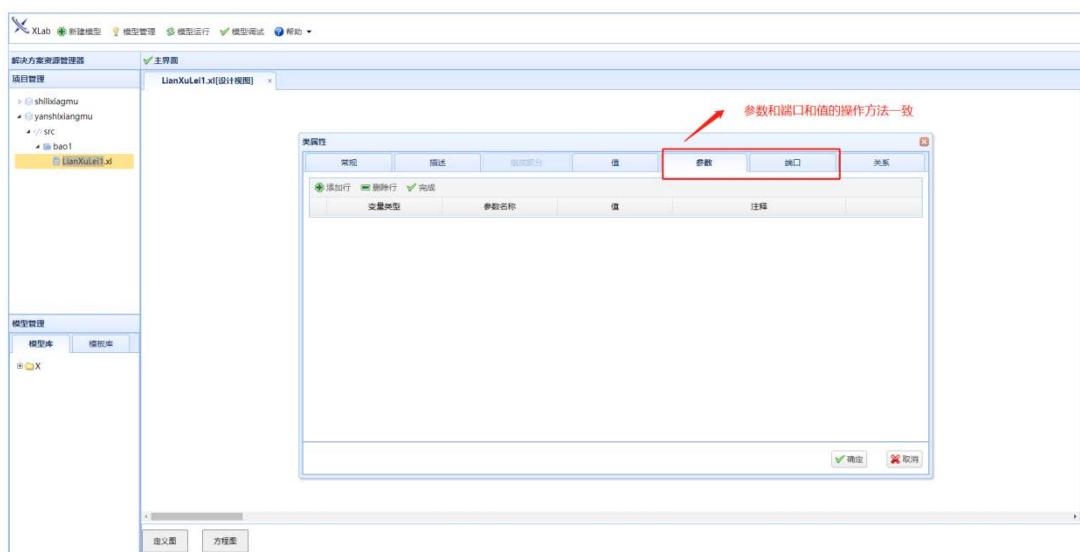
1.7. 定义图-类属性弹窗



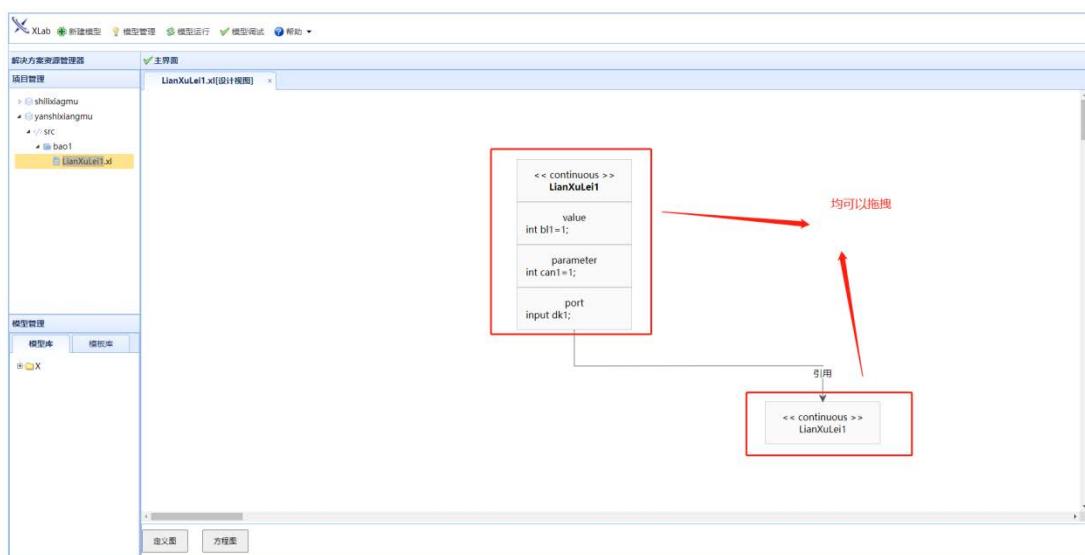
其他绩效评价材料



其他绩效评价材料



其他绩效评价材料

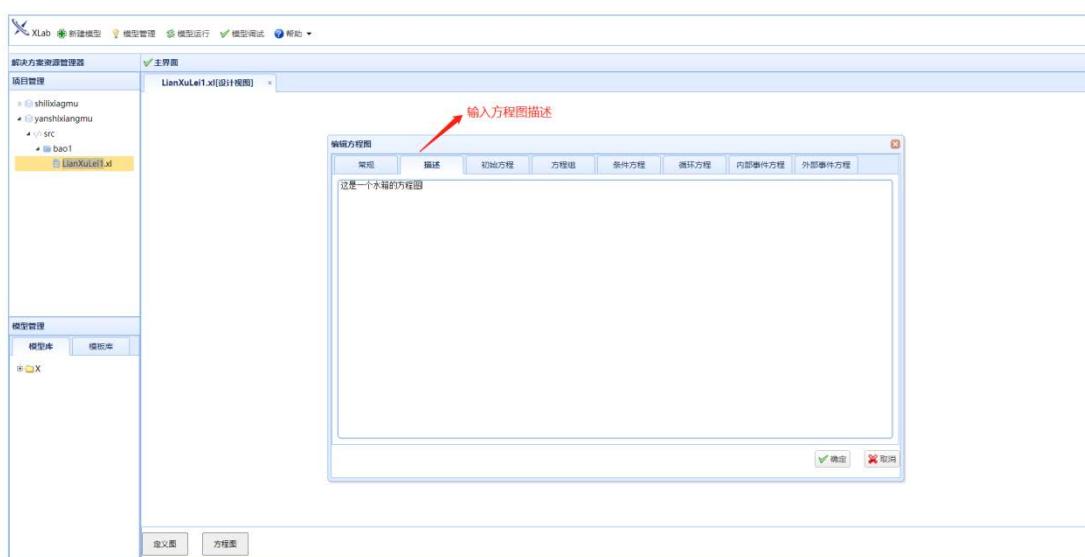


1.8. 方程图

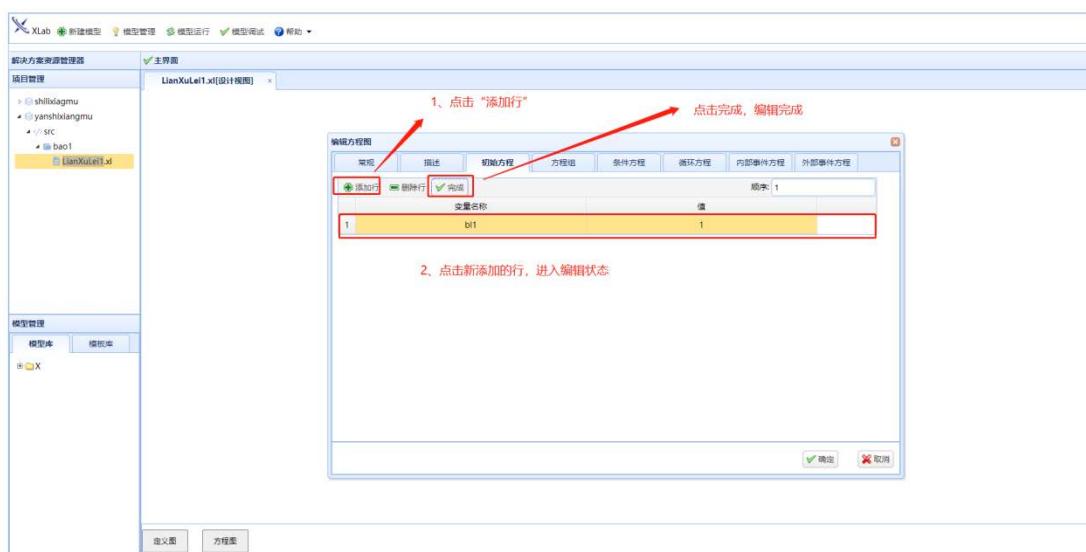


其他绩效评价材料

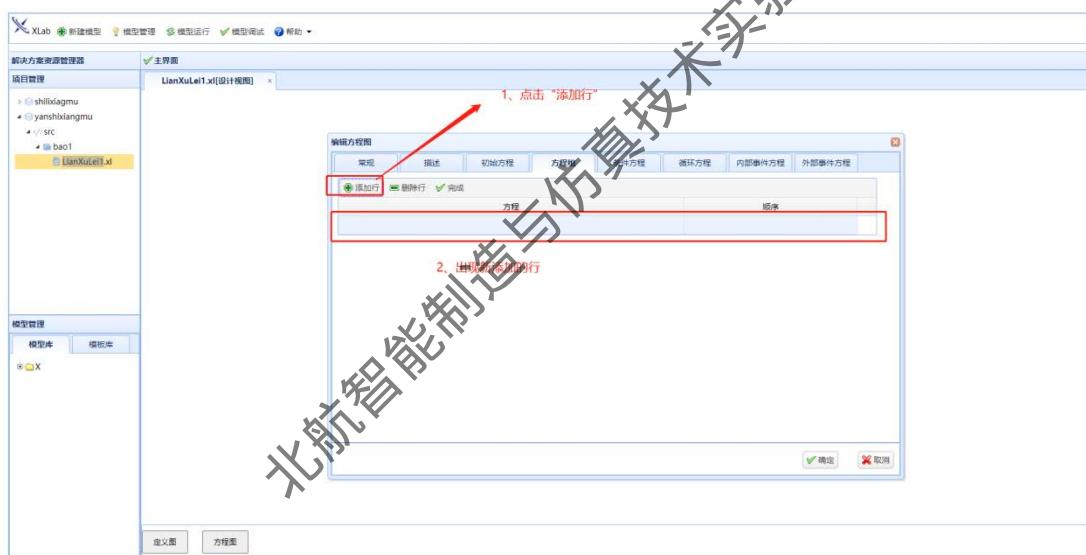
1.9. 编辑方程图



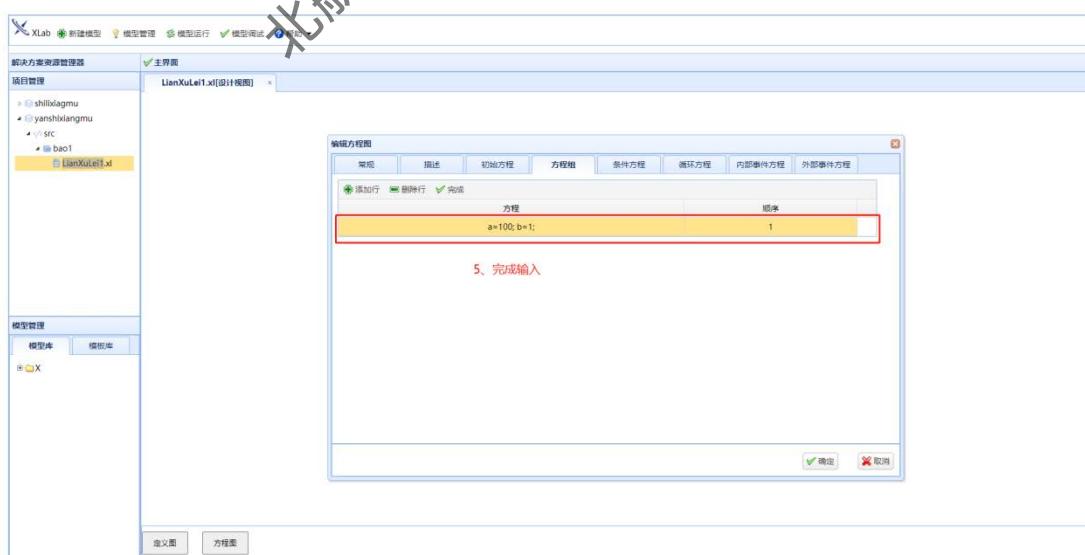
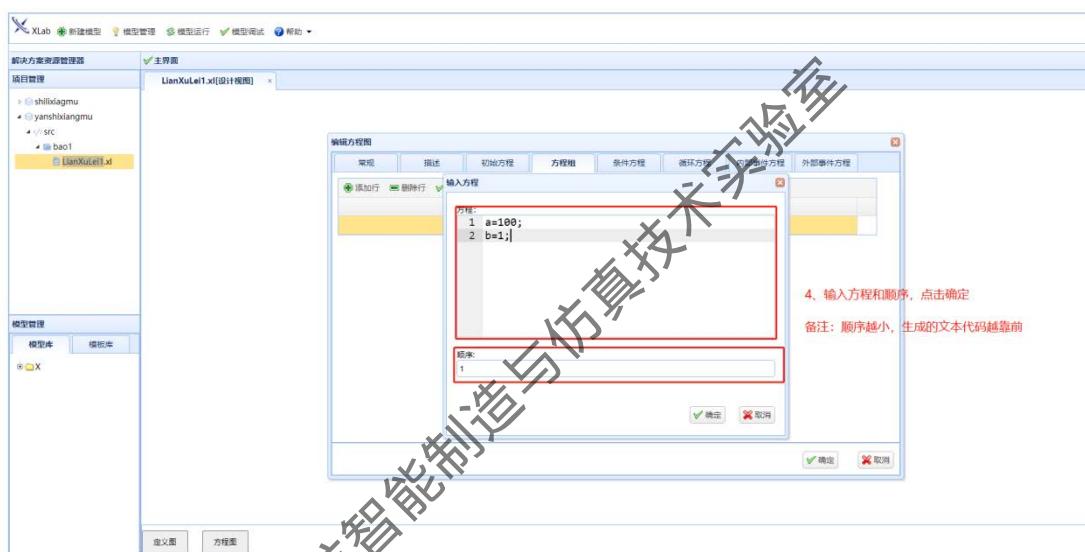
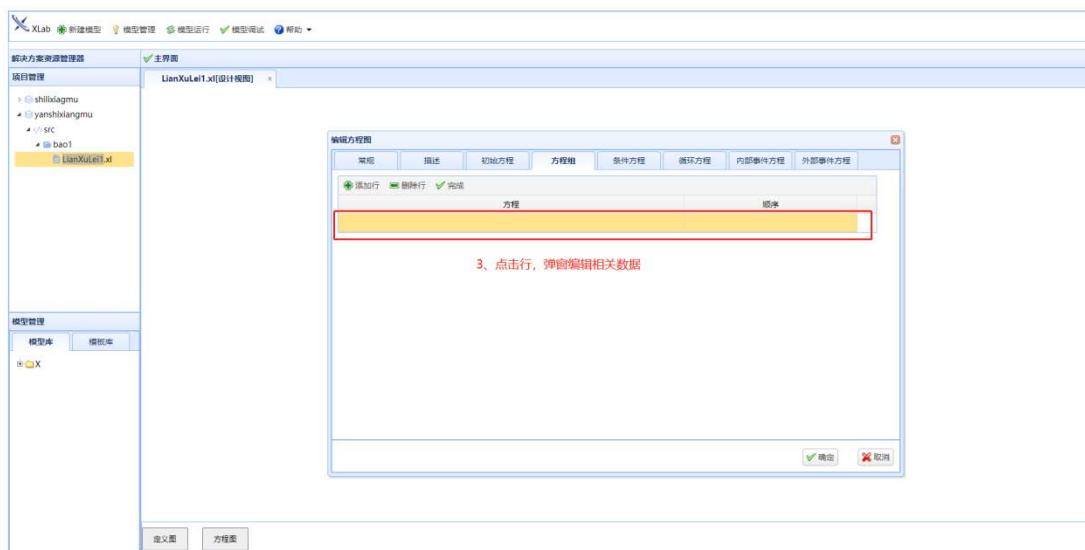
其他绩效评价材料



1.10. 编辑方程图-方程组

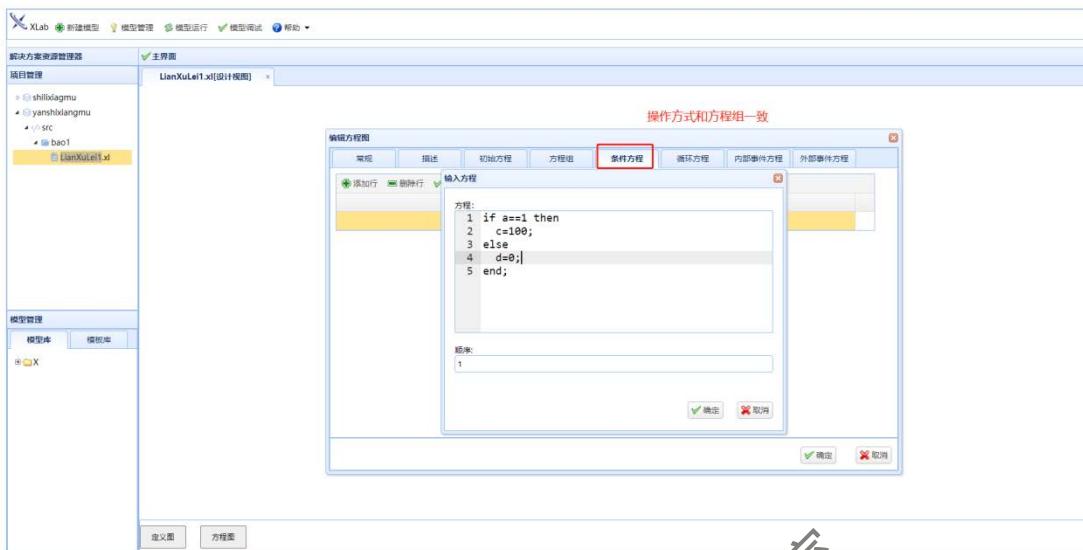


其他绩效评价材料



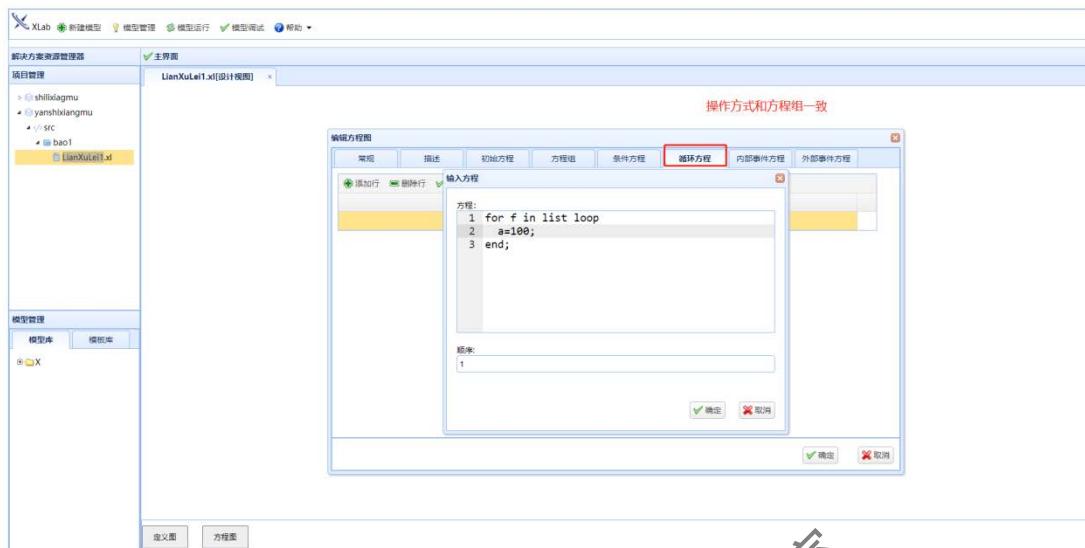
其他绩效评价材料

1.11. 编辑方程图-条件方程



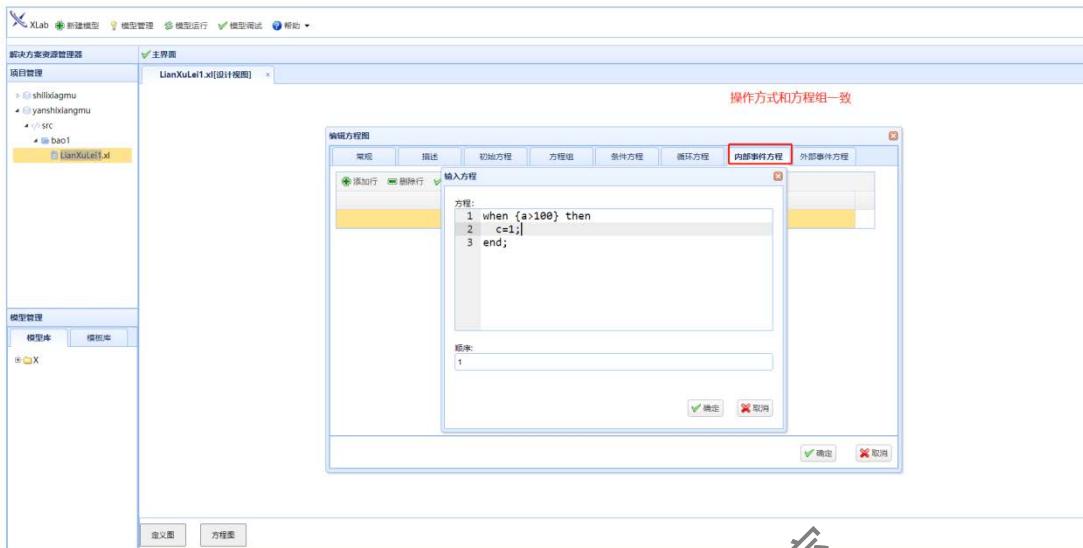
其他绩效评价材料

1.12. 编辑方程图-循环方程



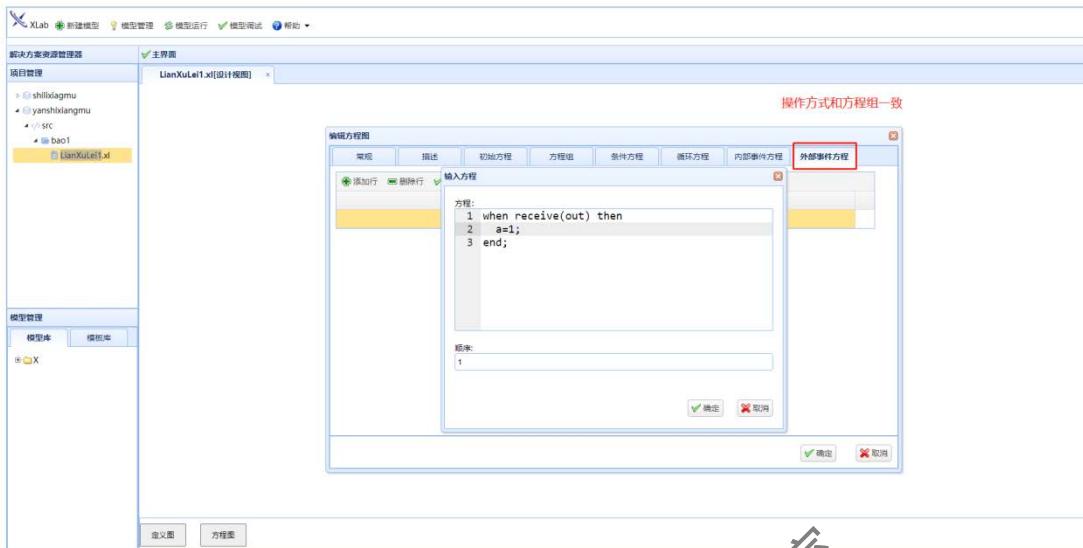
其他绩效评价材料

1.13. 编辑方程图-内部事件方程



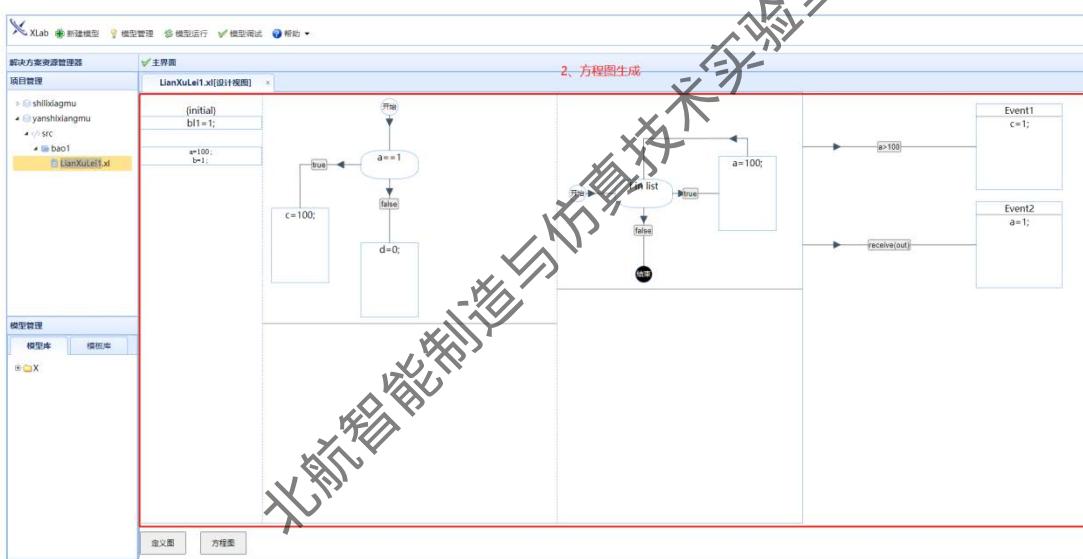
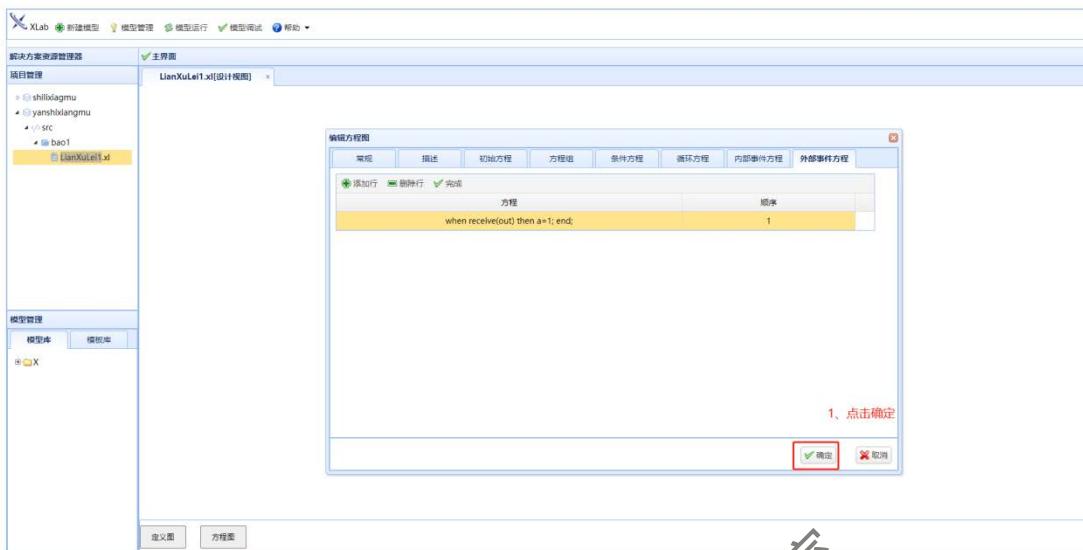
其他绩效评价材料

1.14. 编辑方程图-外部事件方程



其他绩效评价材料

1.15. 编辑方程图-确定



其他绩效评价材料

1.16. 连续类-查看文本（图形-->文本）



其他绩效评价材料

1.17. 连续类-保存文本（文本--》图形）

The screenshot shows the XLab interface with a project named 'yanshilangmu'. A file named 'LianXulei1.xl' is open in the main editor. The code is a continuous class definition:

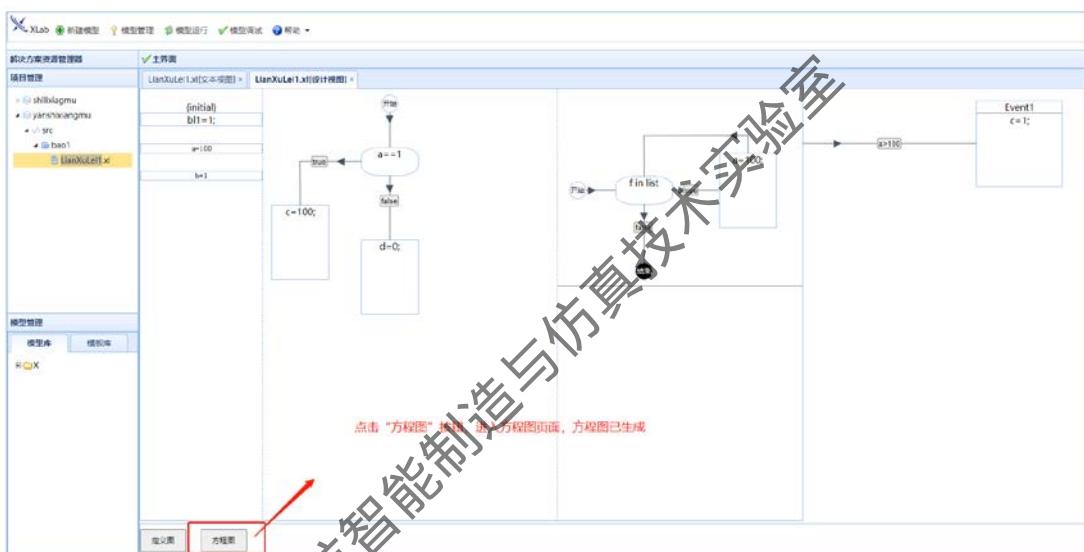
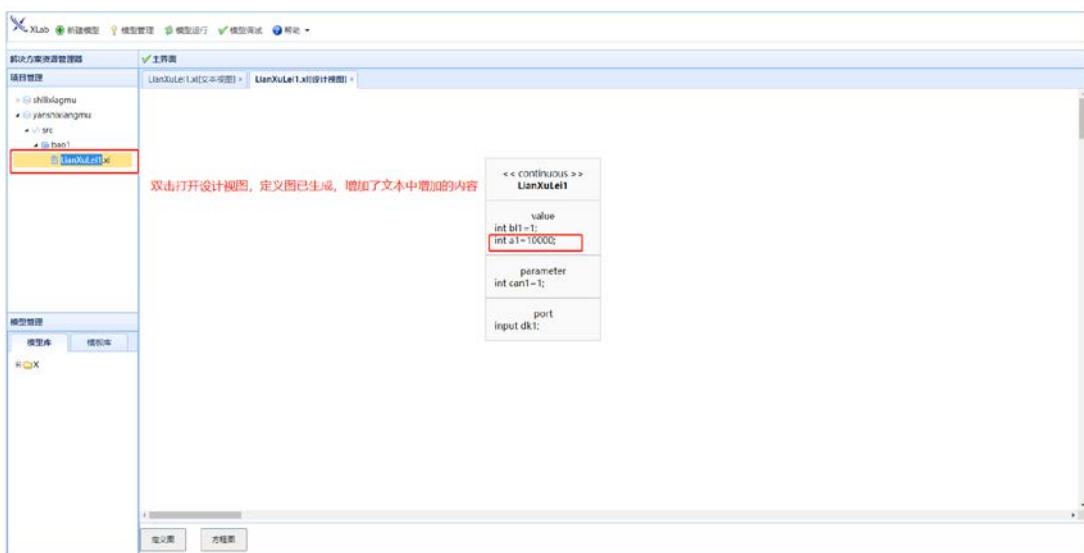
```
continuous LianXulei1
value:
int b1=1;
int a1=10000;
parameter:
int can1=1;
port:
input int dk1;
initial equation:
b1=1;
equation:
a=100;
b1;
if a==1 then
c=100;
else
d=0;
end;
for f in list loop
a=100;
end;
when {a>100} then
c=1;
end;
when receive(out) then
a=1;
end;
end;
```

A red annotation highlights the line 'int a1=10000;' with the text '此处增加了一个value, 整个文本都可以修改' (A value was added here, the entire text can be modified).

The screenshot shows the XLab interface with the same project and file. A red arrow points to the '保存' (Save) button in the toolbar.

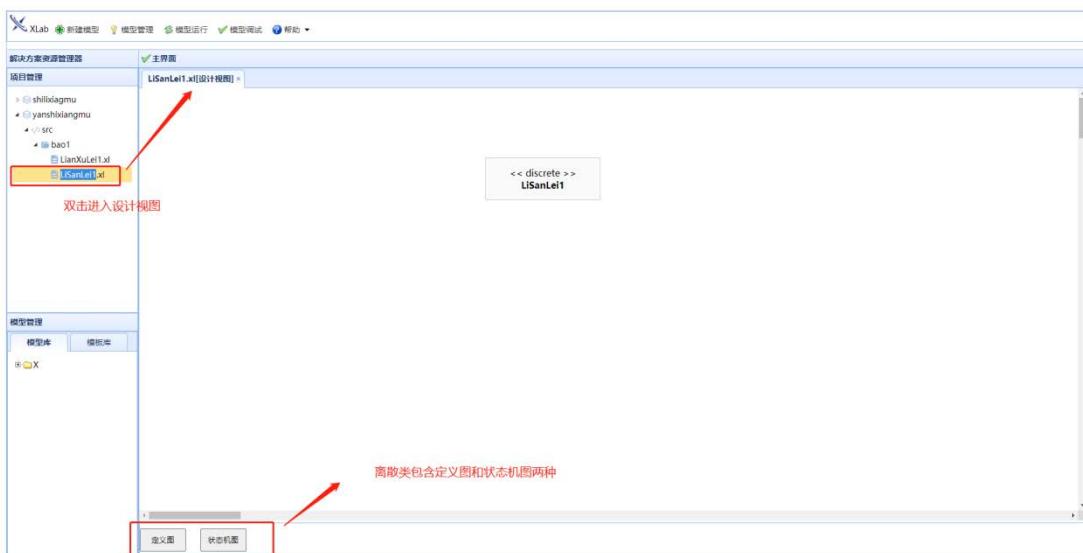
The screenshot shows the XLab interface after saving. A message box displays '保存成功' (Save successful). The status bar at the bottom right also shows '提示保存成功' (Save successful).

其他绩效评价材料



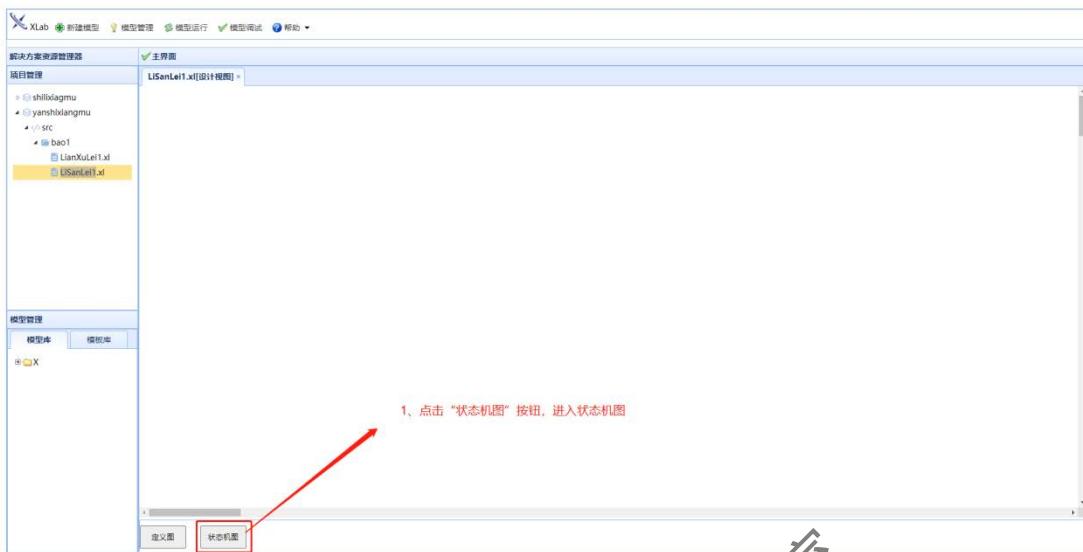
其他绩效评价材料

1.18. 新建离散类



其他绩效评价材料

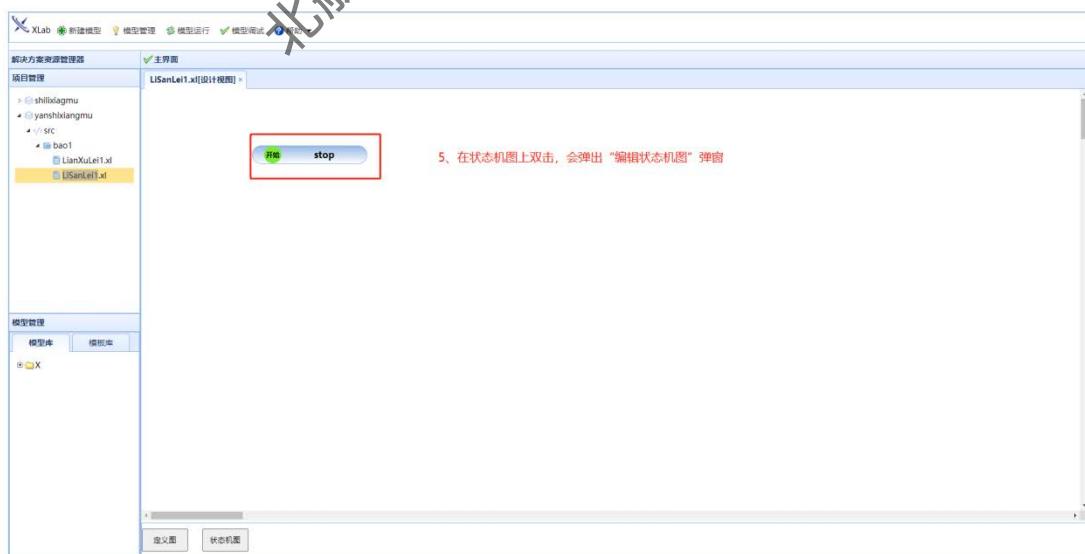
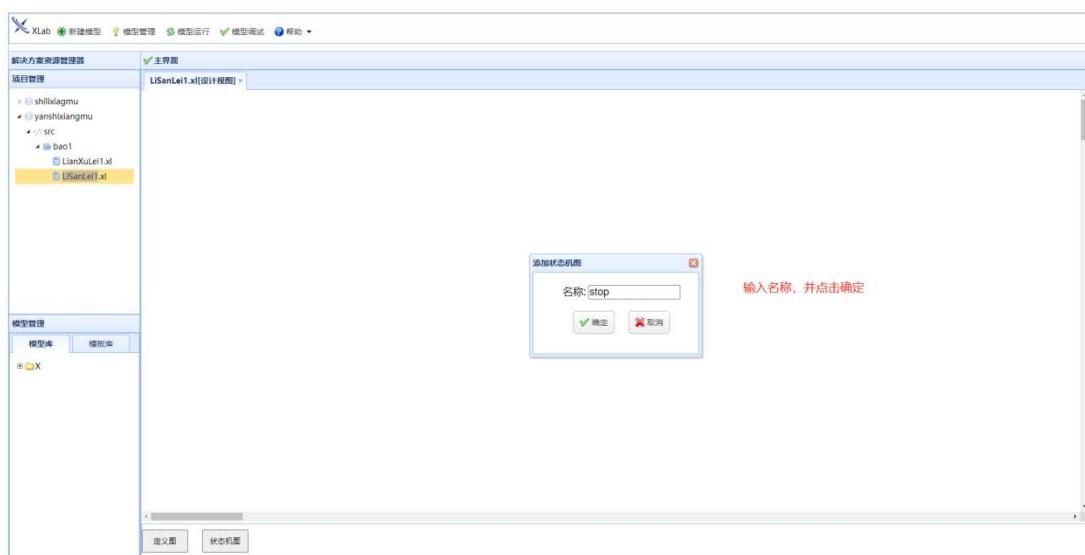
1.19. 离散类-状态机图



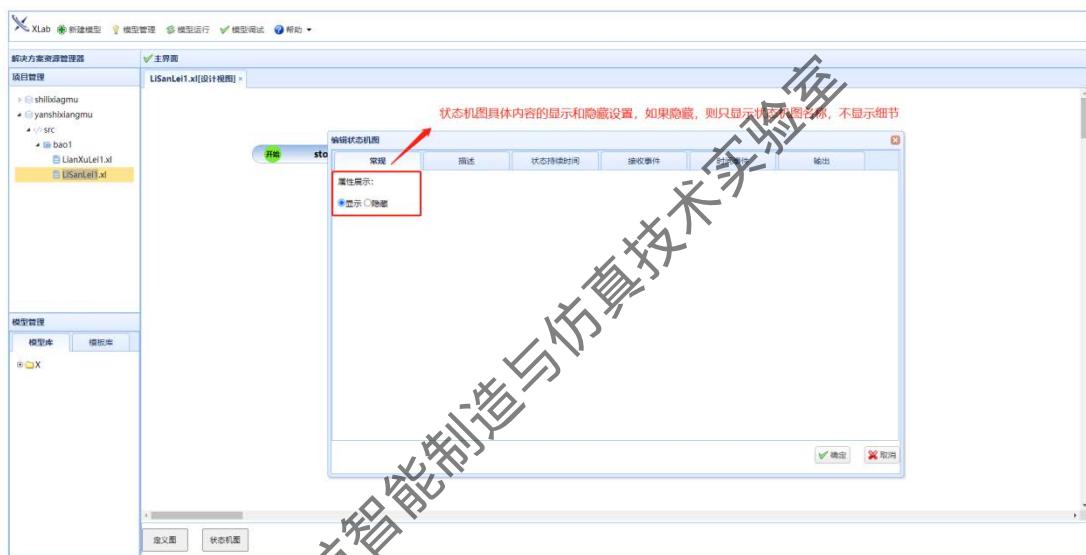
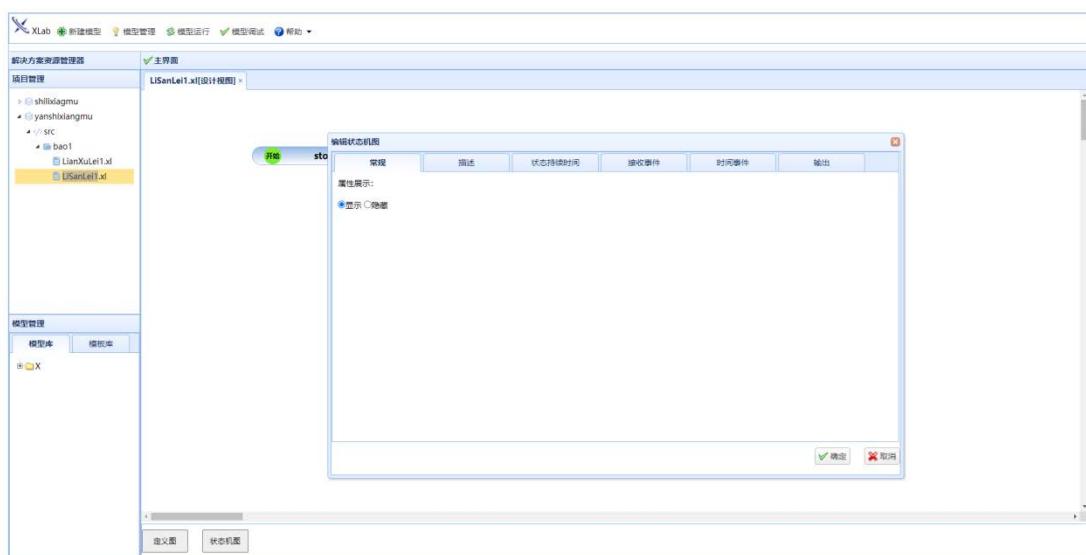
1.20. 离散类-添加初始状态



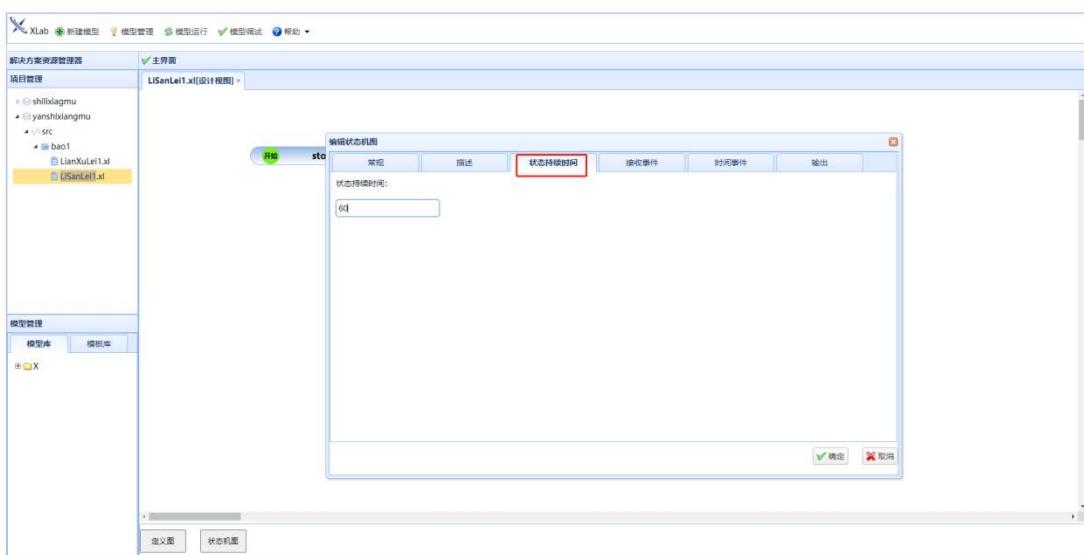
其他绩效评价材料



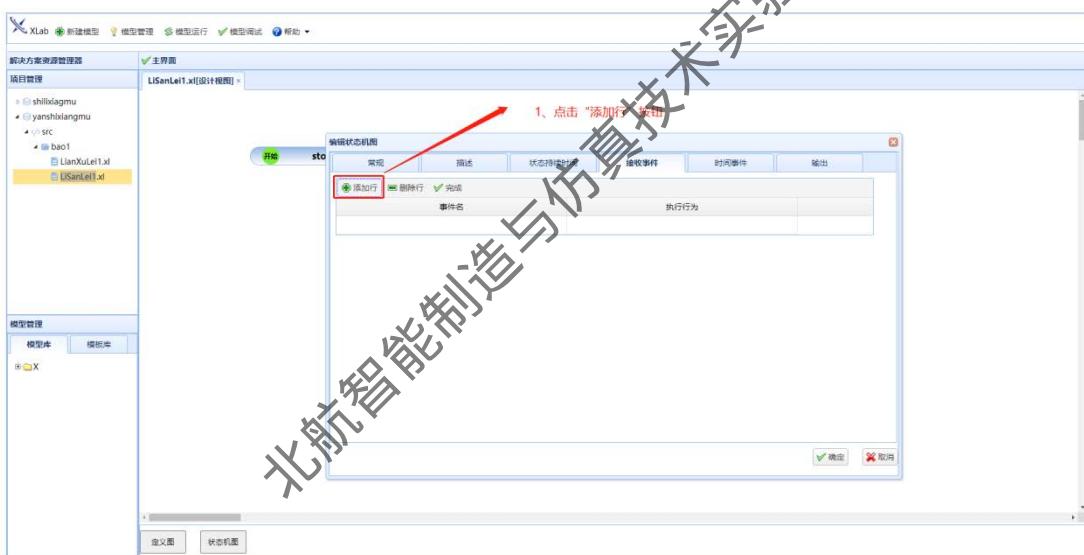
其他绩效评价材料



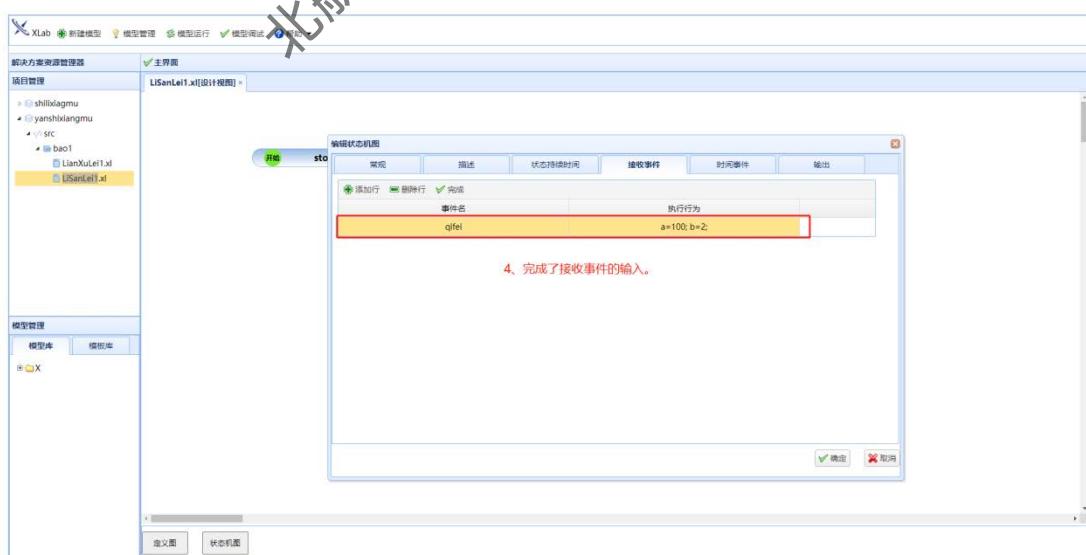
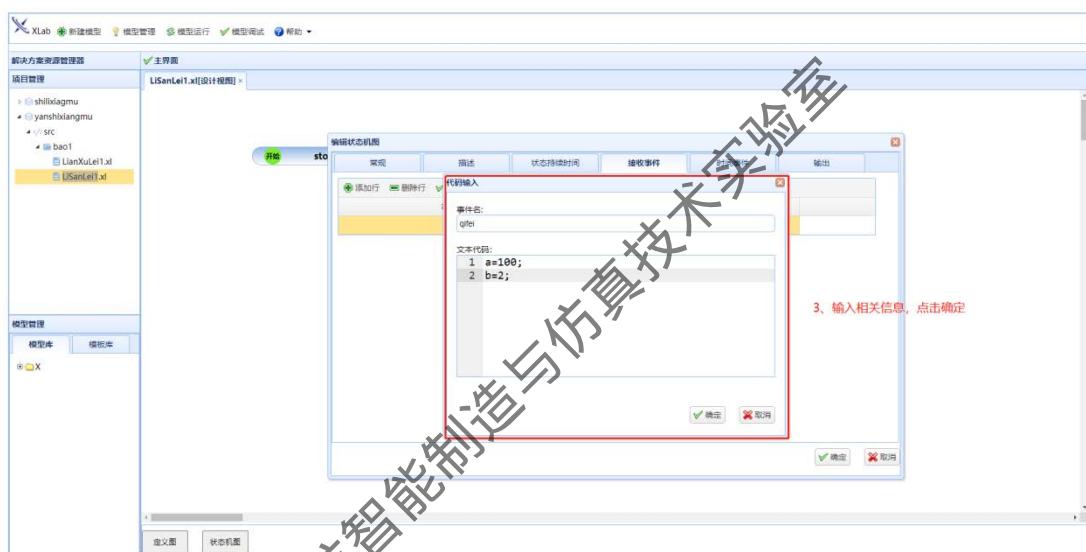
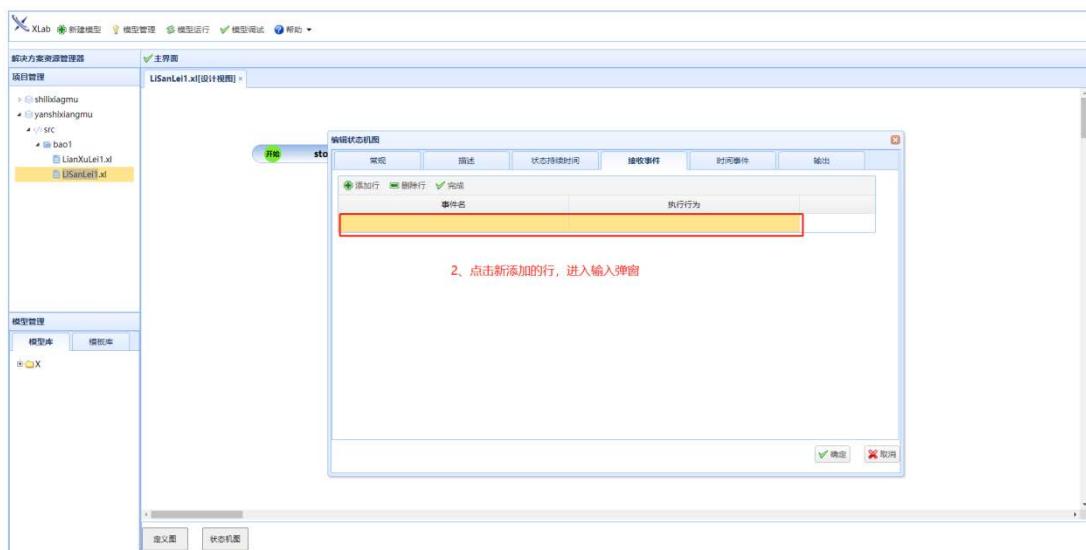
其他绩效评价材料



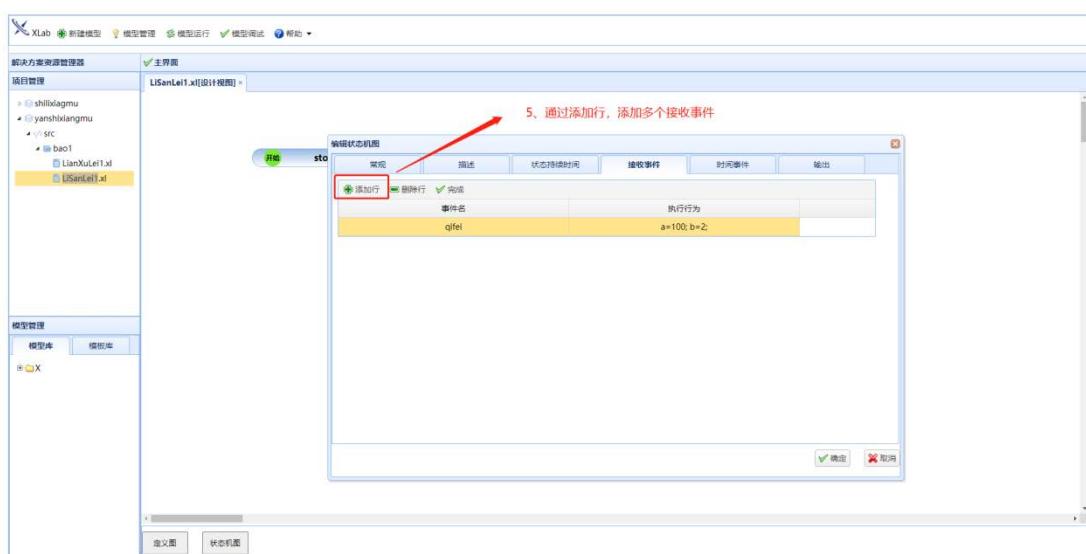
1.21. 编辑状态机图-接收事件



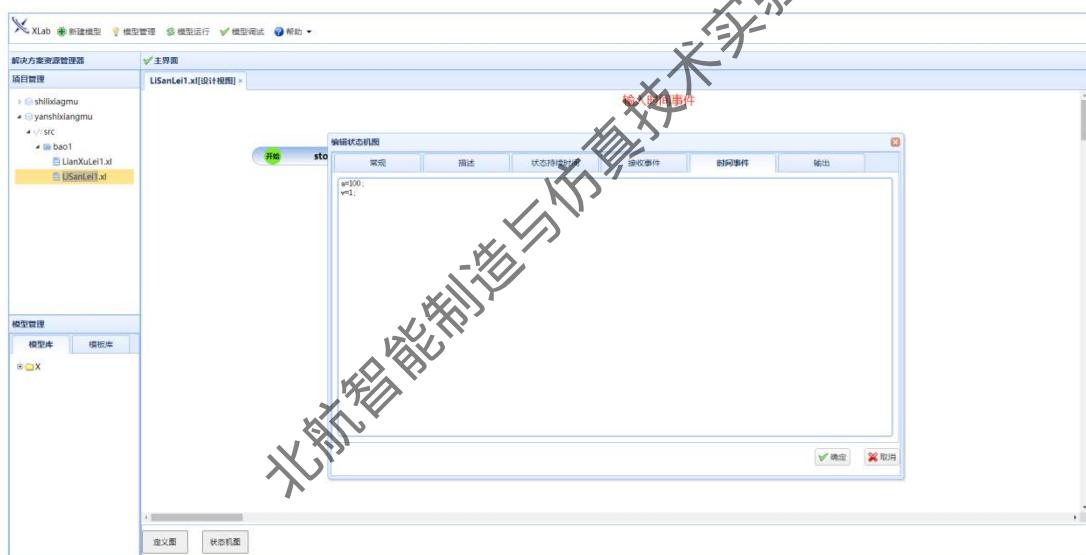
其他绩效评价材料



其他绩效评价材料

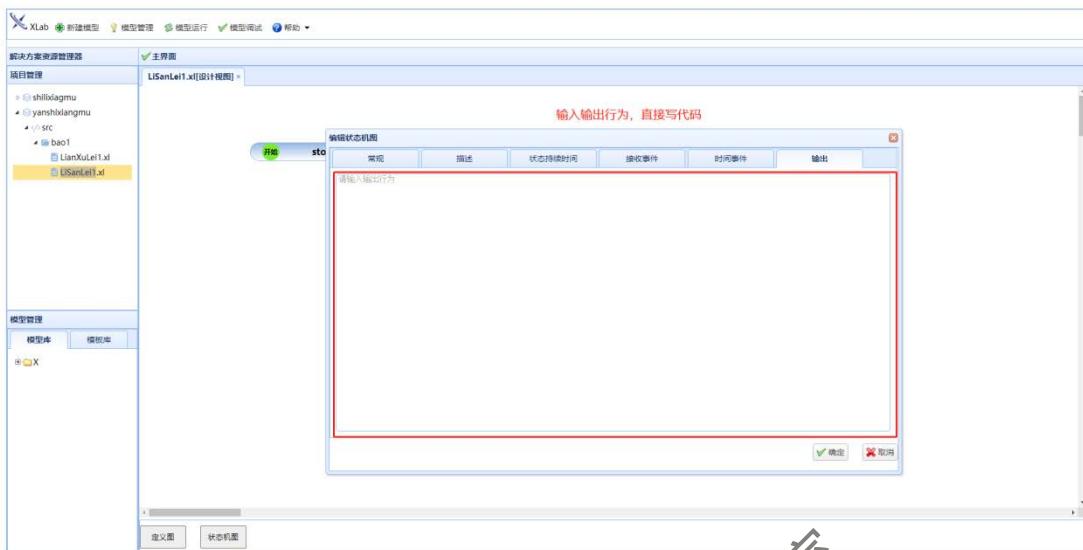


1.22. 编辑状态机图-时间事件

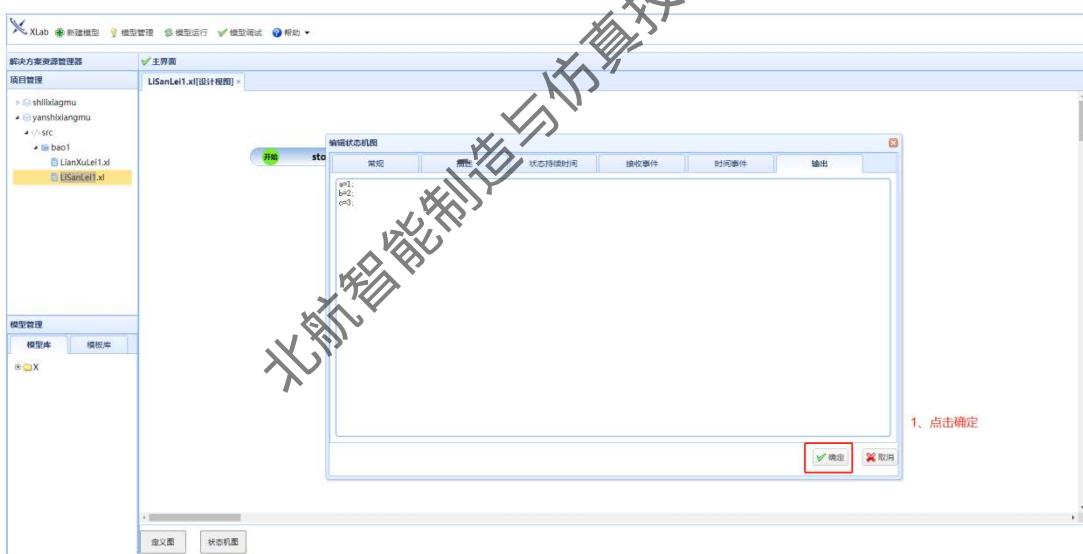


其他绩效评价材料

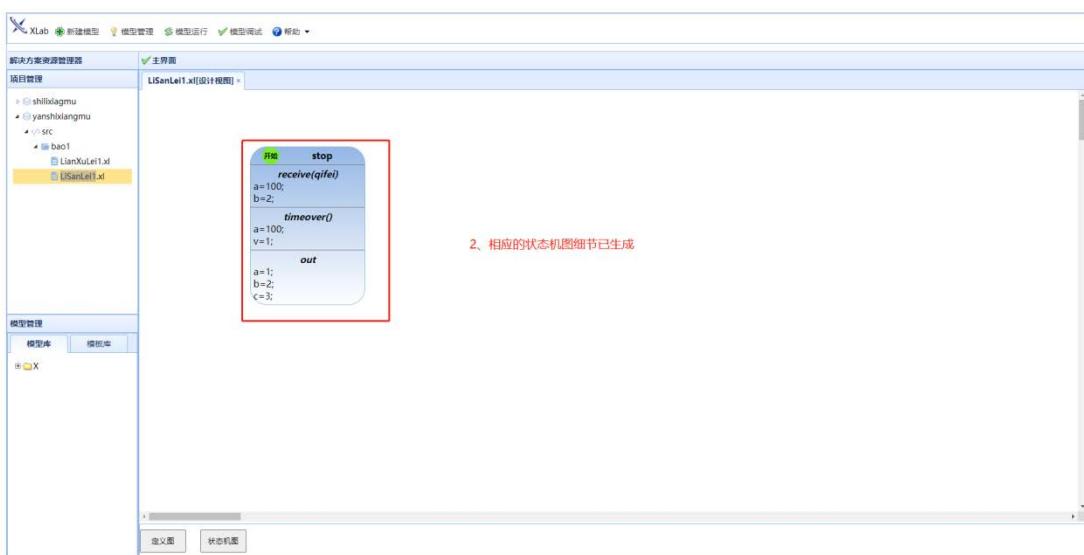
1.23. 编辑状态机图-输出



1.24. 编辑状态机图-确定



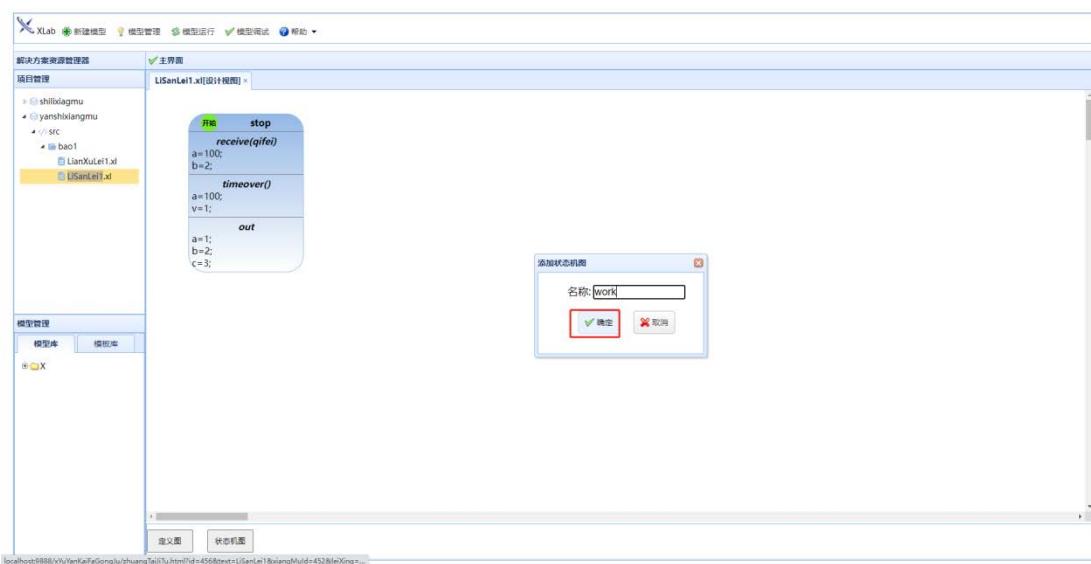
其他绩效评价材料



1.25. 离散类-添加普通状态



其他绩效评价材料

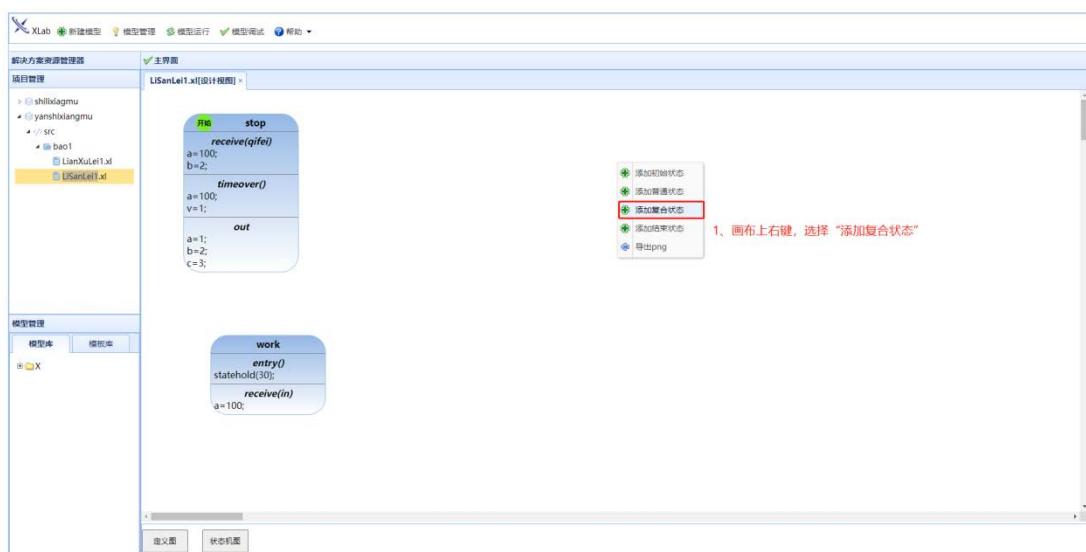


其他绩效评价材料

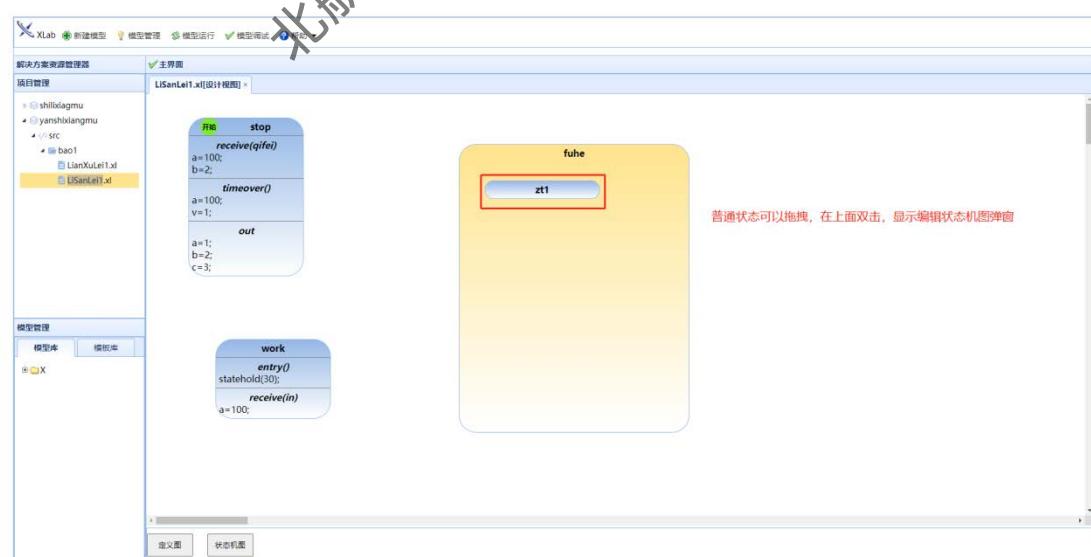
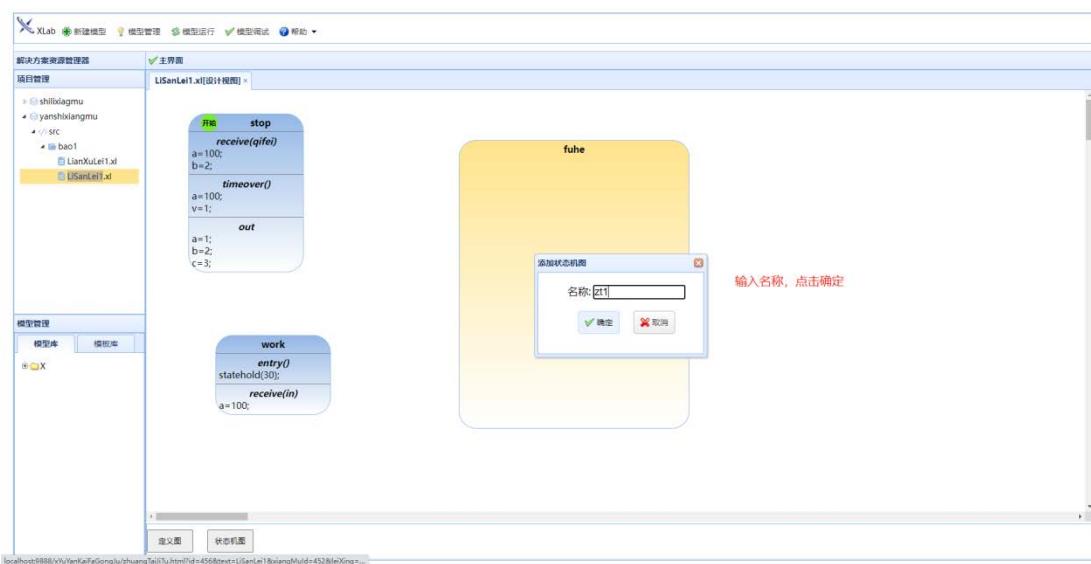


1.26. 离散类-添加复合状态

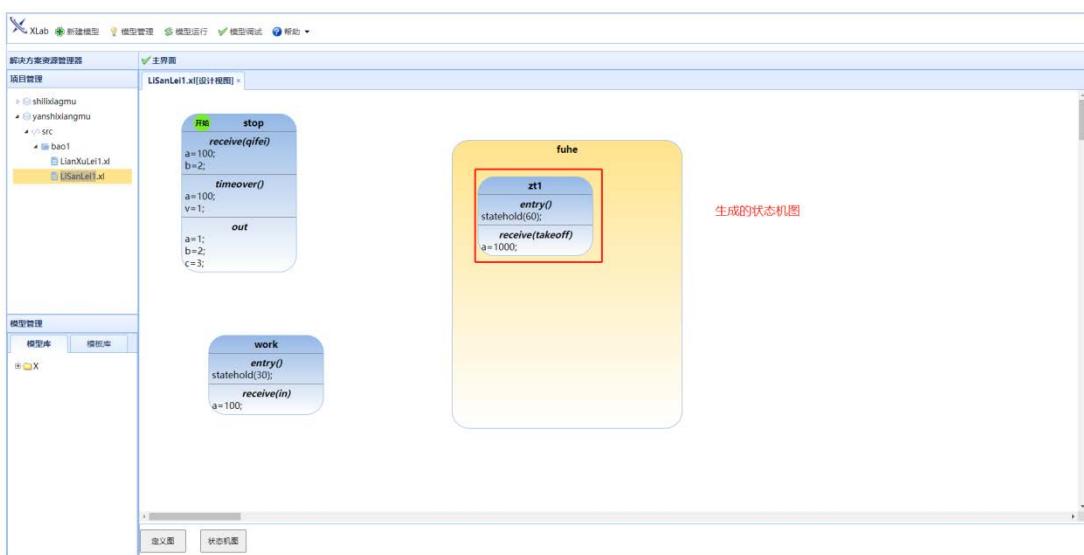
其他绩效评价材料



其他绩效评价材料



其他绩效评价材料

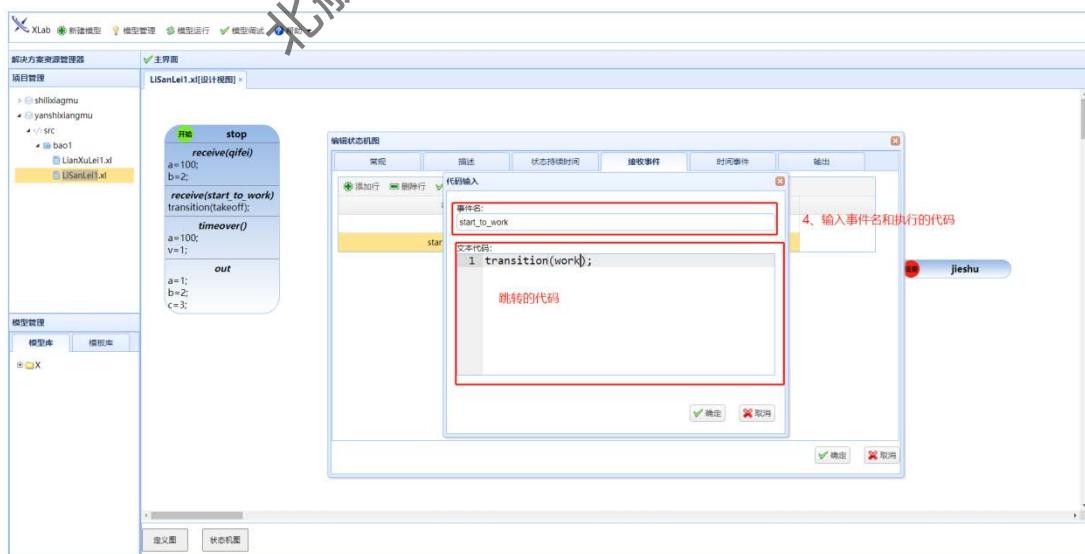
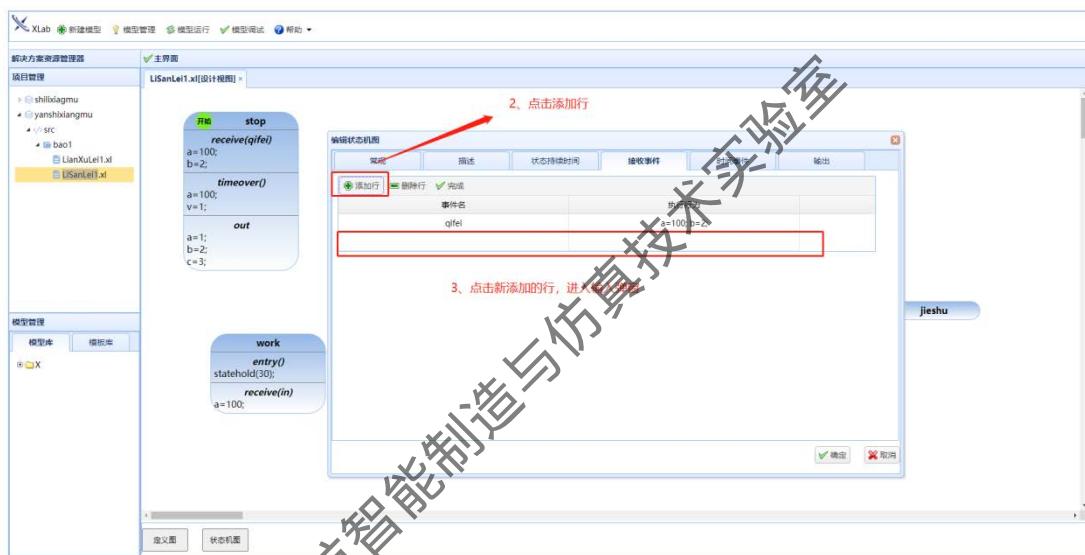
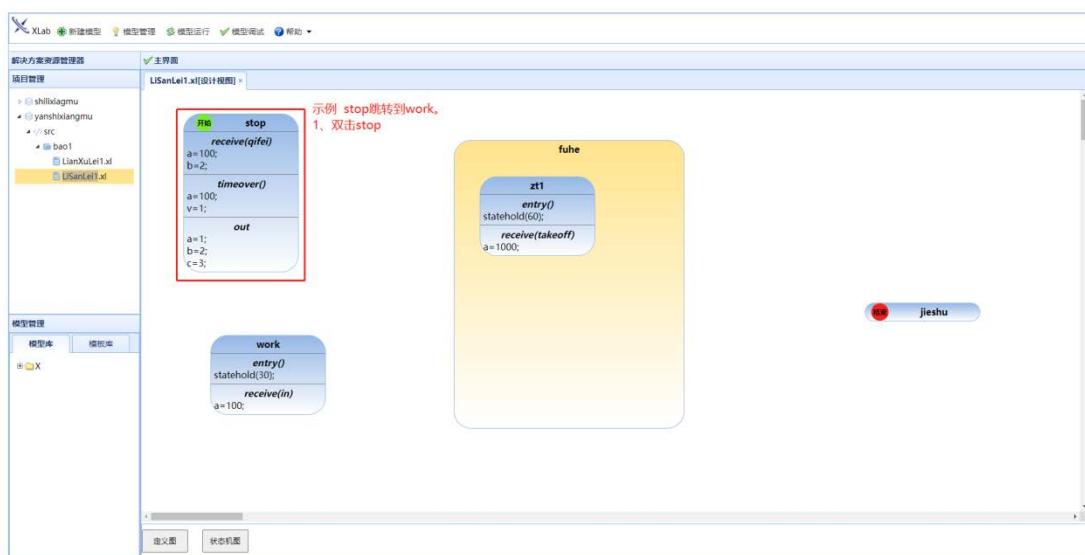


1.27. 离散类-添加结束状态

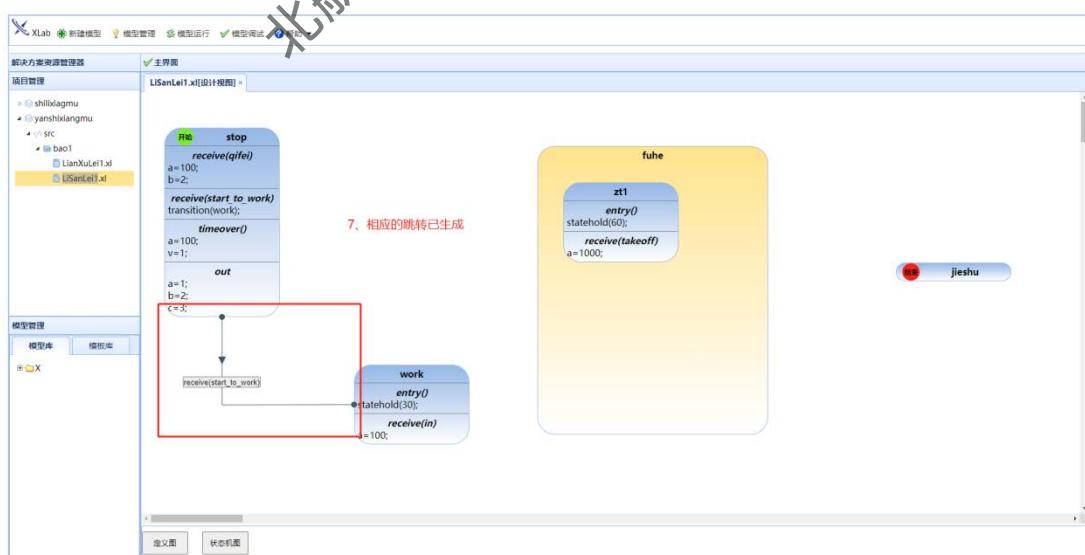
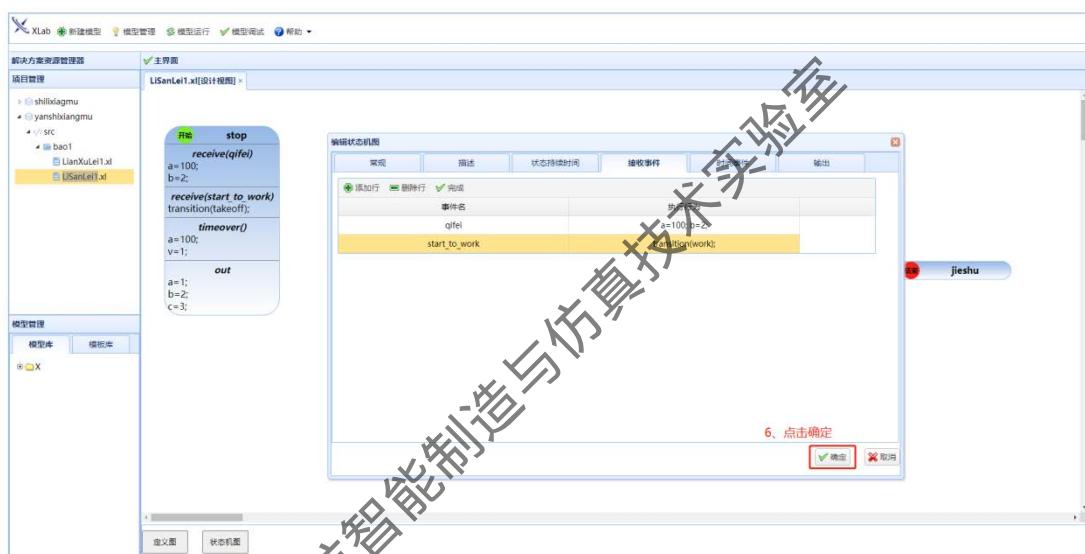
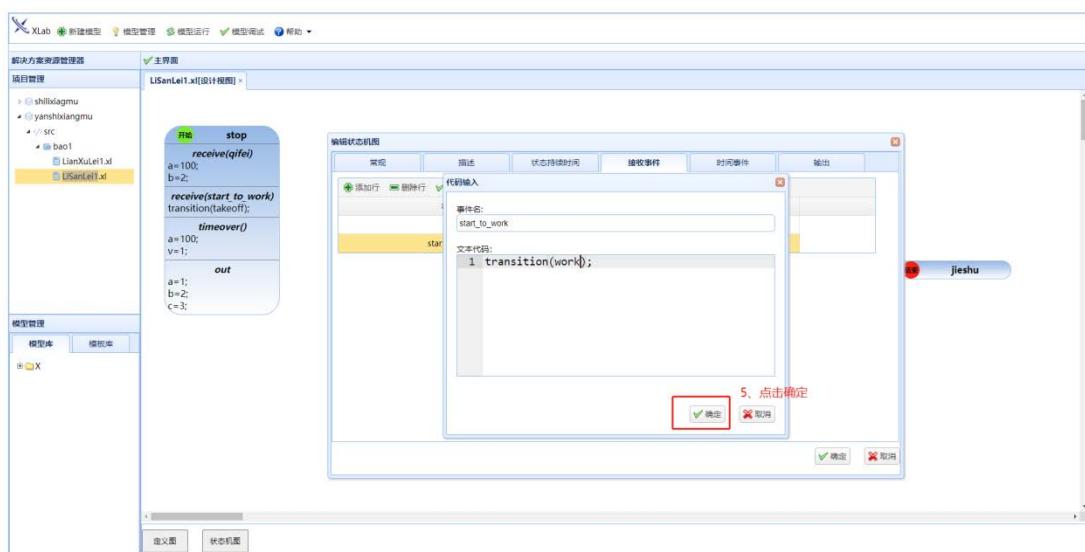


1.28. 离散类-状态机图之前跳转

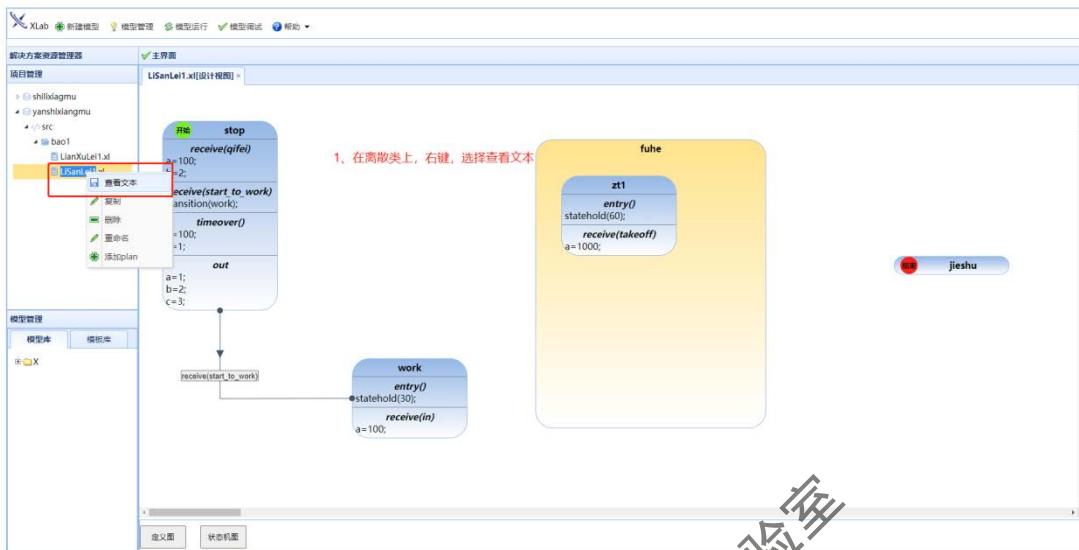
其他绩效评价材料



其他绩效评价材料



1.29. 离散类-查看文本（图形转文本）



```

1 discrete LiSanLei
2 state:
3 initial state stop
4 when receive(qifel) then
5   a=100;
6   b=2;
7 end;
8 when receive(start_to_work) then
9   transition(work);
10 end;
11 when timeover() then
12   a=100;
13   v=1;
14 end;
15 end;
16 state work
17 when entry() then
18   statehold(30);
19 end;
20 when receive(in) then
21   a=100;
22 end;
23 end;
24 state fuhe
25 state zt1
26 when entry() then
27   statehold(60);
28 end;
29 when receive(takeoff) then
30   a=1000;
31 end;
32 end;
33 end;

```

文本代码已生成

1.30. 离散类-保存文本（文本转图形）

The screenshots illustrate the workflow for saving a discrete event simulation model:

- 1. 编写离散类文本:** The first screenshot shows the XLab interface with the code editor containing the following text-based discrete event simulation (DES) model:

```

1 discrete LiSanLei1;
2 value;
3   real _level = 0;
4   real stime =#0;
5 port;
6   event input real worktime;
7   event output real level;
8 state:
9   initial state idle
10  when entry() then
11    statehold(infinite);
12  end;
13  when receive(worktime) then
14    stime = worktime;
15    transition(work);
16  end;
17 end;
18
19 state work
20  when entry() then
21    statehold(stime);
22  end;
23  when timeover() then
24    out
25    _level = _level + 1;
26    send(level,_level);
27  end;
28 end;
29 end;
30 end;

```

- 2. 点击保存:** The second screenshot shows the same interface after clicking the "保存" (Save) button. The status bar at the bottom right indicates "localhost:9080显示 (保存成功)" (localhost:9080 displayed (Saved successfully)).

- 3. 保存成功:** The third screenshot shows the final state of the interface with the saved model displayed in the main window.

其他绩效评价材料

4. 关掉已开发的该类的设计视图

```
1 discrete LiSanLei1
2 value:
3     real _level = 0;
4     real stime=0;
5 port:
6     event input real worktime;
7     event output real level;
8 state:
9     initial state idle
10    when entry() then
11        statehold(infinite);
12    end;
13    when receive(worktime) then
14        stime = worktime;
15        transition(work);
16    end;
17 end;
18
19 state work
20    when entry() then
21        statehold(stime);
22    end;
23    when timeover() then
24        out
25            _level = _level + 1;
26            send(level,_level);
27        end;
28    end;
29 end;
30 end;
```

5. 双击重新开发设计视图

```
1 discrete LiSanLei1
2 value:
3     real _level = 0;
4     real stime=0;
5 port:
6     event input real worktime;
7     event output real level;
8 state:
9     initial state idle
10    when entry() then
11        statehold(infinite);
12    end;
13    when receive(worktime) then
14        stime = worktime;
15        transition(work);
16    end;
17 end;
18
19 state work
20    when entry() then
21        statehold(stime);
22    end;
23    when timeover() then
24        out
25            _level = _level + 1;
26            send(level,_level);
27        end;
28    end;
29 end;
30 end;
```

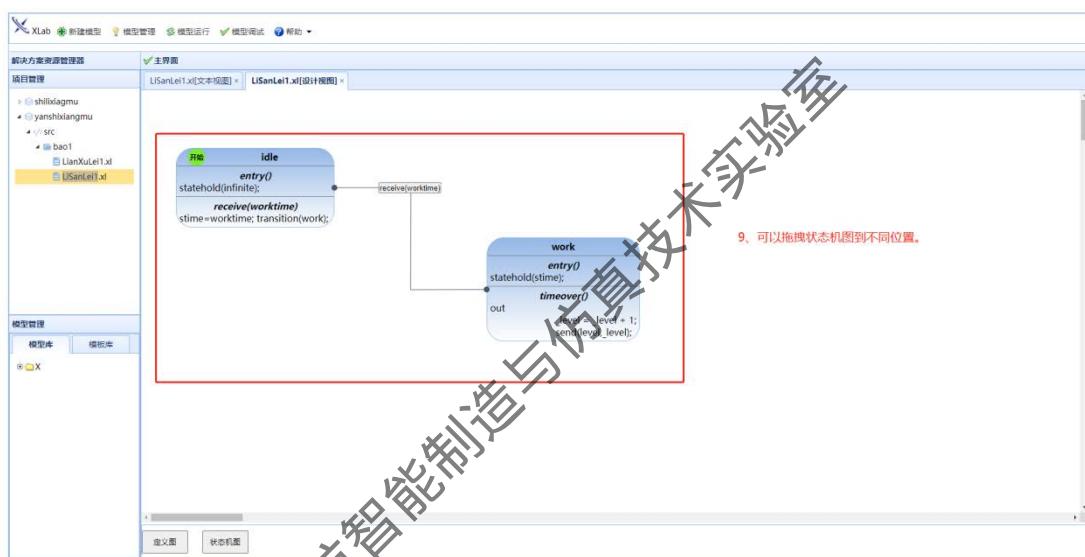
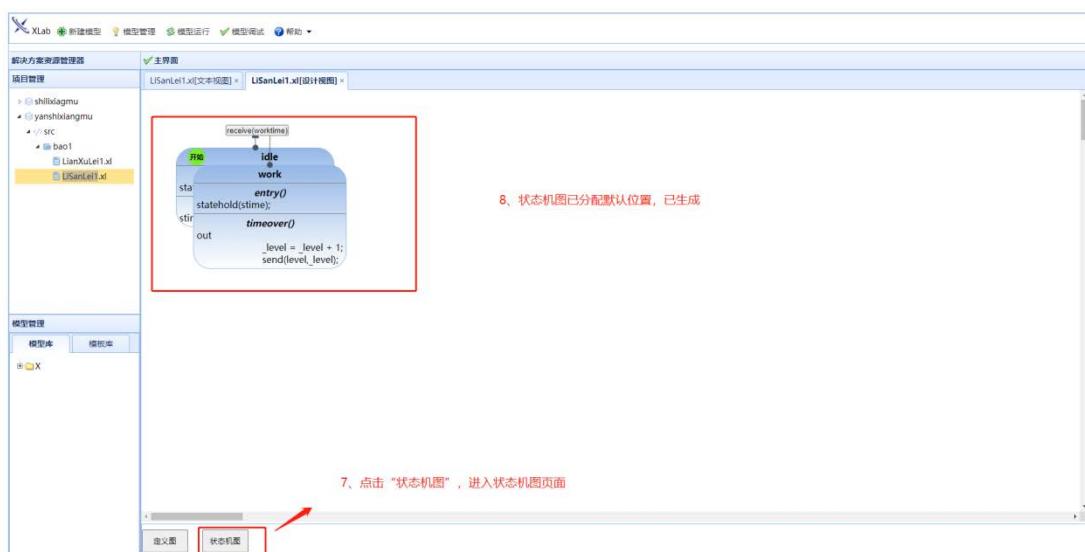
6. 定义图已生成

```
<< discrete >>
LiSanLei1

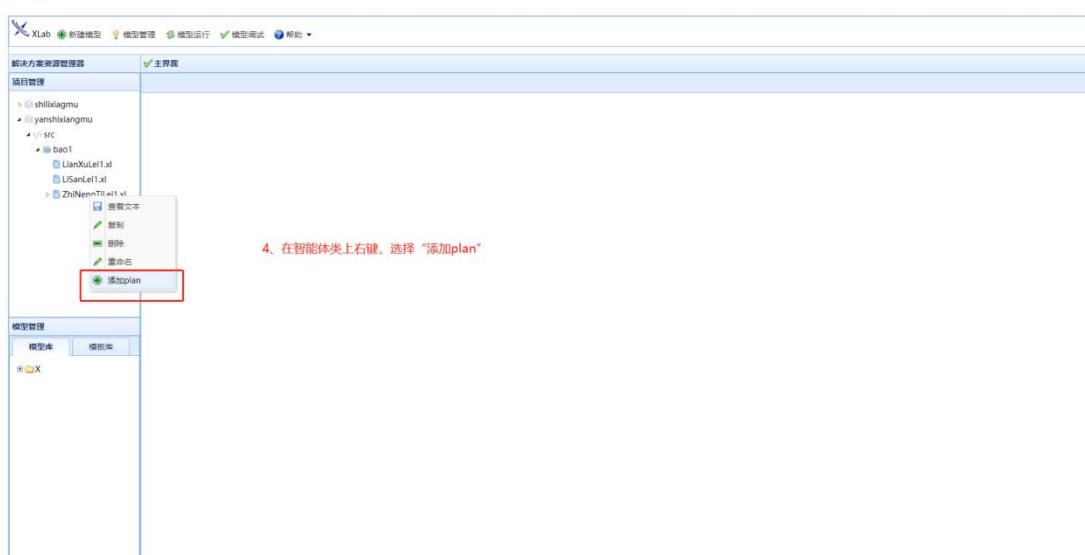
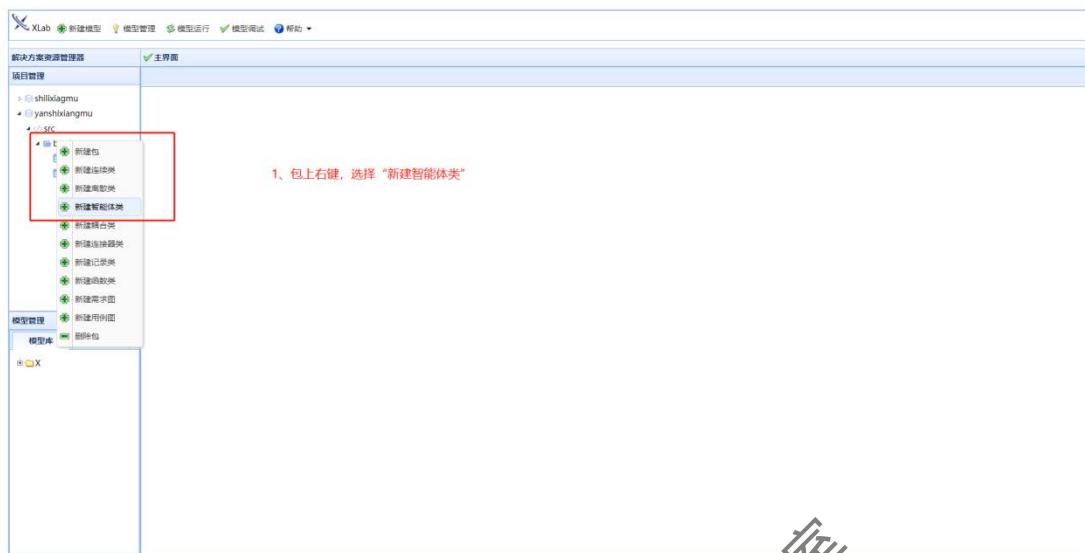
value
real _level=0;
real stime=0;

port
event input worktime;
event output level;
```

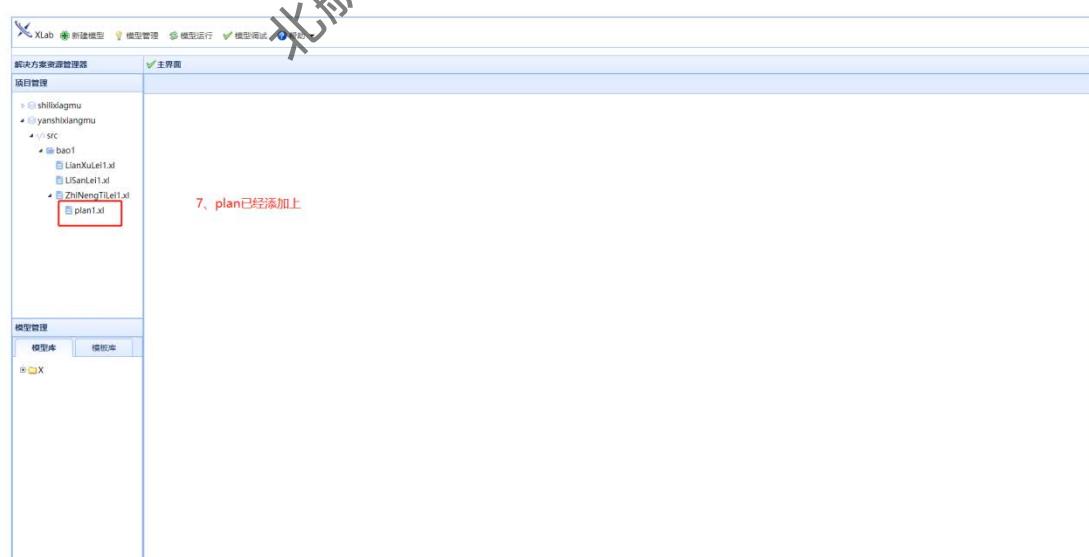
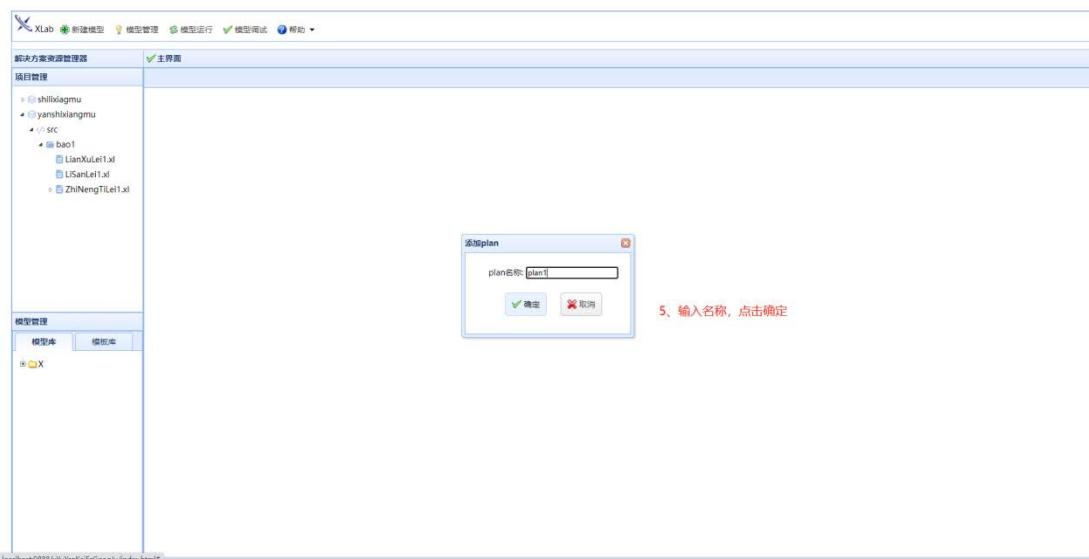
其他绩效评价材料



1.31. 新建智能体类

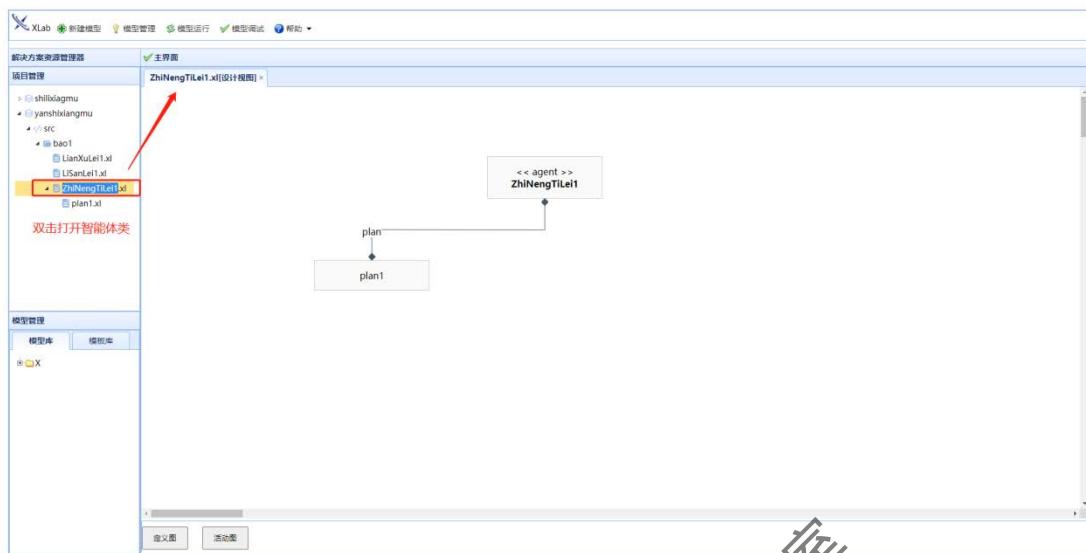


其他绩效评价材料

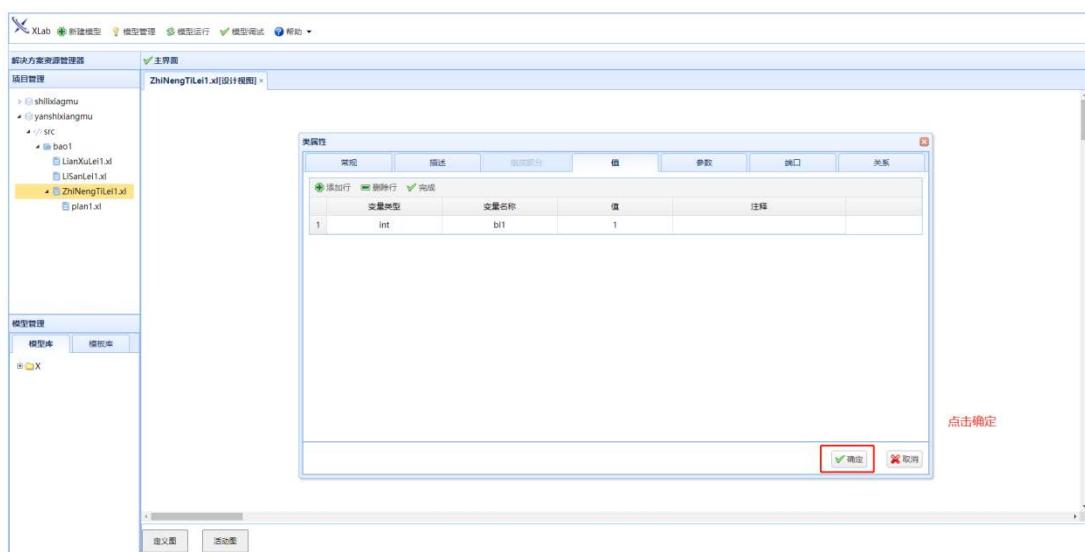


其他绩效评价材料

1.32. 打开智能体类

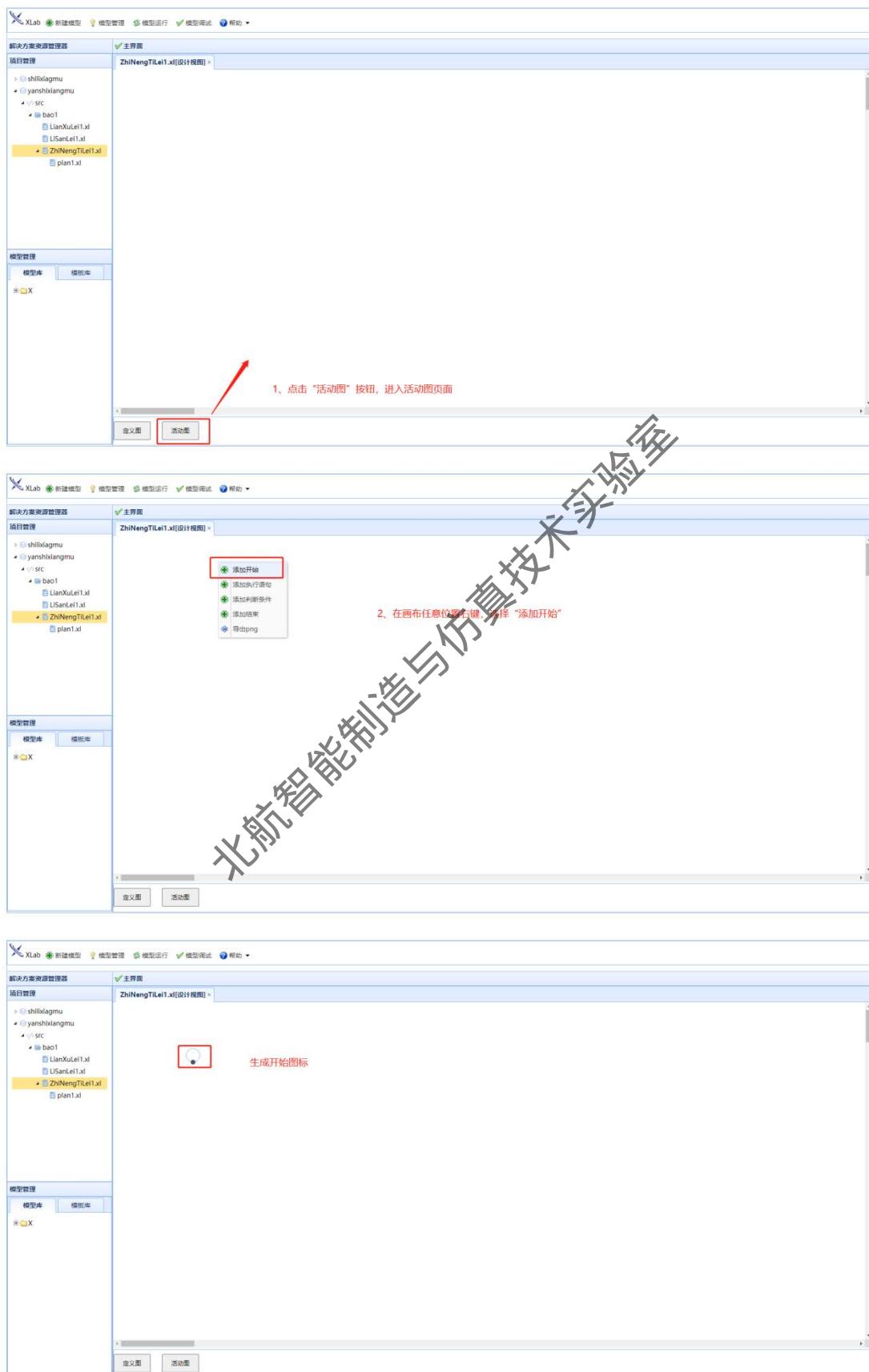


其他绩效评价材料

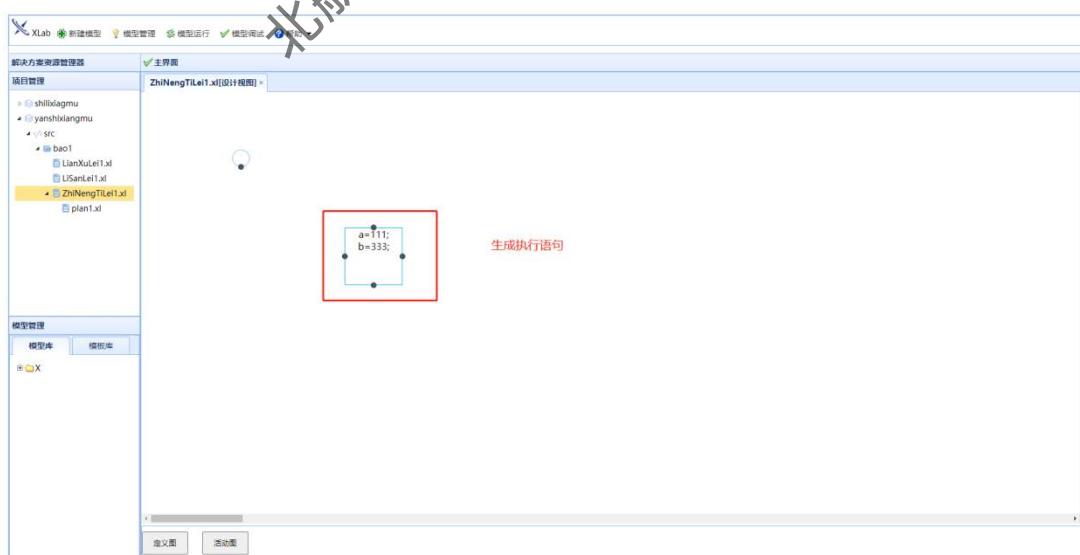
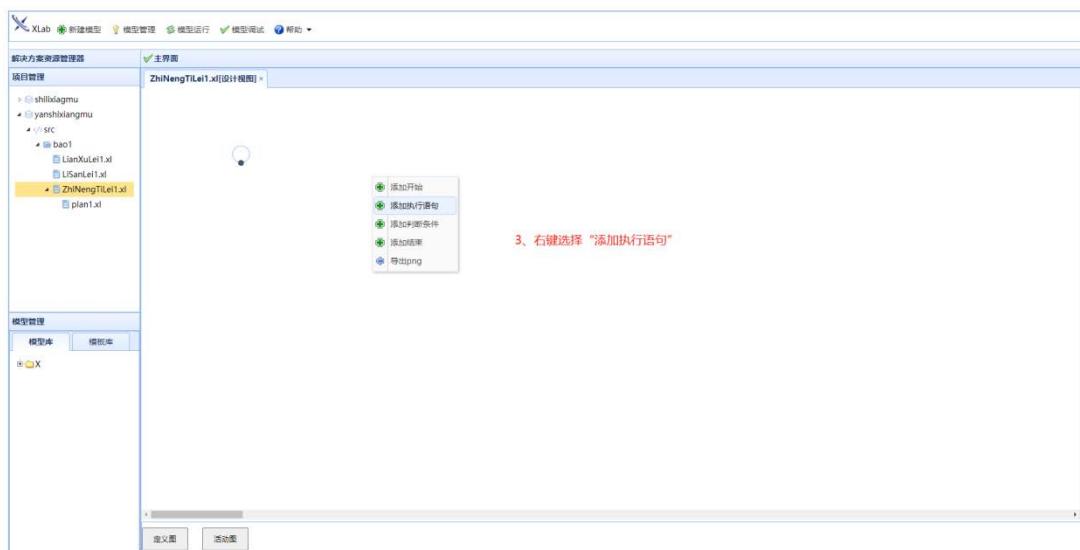


其他绩效评价材料

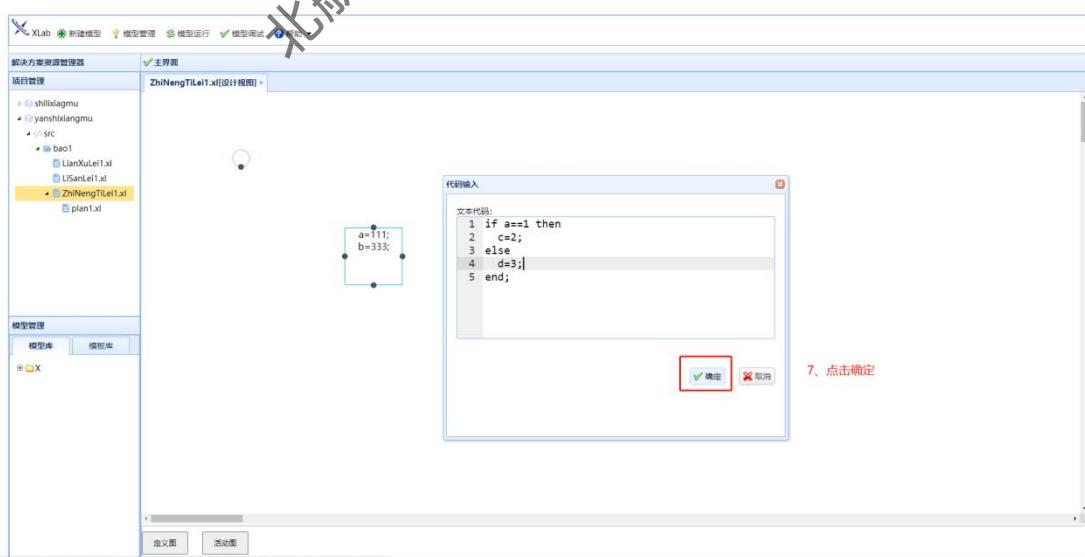
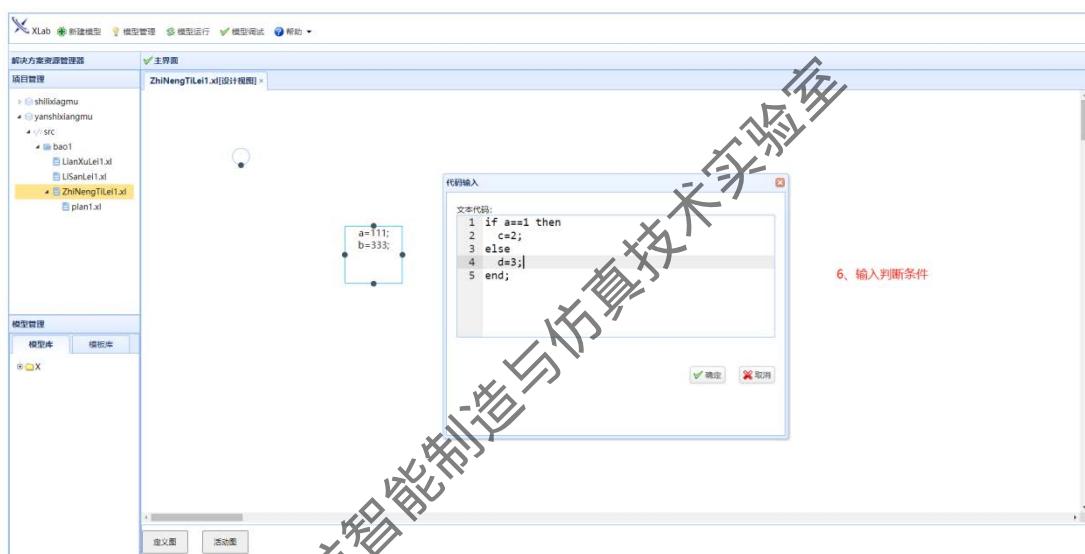
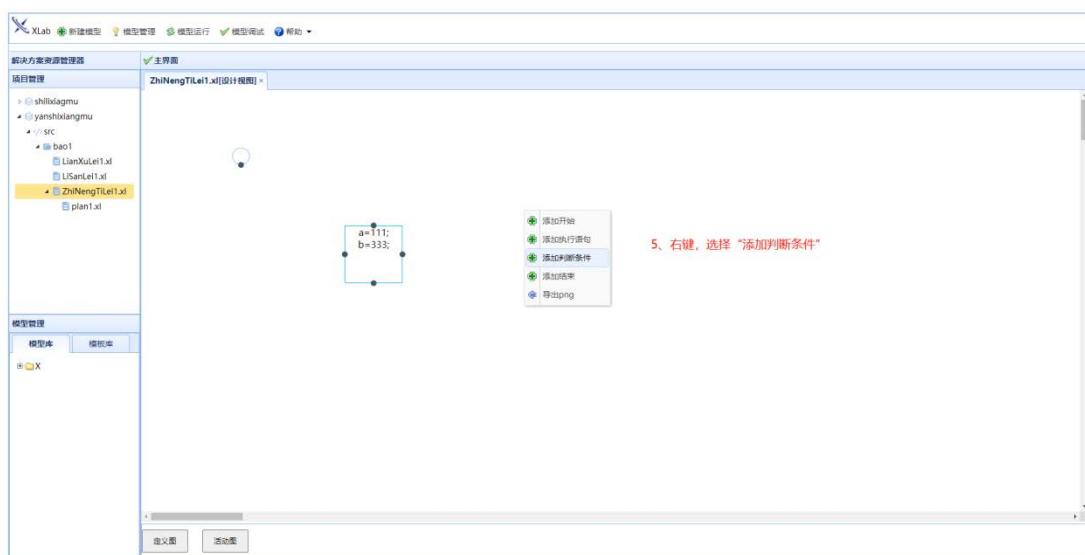
1.33. 智能体类-活动图



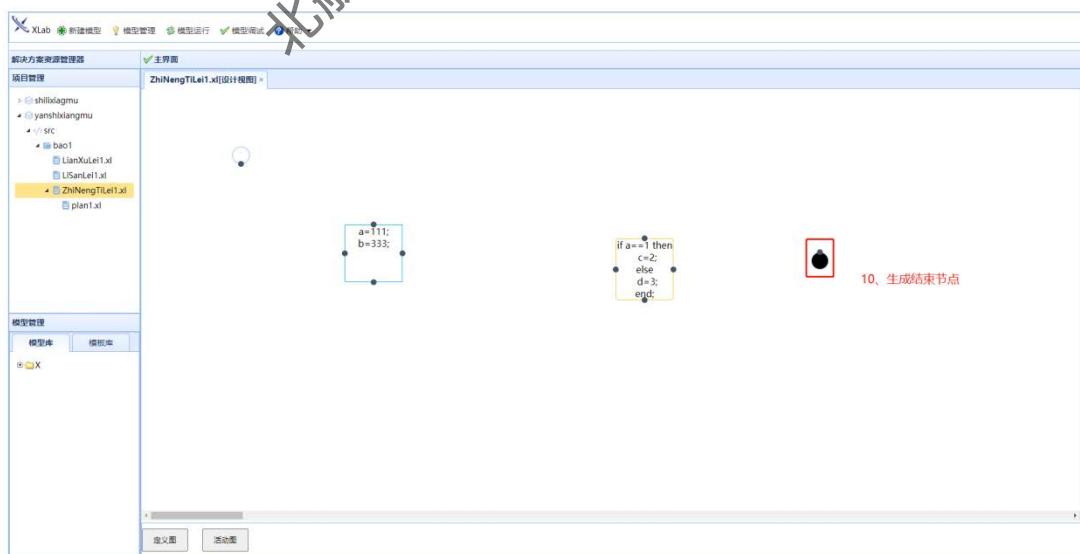
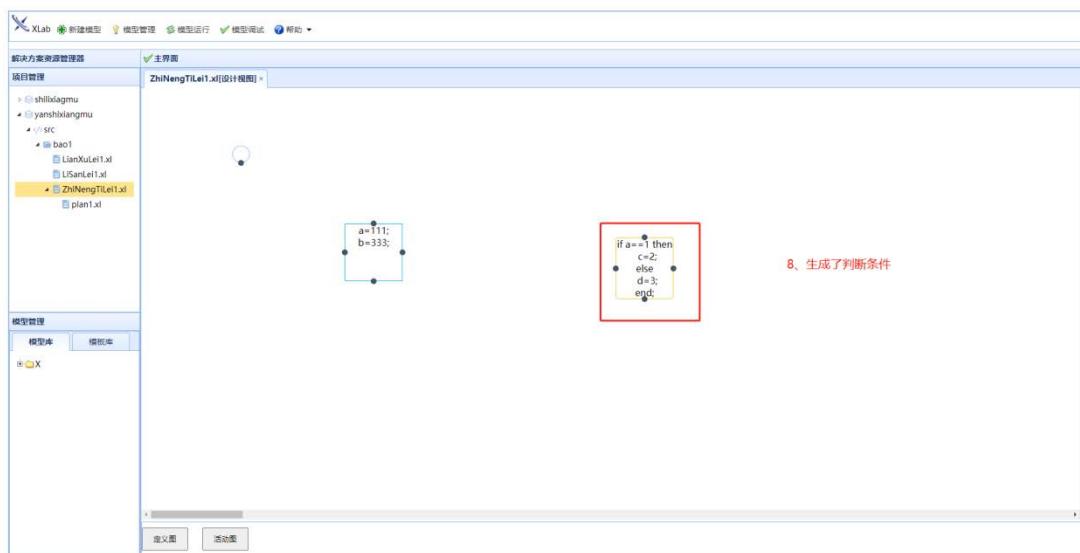
其他绩效评价材料



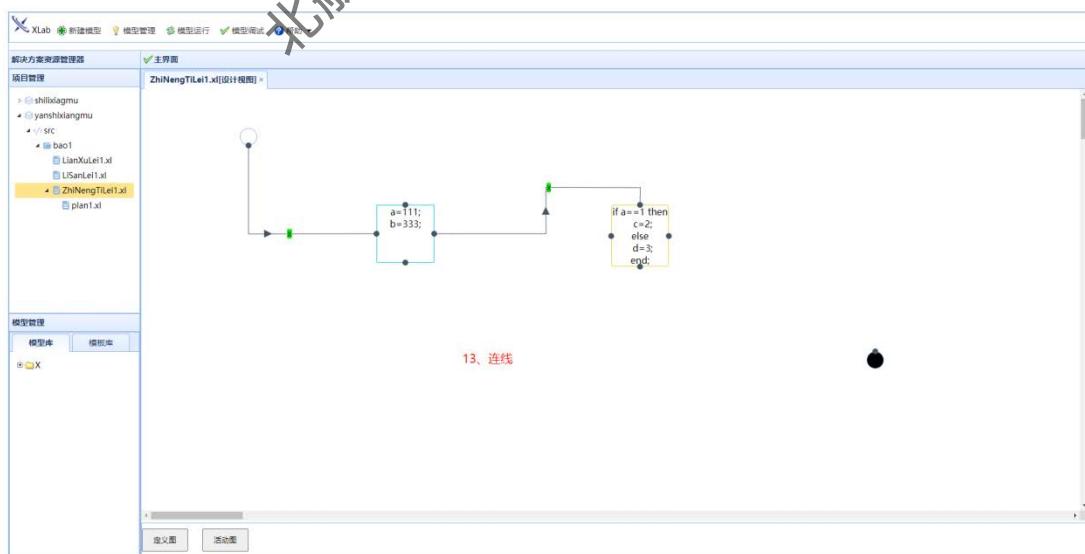
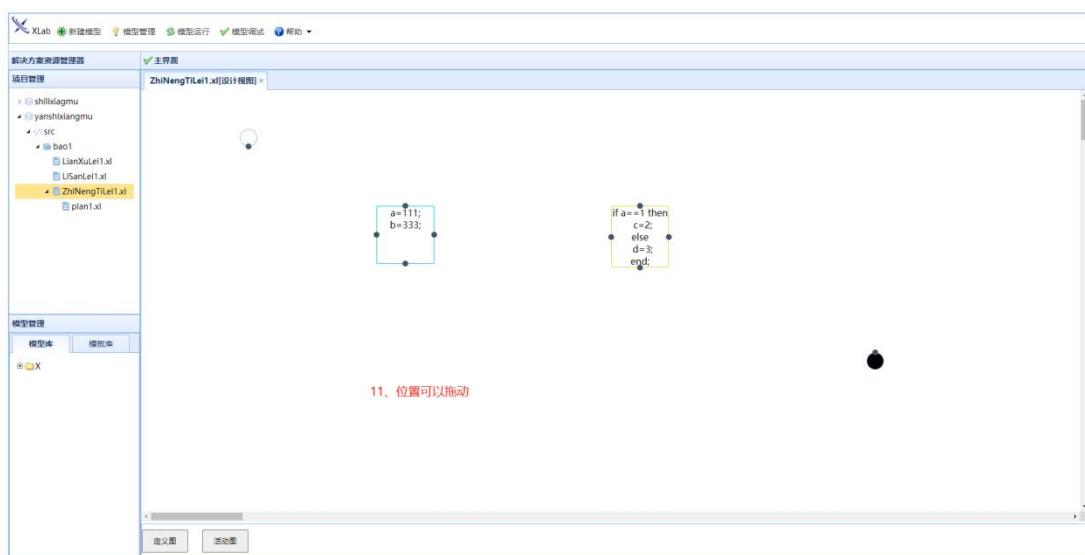
其他绩效评价材料



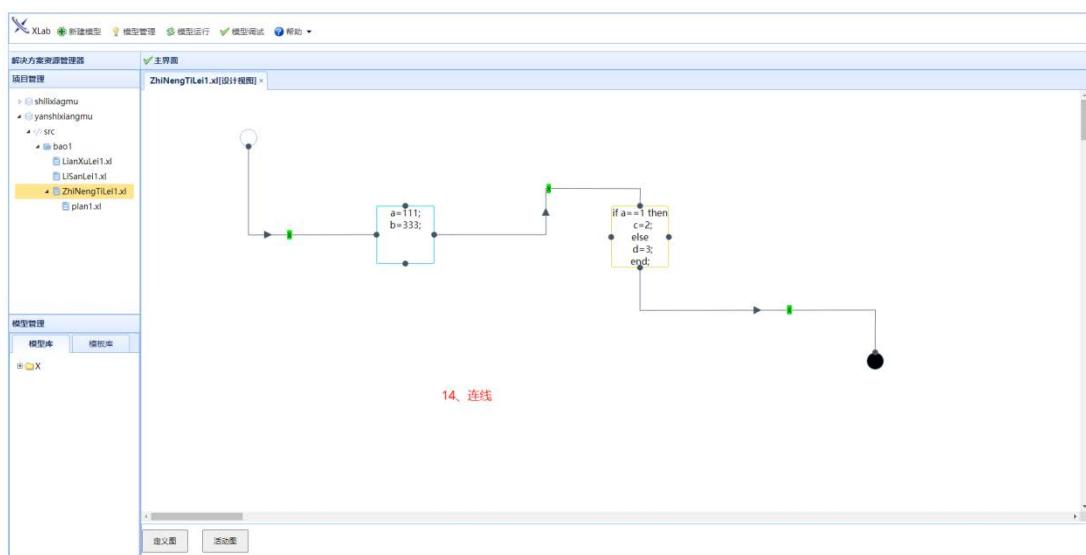
其他绩效评价材料



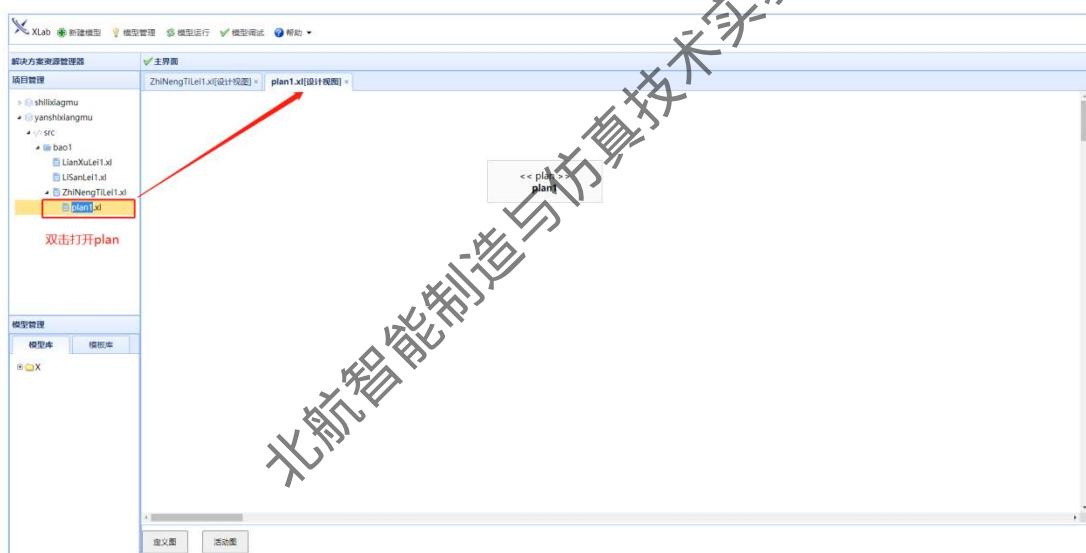
其他绩效评价材料



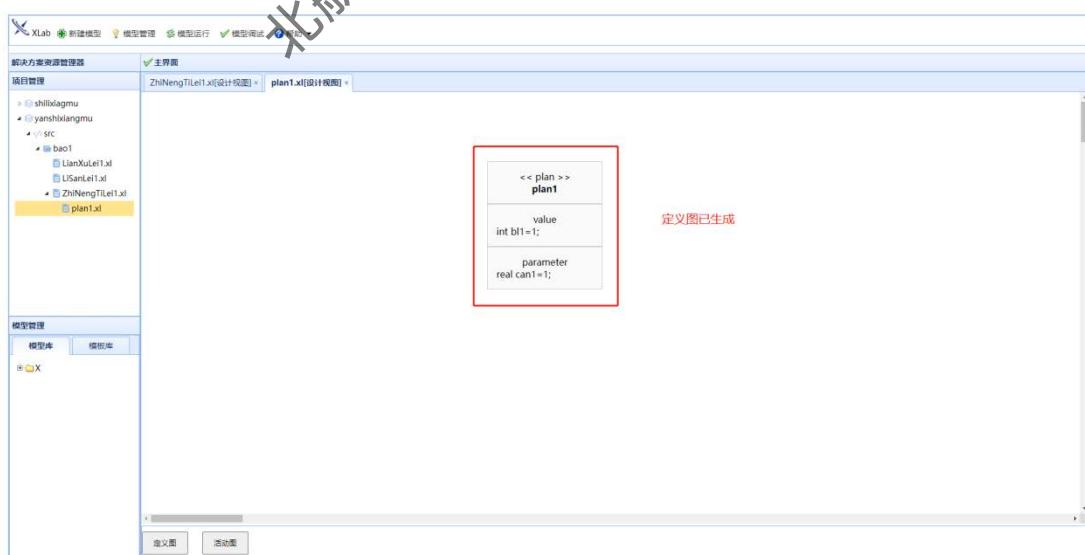
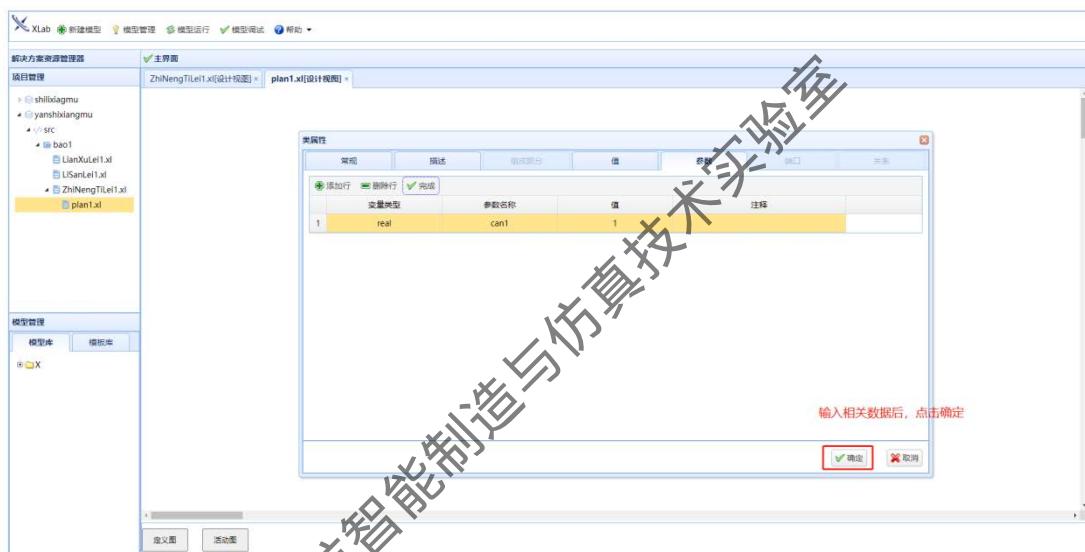
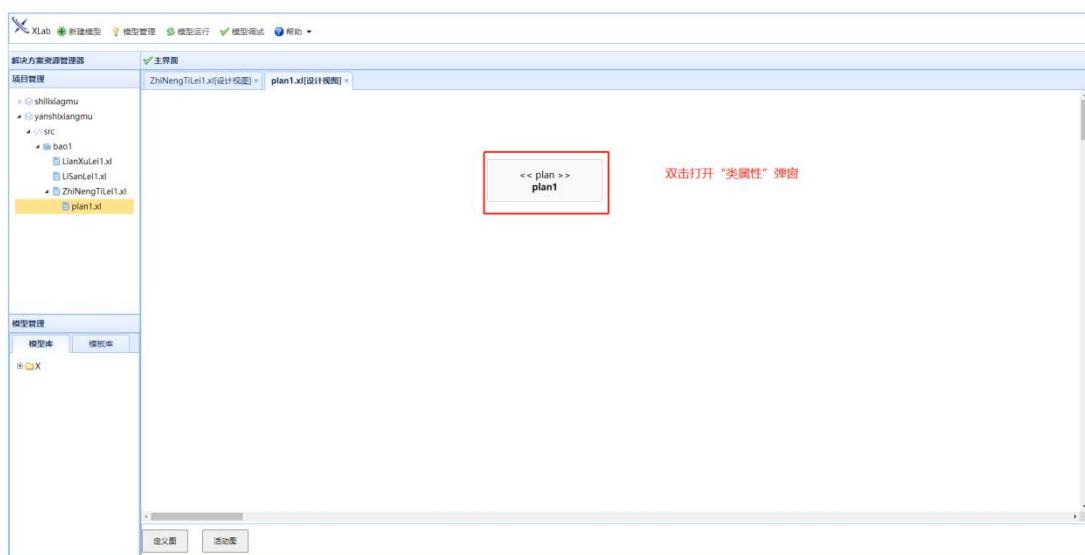
其他绩效评价材料



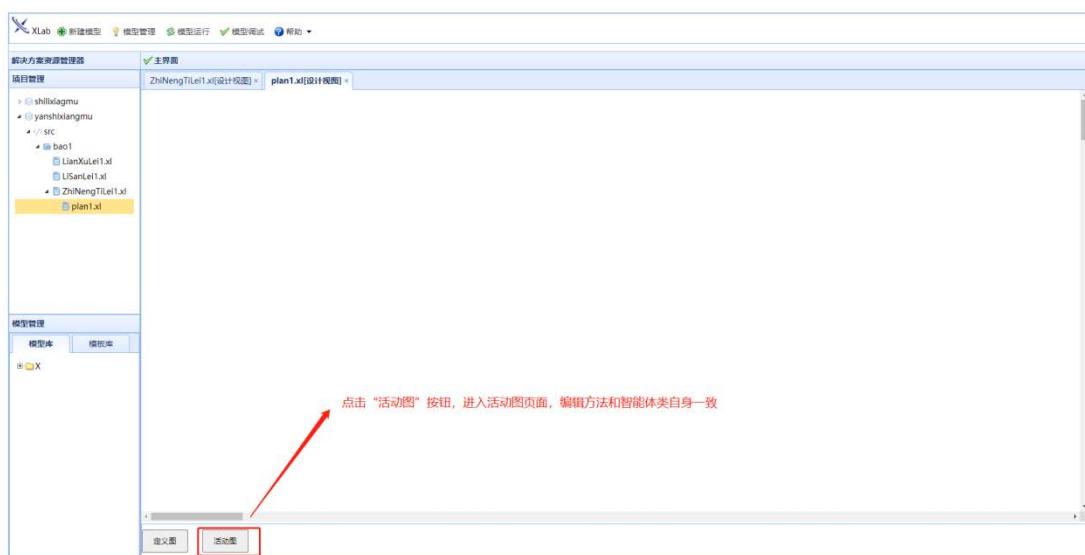
1.34. 智能体类-打开 plan



其他绩效评价材料



其他绩效评价材料



1.35. 智能体类-查看文本（图形到文本）



其他绩效评价材料

The screenshot shows the XLab interface with the 'Main Interface' tab selected. The left sidebar shows project management with files like 'shixiangmu', 'yanshihangmu', 'src', 'bao1', 'LianXulei1.xl', 'LiSanLei1.xl', and 'ZhiNengTiLei1.xl'. The right pane displays a code editor with the following text:

```
1 agent ZhiNengTiLei1
2 value:
3   int bli=1;
4 parameter:
5   int can1=1;
6 port:
7   input int dk1;
8 action:
9
10 a=111;
11 b=333;
12 if a==1 then
13   c=2;
14 else
15   d=3;
16 end;
17
18 plan plan1
19 value:
20   int bli=1;
21 parameter:
22   real can1=1;
23 end;
24 end;
```

A red annotation '2. 文本已生成' (Text generated) is placed next to the code.

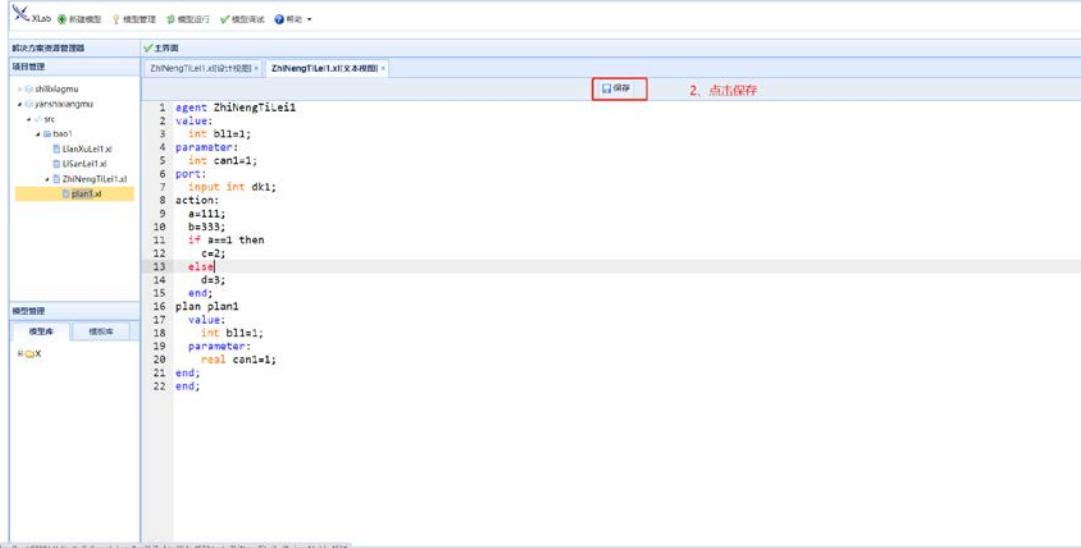
1.36. 智能体类-文本保存（文本到图形）

The screenshot shows the XLab interface with the 'Main Interface' tab selected. The left sidebar shows project management with files like 'shixiangmu', 'yanshihangmu', 'src', 'bao1', 'LianXulei1.xl', 'LiSanLei1.xl', and 'ZhiNengTiLei1.xl'. The right pane displays a code editor with the following text:

```
1 agent ZhiNengTiLei1
2 value:
3   int bli=1;
4 parameter:
5   int can1=1;
6 port:
7   input int dk1;
8 action:
9   a=111;
10  b=333;
11  if a==1 then
12    c=2;
13  else
14    d=3;
15  end;
16 plan plan1
17 value:
18   int bli=1;
19 parameter:
20   real can1=1;
21 end;
22 end;
```

A red box highlights the first 16 lines of the code. A red annotation '1. 修改文本' (Modify text) is placed near the bottom right of the highlighted area.

其他绩效评价材料



XLab 新建模型 模型管理 模型运行 模型调试 帮助

解决方案资源管理器

项目管理 ZhiNengTiLei1.xl[设计视图] ZhiNengTiLei1.xl[文本视图]

2. 点击保存

```
1 agent ZhiNengTiLei1
2 value;
3 int b1=1;
4 parameter;
5 int can1=1;
6 port;
7 input int dk1;
8 action;
9 a=111;
10 b=333;
11 if a==1 then
12 c=2;
13 else
14 d=3;
15 end;
16 plan plan1
17 value;
18 int b1=1;
19 parameter;
20 real can1=1;
21 end;
22 end;
```



XLab 新建模型 模型管理 模型运行 模型调试 帮助

解决方案资源管理器

项目管理 ZhiNengTiLei1.xl[设计视图] ZhiNengTiLei1.xl[文本视图]

localhost:9888 显示 3. 保存成功

```
1 agent ZhiNengTiLei1
2 value;
3 int b1=1;
4 parameter;
5 int can1=1;
6 port;
7 input int dk1;
8 action;
9 a=111;
10 b=333;
11 if a==1 then
12 c=2;
13 else
14 d=3;
15 end;
16 plan plan1
17 value;
18 int b1=1;
19 parameter;
20 real can1=1;
21 end;
22 end;
```



XLab 新建模型 模型管理 模型运行 模型调试 帮助

解决方案资源管理器

项目管理 ZhiNengTiLei1.xl[设计视图] ZhiNengTiLei1.xl[文本视图]

4. 点回设计视图, 定义图已生成

```
< < agent > >
ZhiNengTiLei1

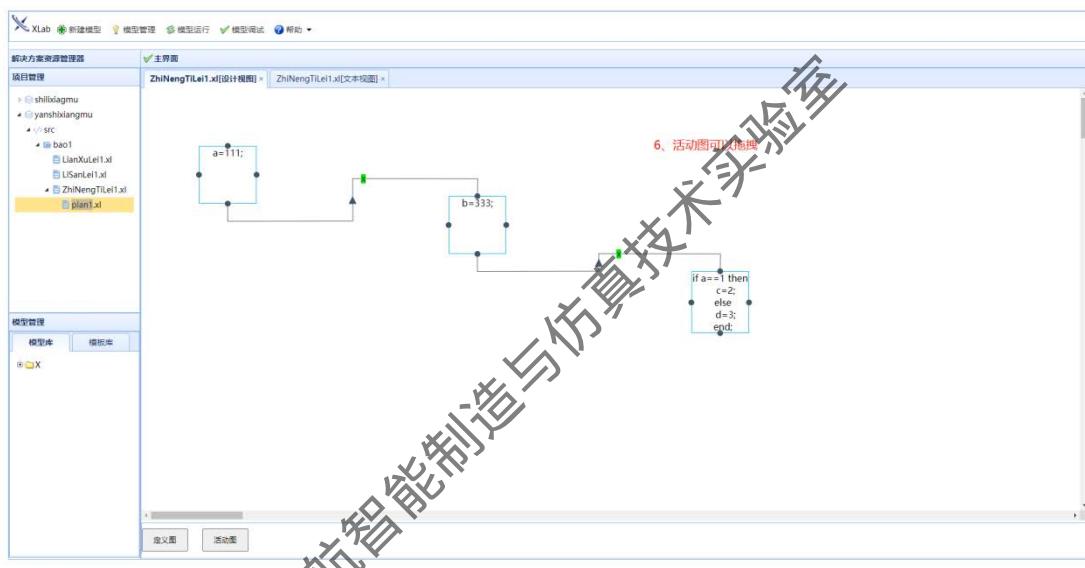
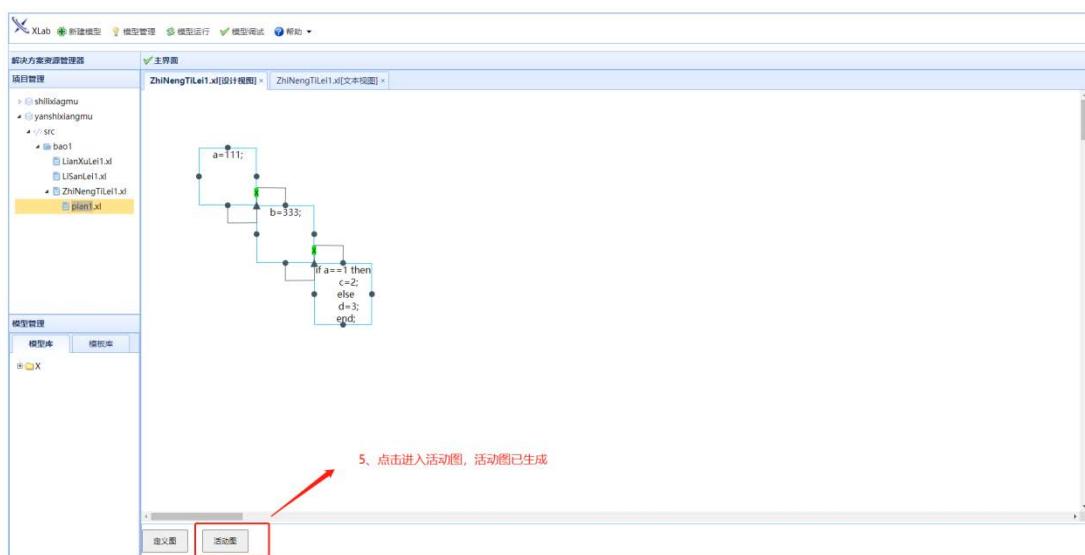
value
int b1=1;

parameter
int can1=1;

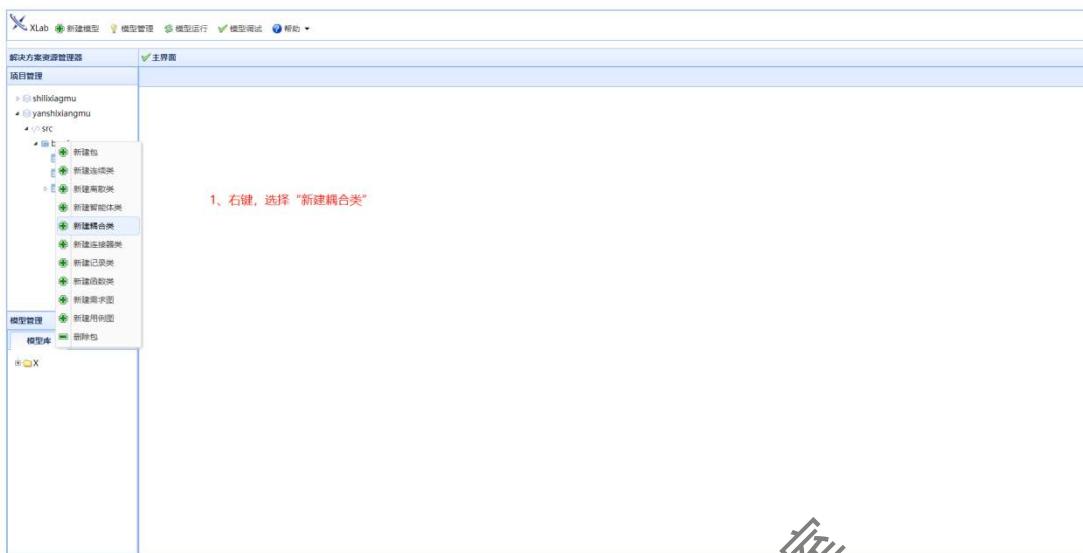
port
input dk1;
```

定义图 动态图

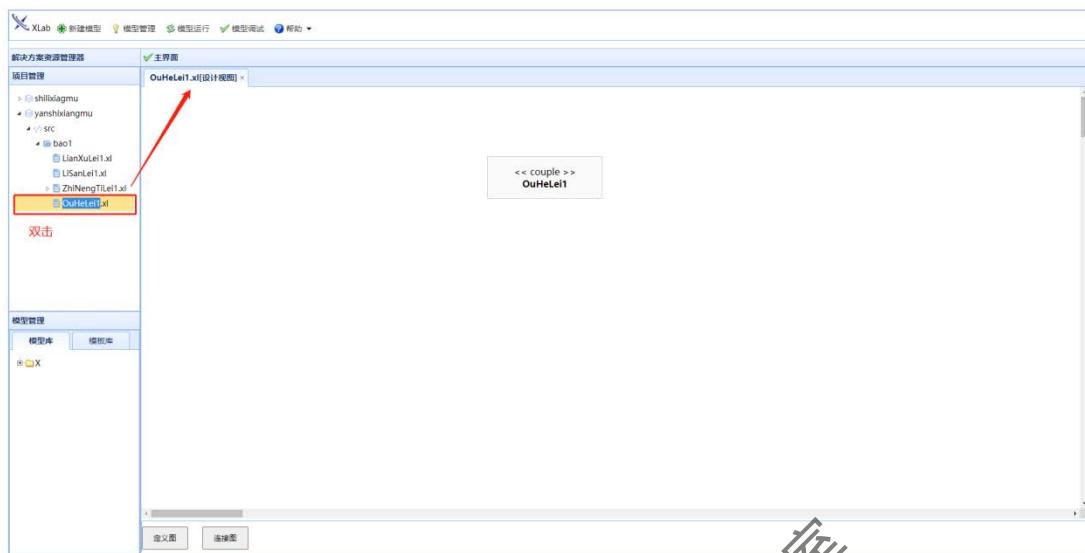
其他绩效评价材料



1.37. 新建耦合类



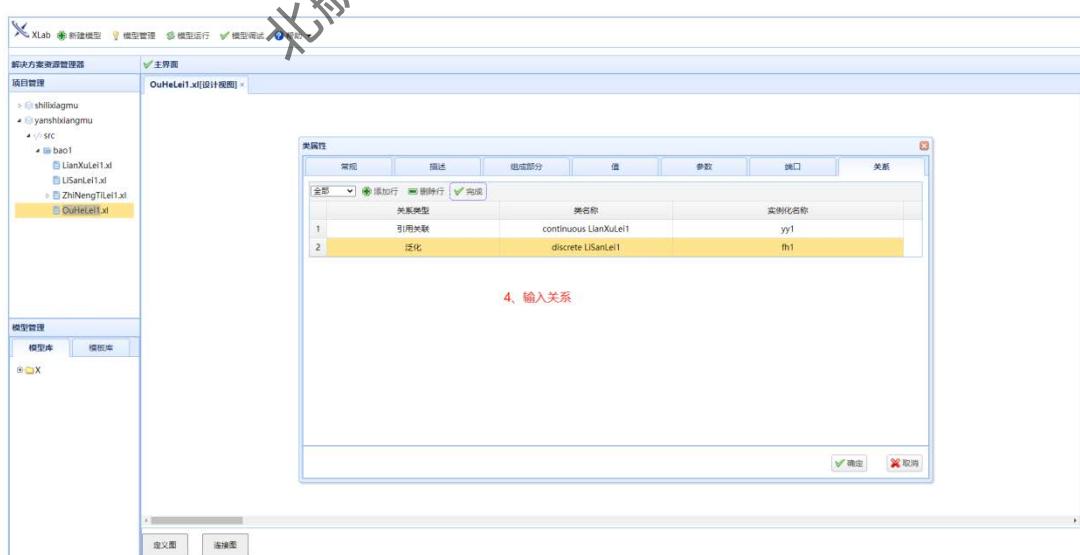
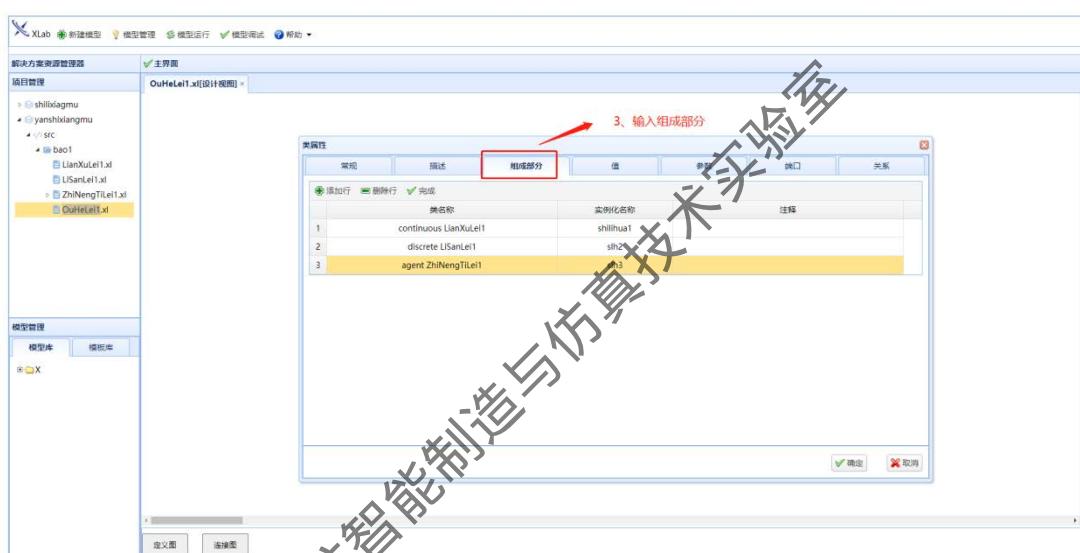
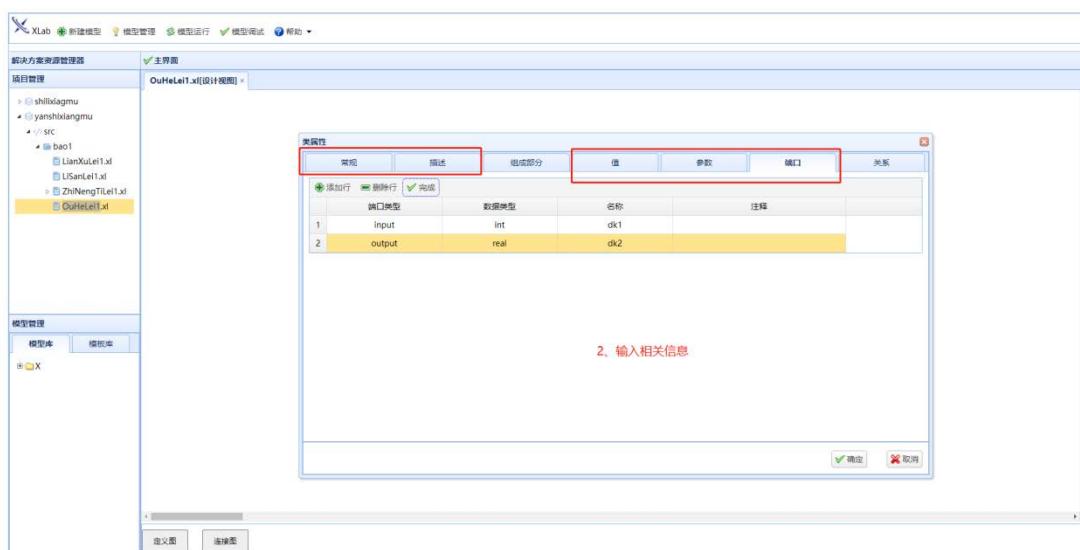
1.38. 打开耦合类



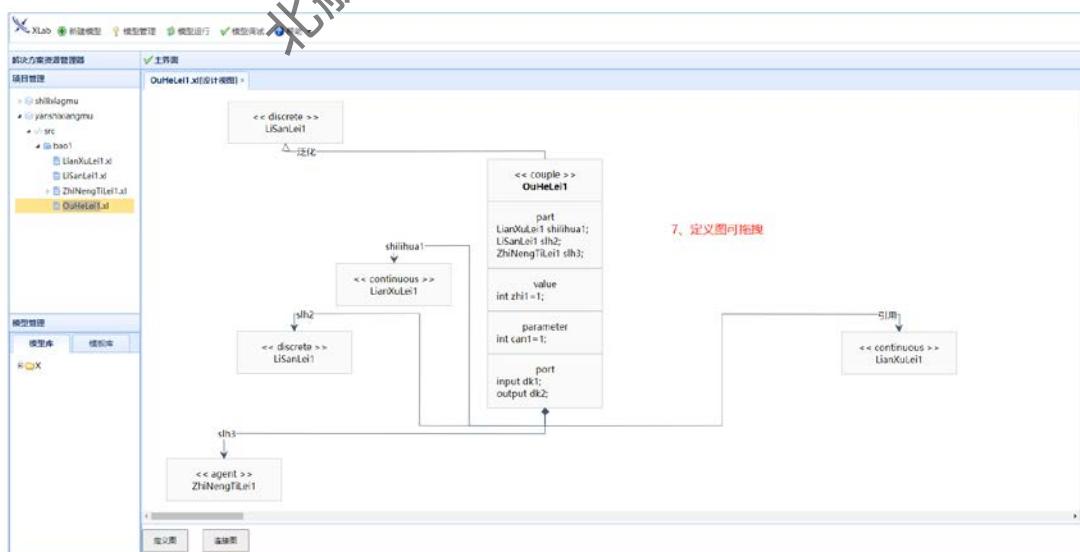
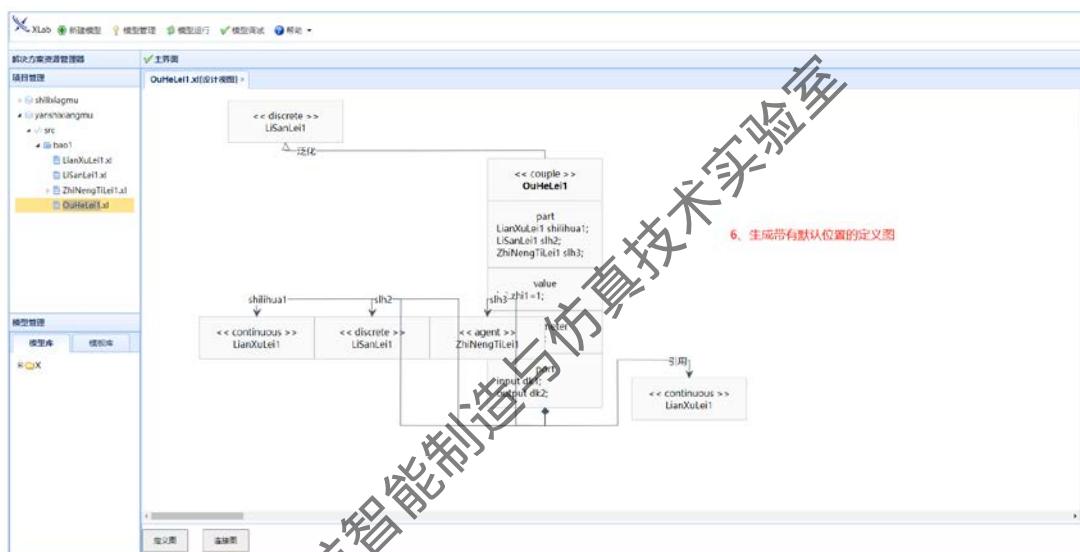
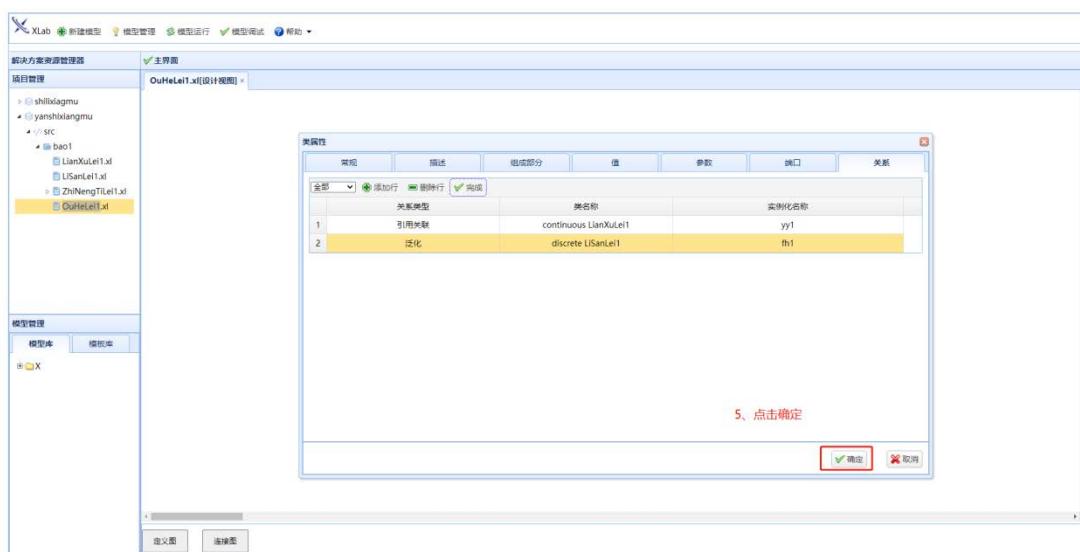
1.39. 耦合类-定义图



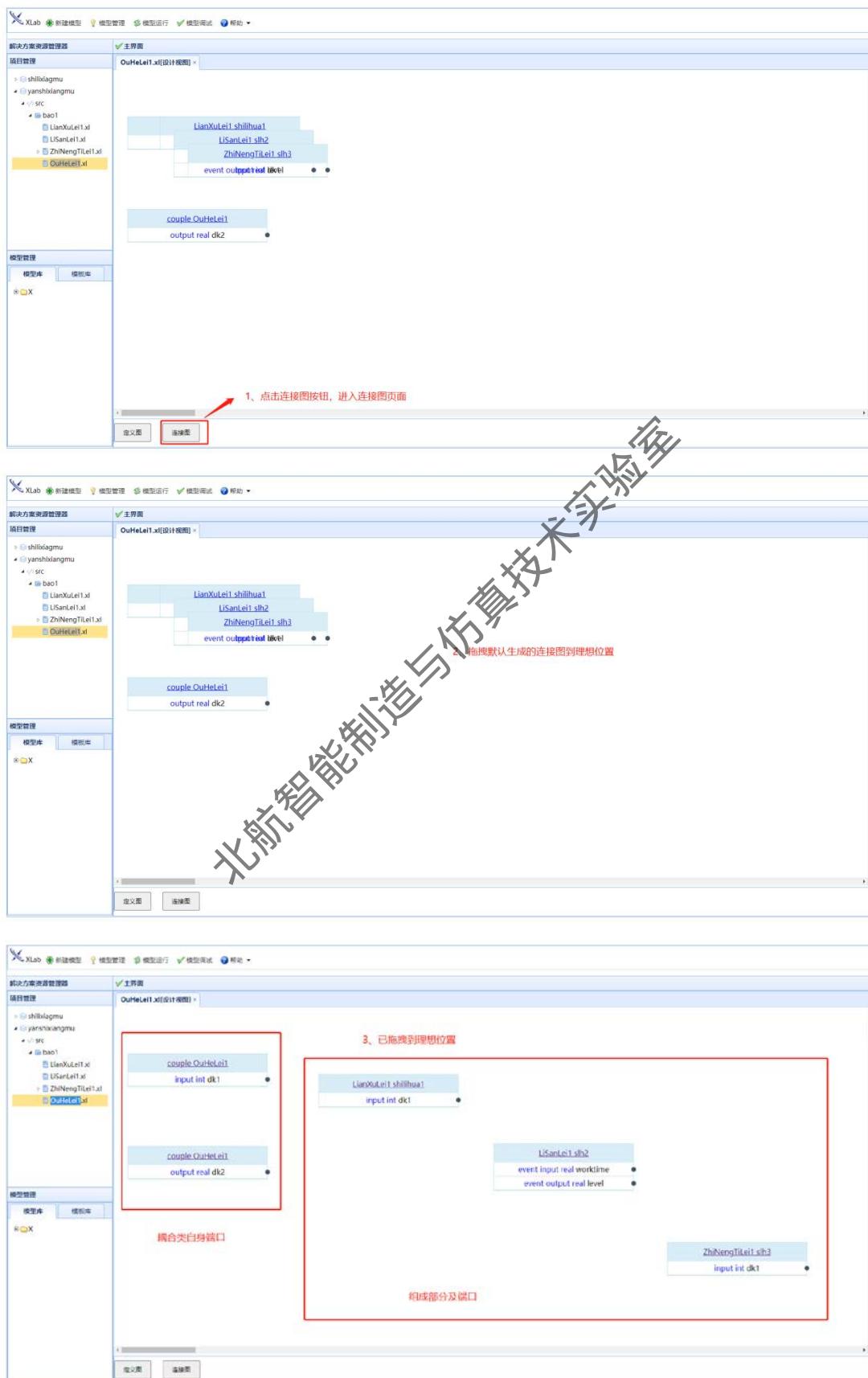
其他绩效评价材料



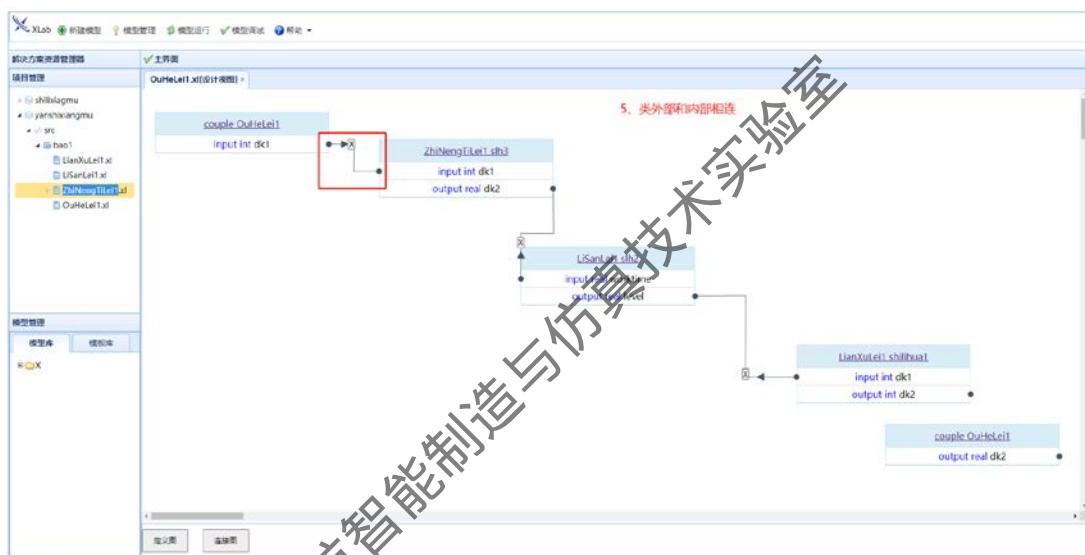
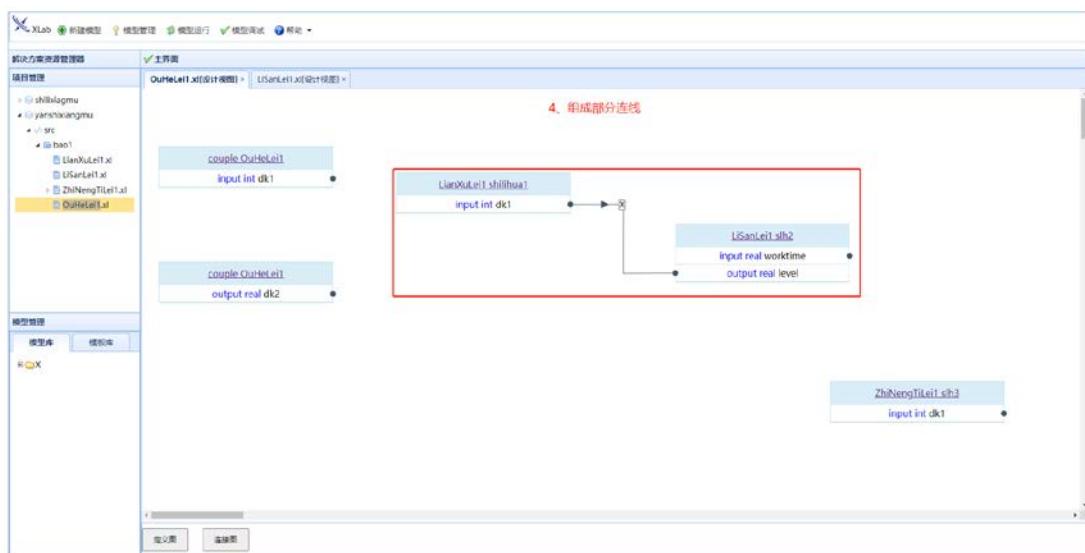
其他绩效评价材料



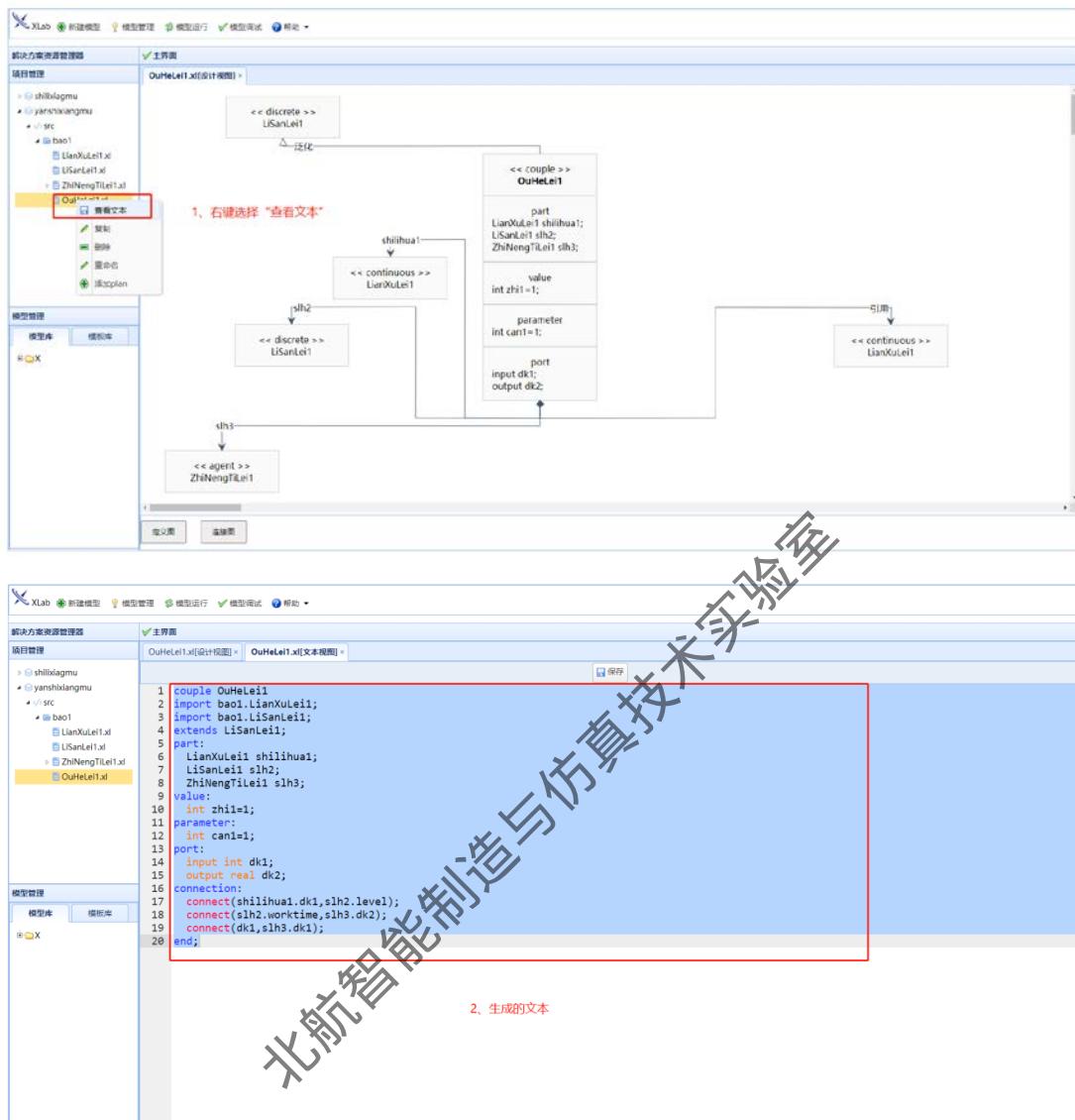
1.40. 耦合类-连接图



其他绩效评价材料



1.41. 耦合类-查看文本（图形到文本）



1.42. 耦合类-文本保存（文本到图形）

1. 编辑文本

```

1 couple OuHeLei1
2 import ControlModel;
3 import JiaoZhuModel;
4 part:
5   ControlModel control;
6   JiaoZhuModel jiaozhu1,jiaozhu2,jiaozhu3,jiaozhu4,jiaozhu5,jiaozhu6,jiaozhu7,jiaozhu8,jiaozhu9,jiaozhu10;
7 connection:
8   connect(control.i0, jiaozhu0.worktime);
9   connect(control.i1, jiaozhu1.worktime);
10  connect(control.i2, jiaozhu2.worktime);
11  connect(control.i3, jiaozhu3.worktime);
12  connect(control.i4, jiaozhu4.worktime);
13  connect(control.i5, jiaozhu5.worktime);
14  connect(control.i6, jiaozhu6.worktime);
15  connect(control.i7, jiaozhu7.worktime);
16  connect(control.i8, jiaozhu8.worktime);
17  connect(control.i9, jiaozhu9.worktime);
18  connect(jiaozhu0.level, control.i0);
19  connect(jiaozhu1.level, control.i1);
20  connect(jiaozhu2.level, control.i2);
21  connect(jiaozhu3.level, control.i3);
22  connect(jiaozhu4.level, control.i4);
23  connect(jiaozhu5.level, control.i5);
24  connect(jiaozhu6.level, control.i6);
25  connect(jiaozhu7.level, control.i7);
26  connect(jiaozhu8.level, control.i8);
27  connect(jiaozhu9.level, control.i9);
28  connect(jiaozhu0.level, control.i9);
29 end;

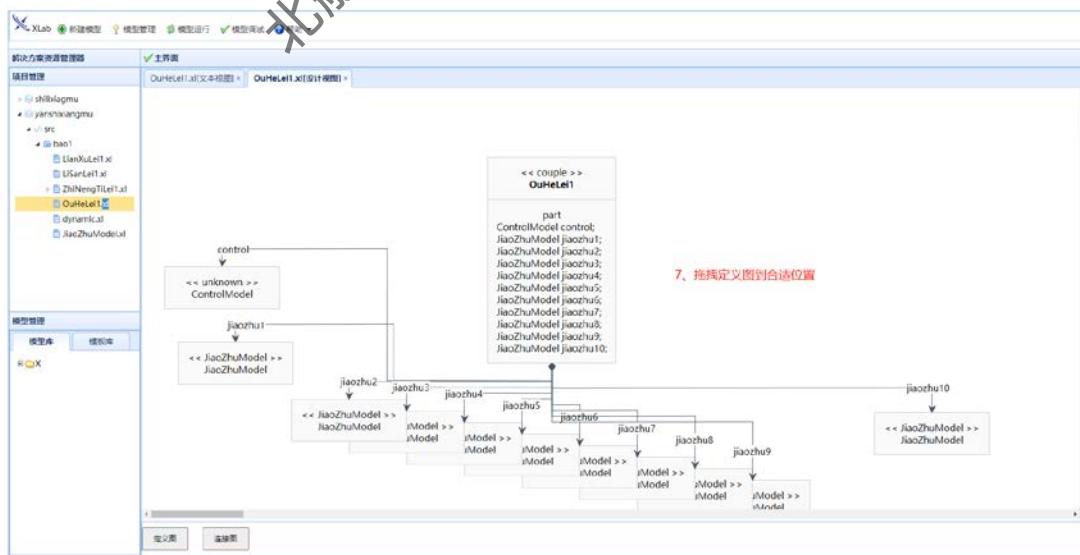
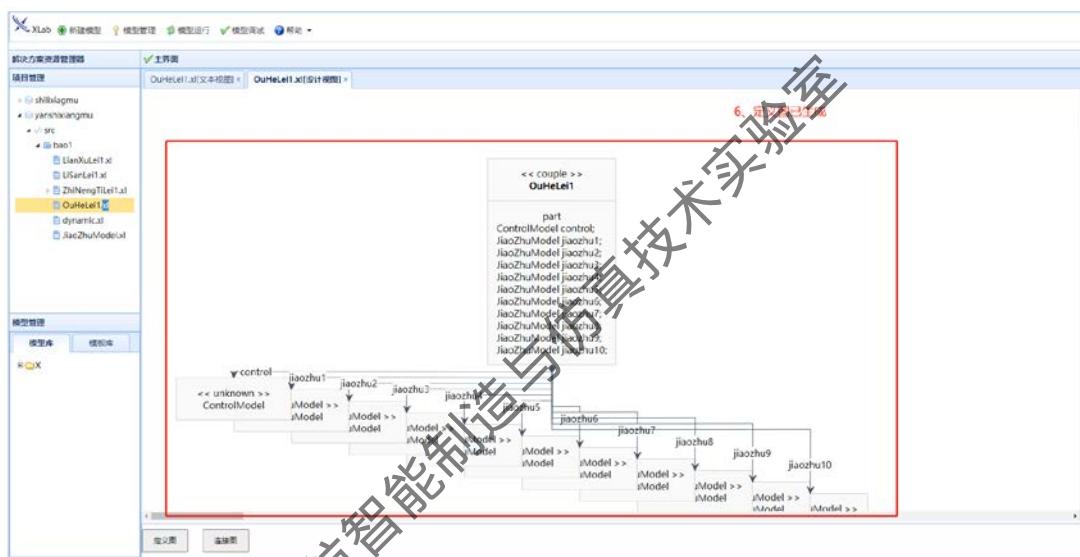
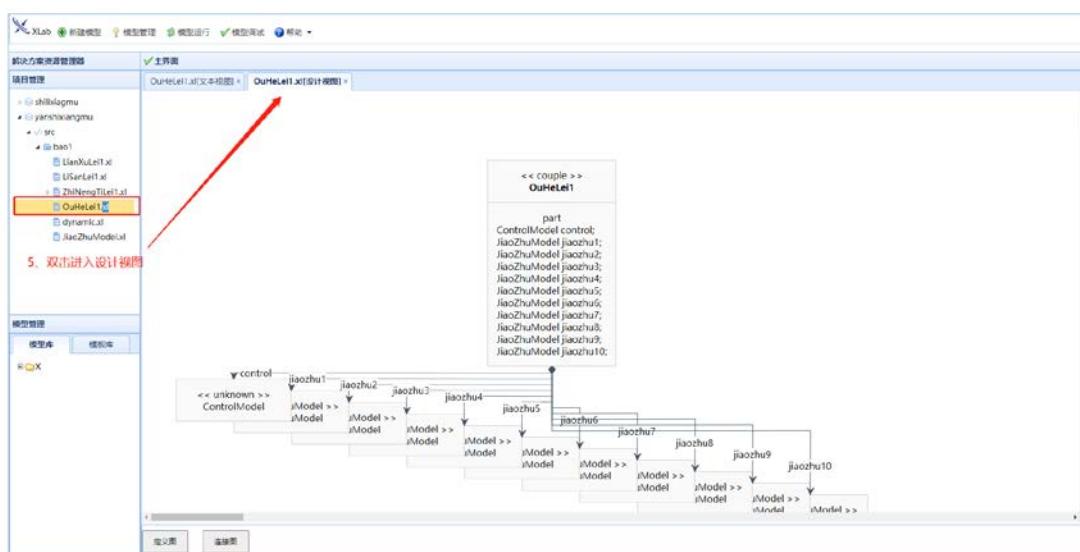
```

2. 点击保存

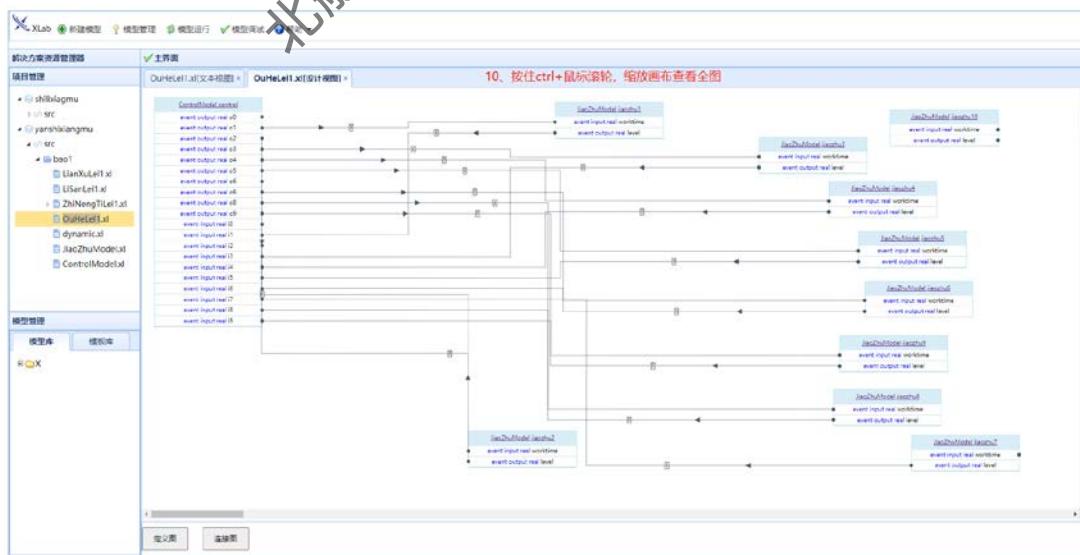
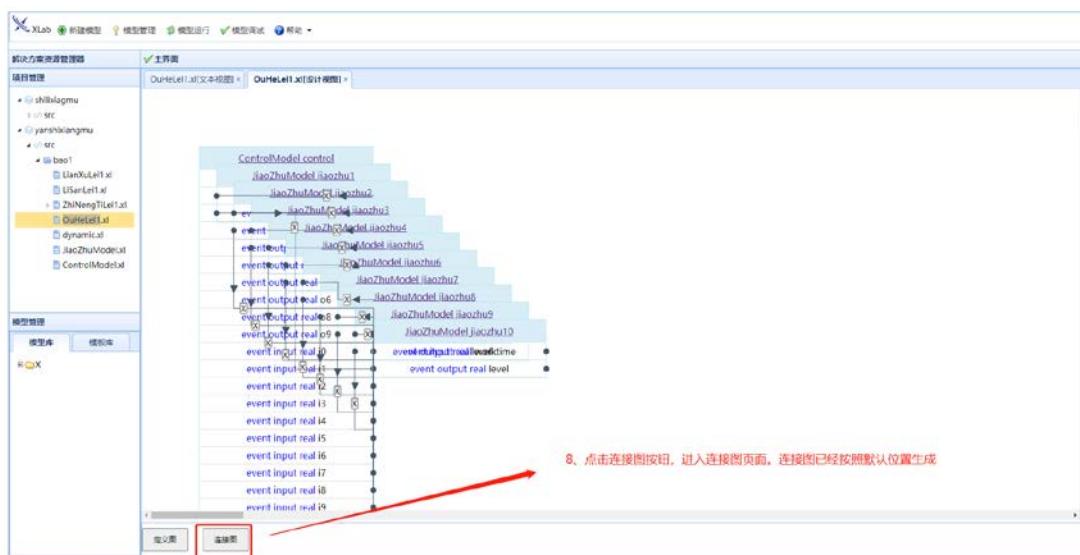
localhost:9888/xlab/editor/execute?modelId=451&view=OuHeLei1&showingModel=true

3. 保存成功

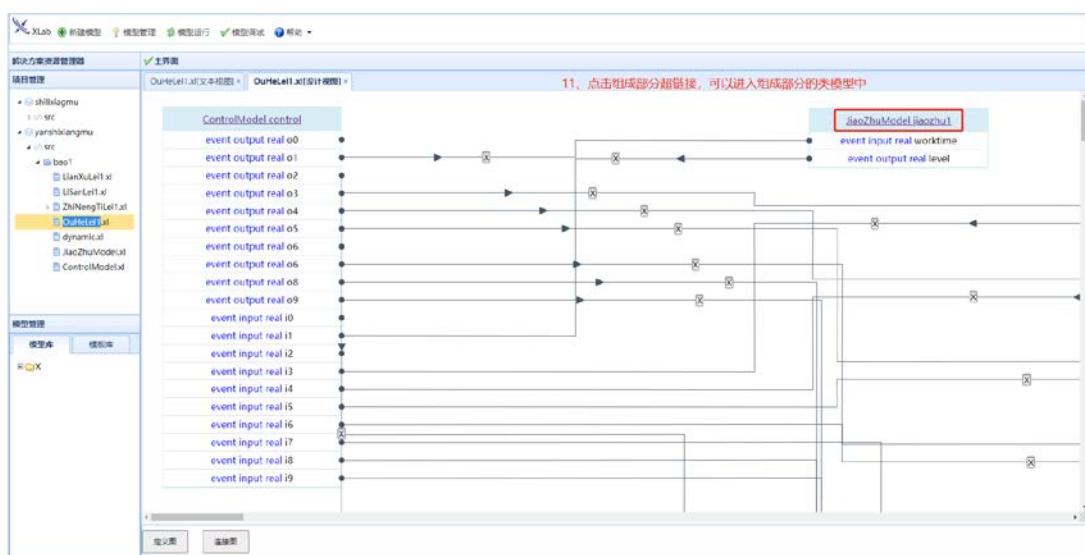
其他绩效评价材料



其他绩效评价材料



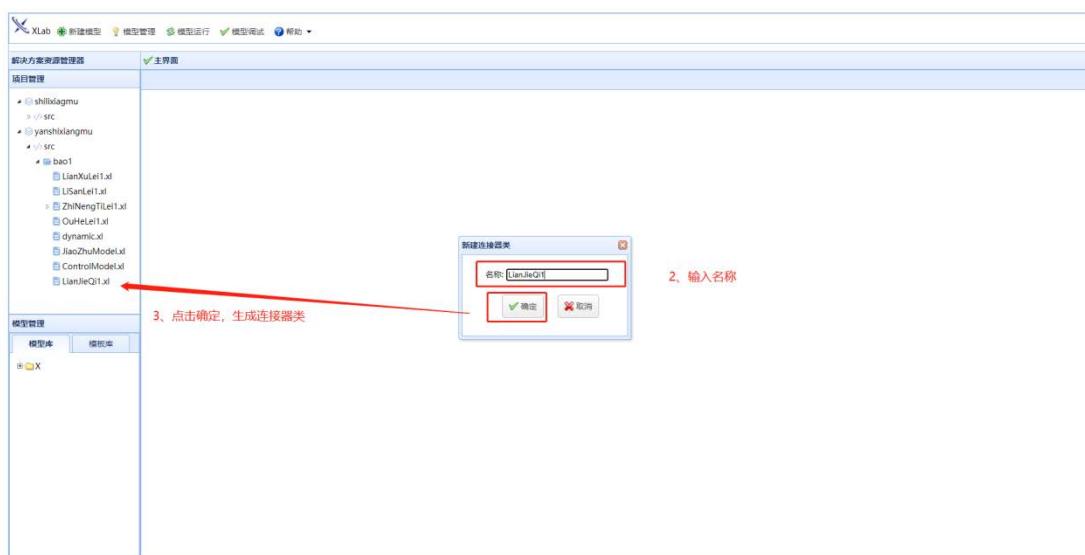
其他绩效评价材料



1.43. 新建连接器类



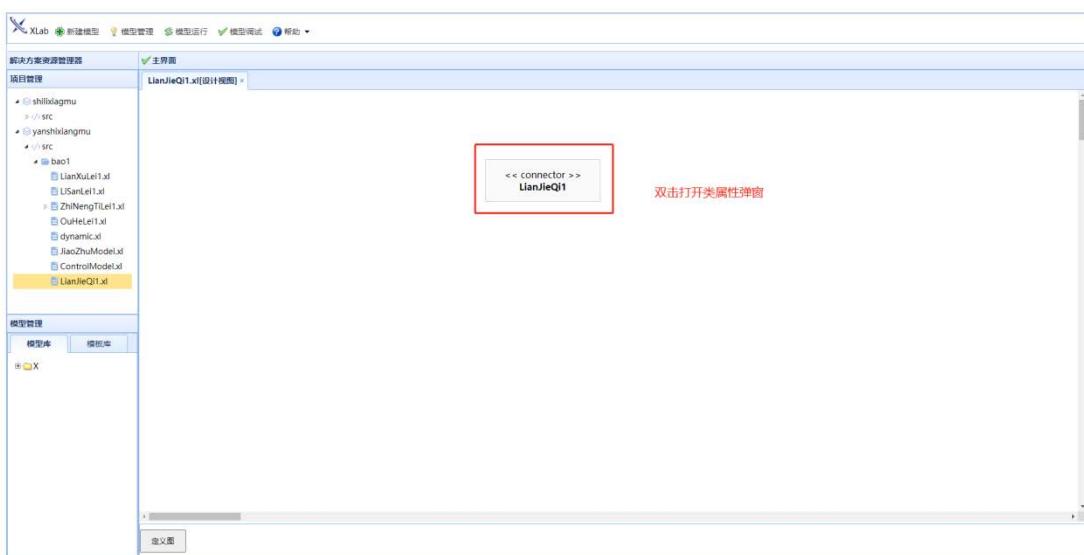
其他绩效评价材料



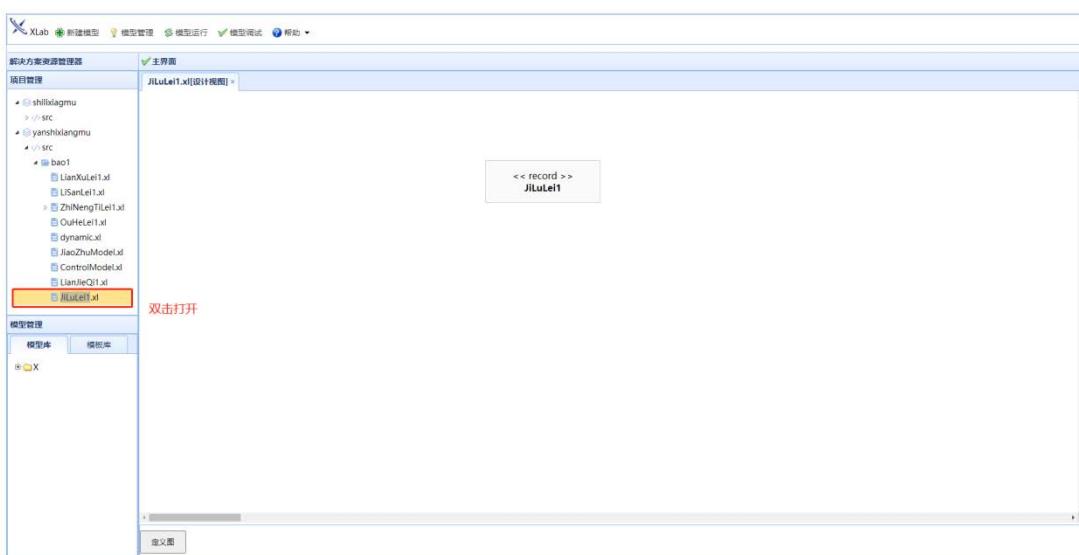
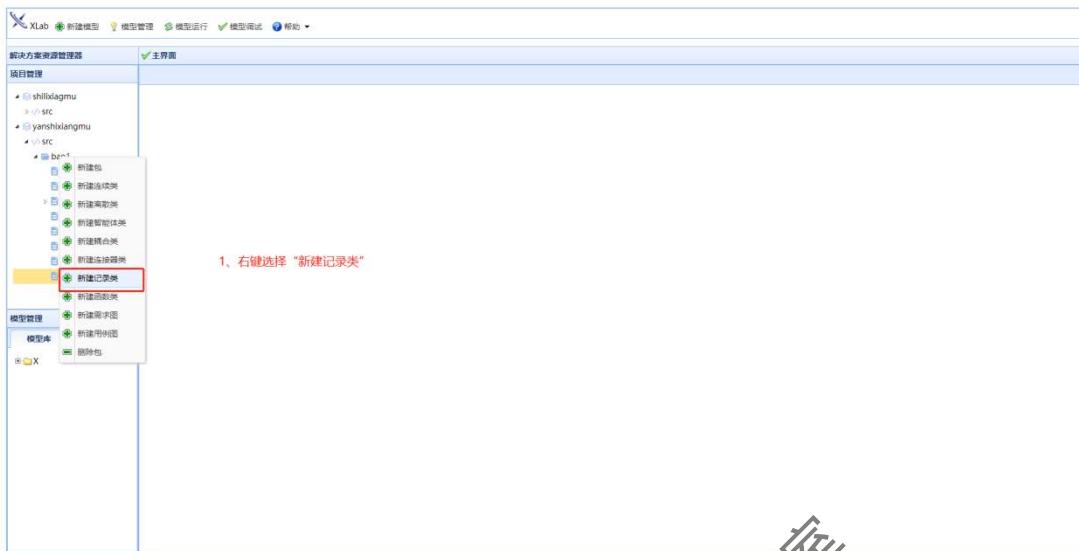
1.44. 打开连接器类



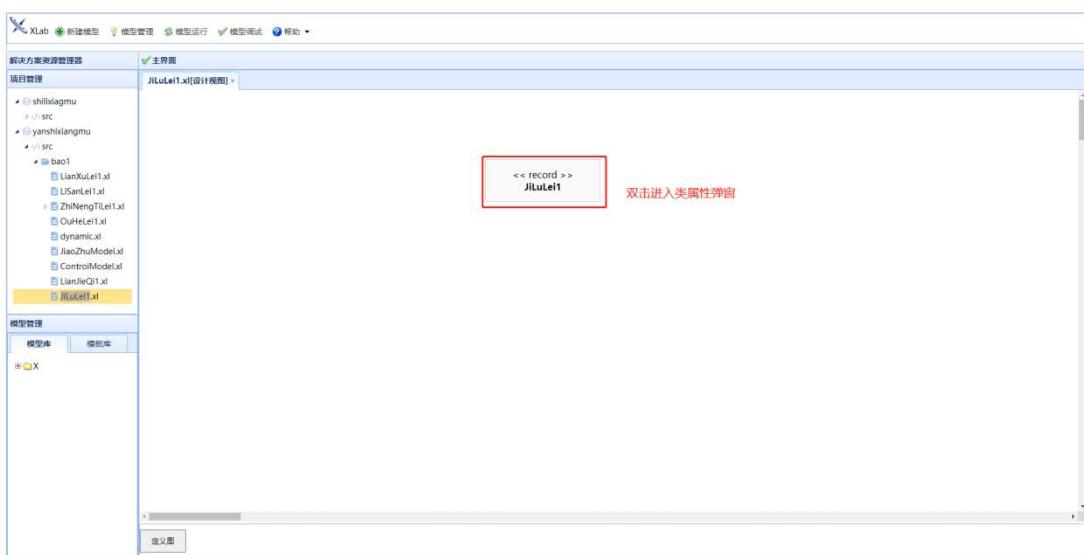
其他绩效评价材料



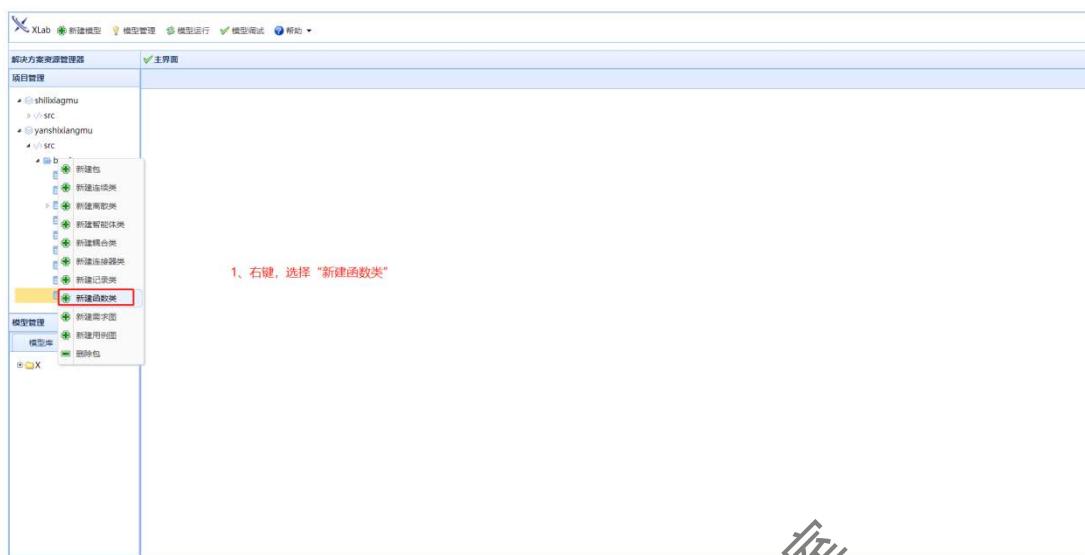
1.45. 新建记录类



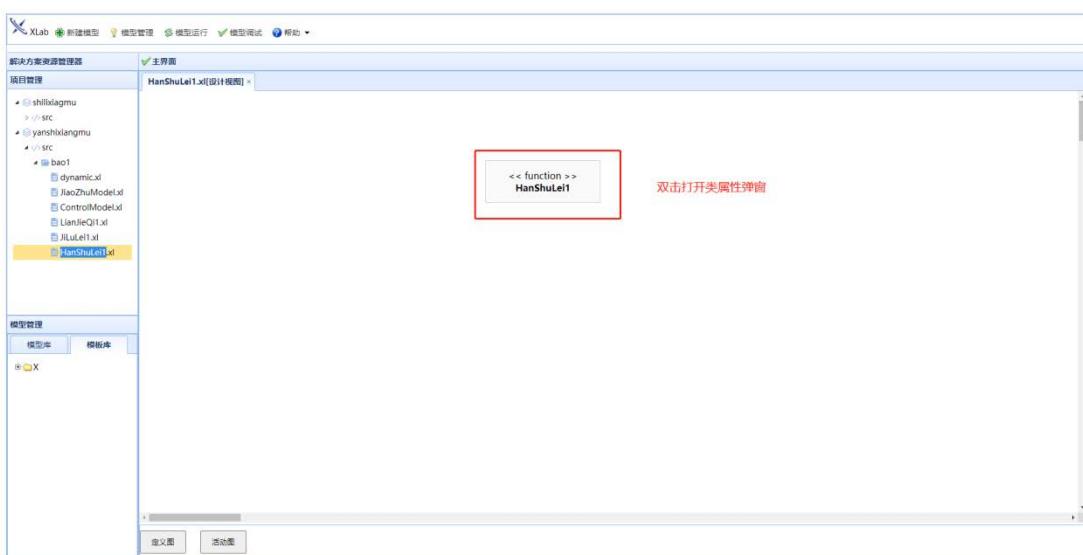
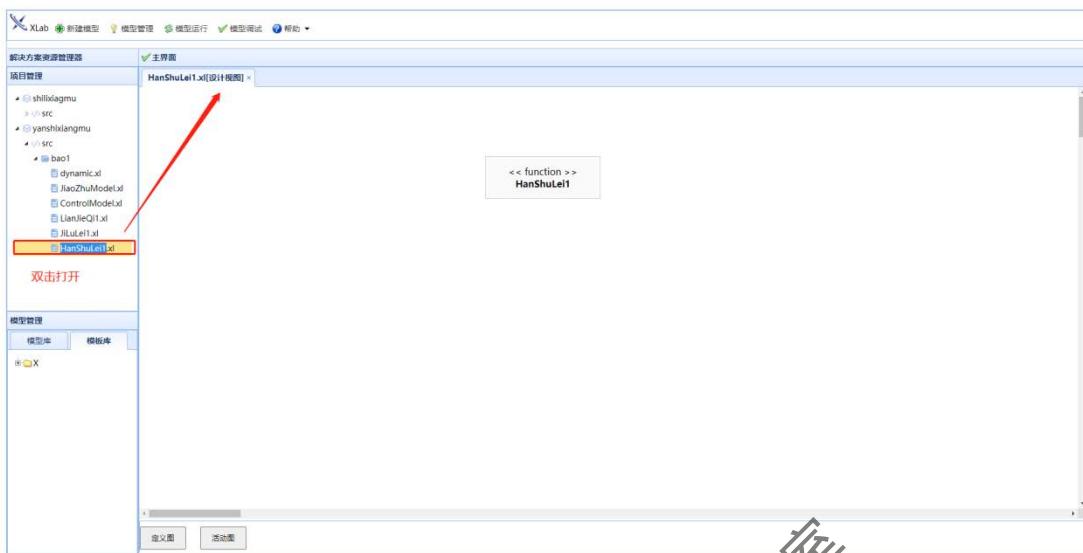
其他绩效评价材料



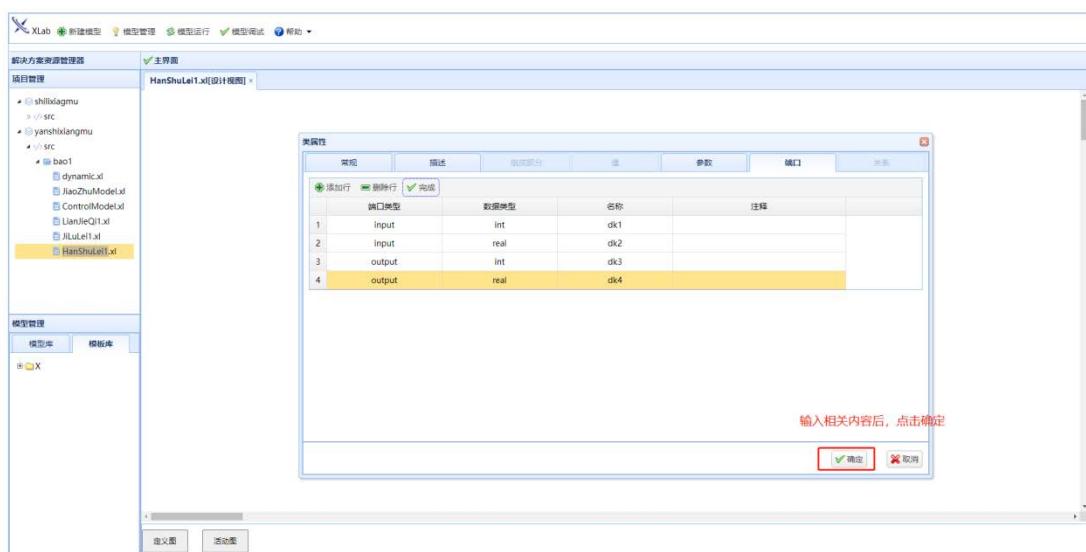
1.46. 新建函数类



1.47. 打开函数类



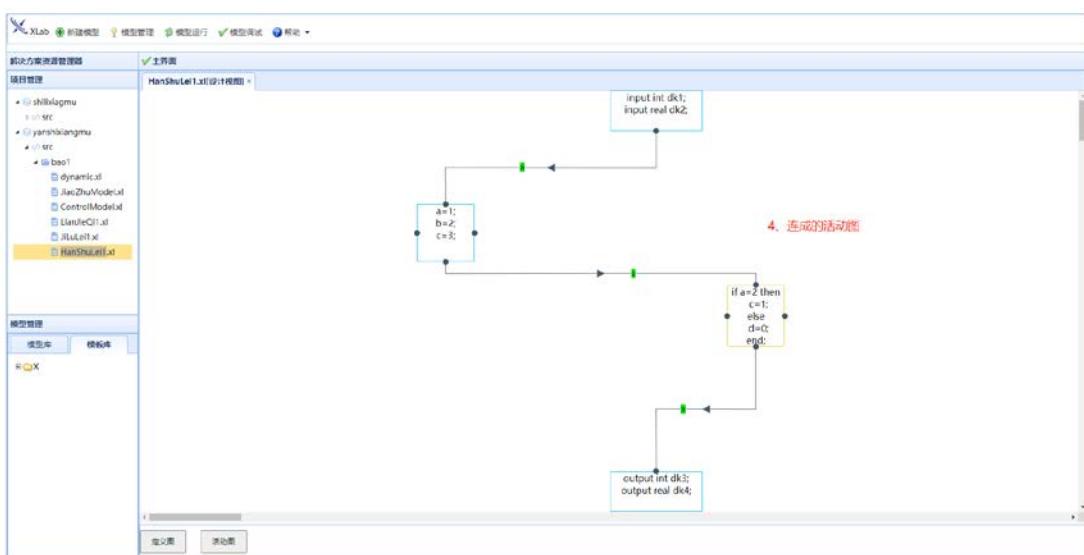
其他绩效评价材料



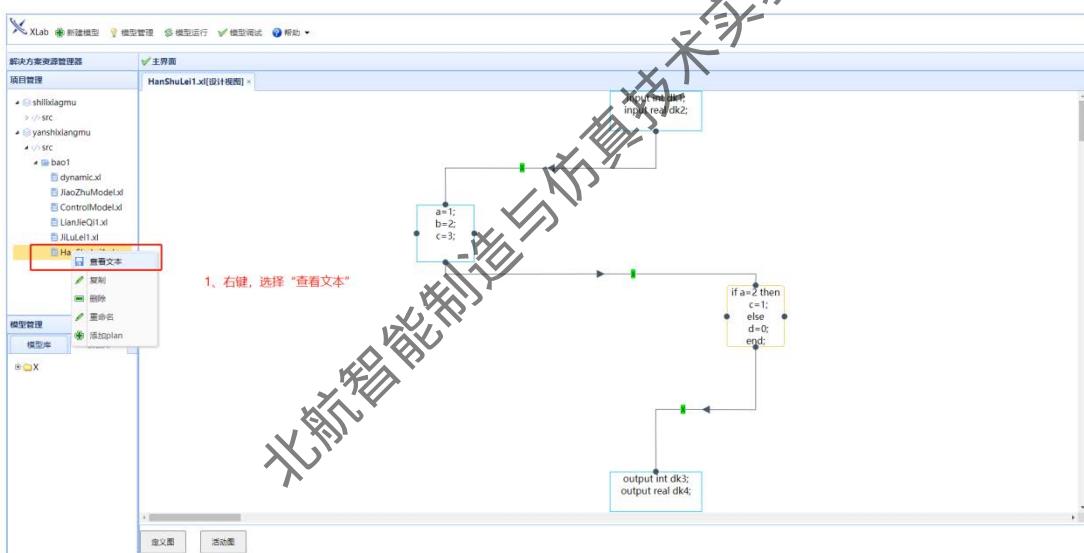
1.48. 函数类-活动图



其他绩效评价材料



1.49. 函数类-查看文本（图形到文本）



其他绩效评价材料

The screenshot shows the XLab software interface. On the left, the 'Project Explorer' pane lists several files under 'shixiangmu' and 'src'. In the center, the 'Main View' tab is active, displaying a code editor with the following C code:

```
1 function HanShuLei1
2 parameter:
3     int can1;
4     real can2=2;
5 port:
6     input int dk1;
7     input real dk2;
8     output int dk3;
9     output real dk4;
10 action:
11     a=1;
12     b=2;
13     c=3;
14     if a==2 then
15         c=1;
16     else
17         d=0;
18 end;
19 end;
```

A red box highlights the code area, and the text '2. 文本代码已生成' (Text code has been generated) is displayed at the bottom right of the code editor.

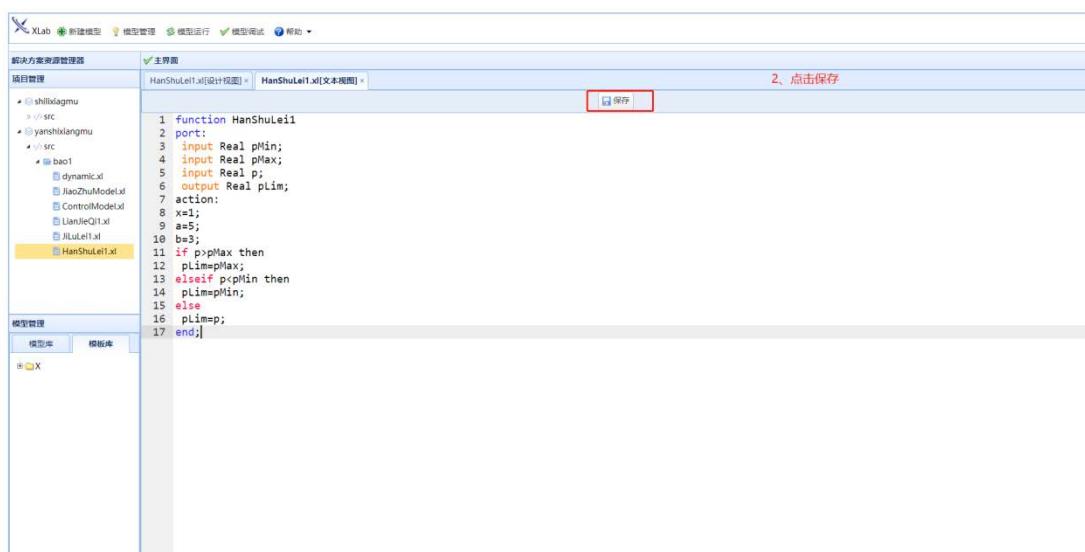
1.50. 函数类-文本保存（文本到图形）

The screenshot shows the XLab software interface. The 'Project Explorer' pane is identical to the previous one. The 'Main View' tab is active, displaying a code editor with the following C code:

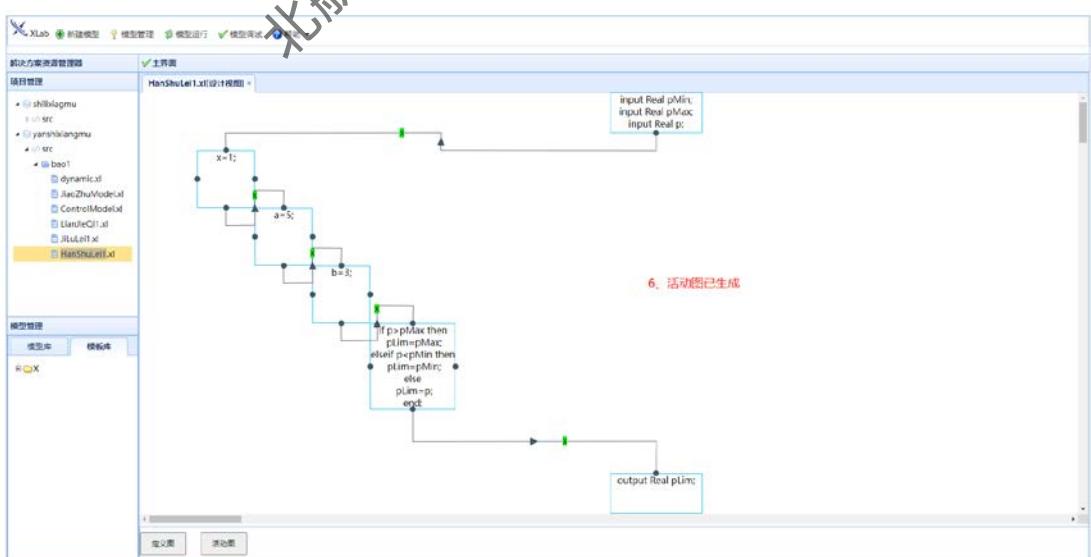
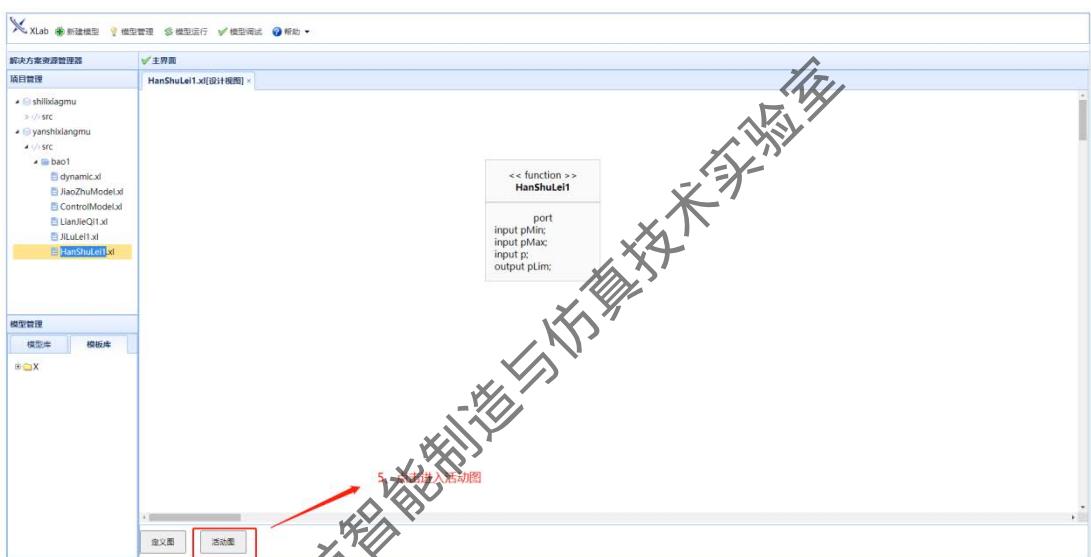
```
1 function HanShuLei1
2 port:
3     input Real pMin;
4     input Real pMax;
5     input Real p;
6     output Real plim;
7 action:
8     x=1;
9     a=5;
10    b=3;
11    if p>pMax then
12        plim=pMax;
13    elseif p<pMin then
14        plim=pMin;
15    else
16        plim=p;
17    end;
18 end;
```

A red box highlights the code area, and the text '1. 编辑文本' (Edit text) is displayed at the bottom right of the code editor.

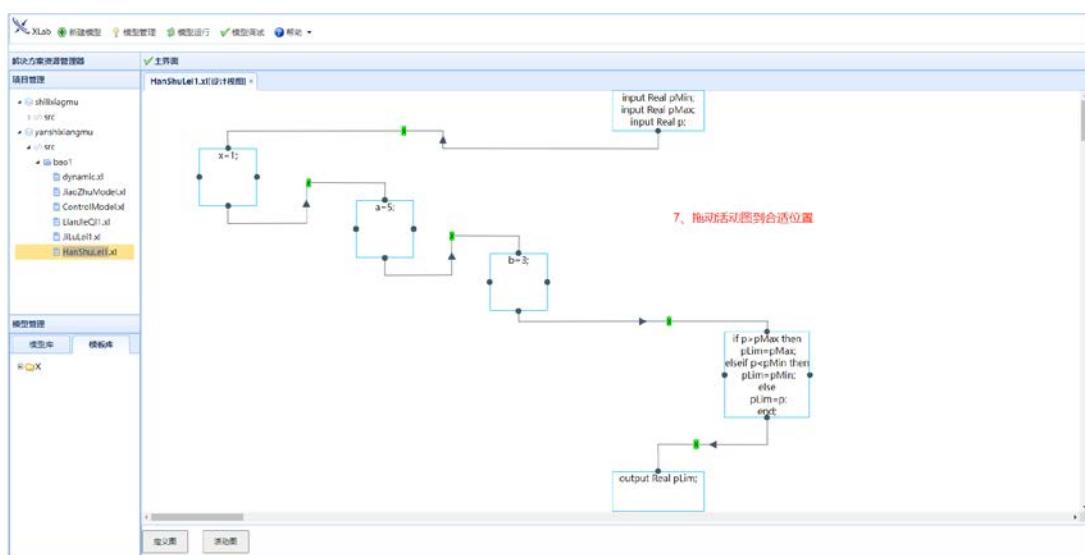
其他绩效评价材料



其他绩效评价材料



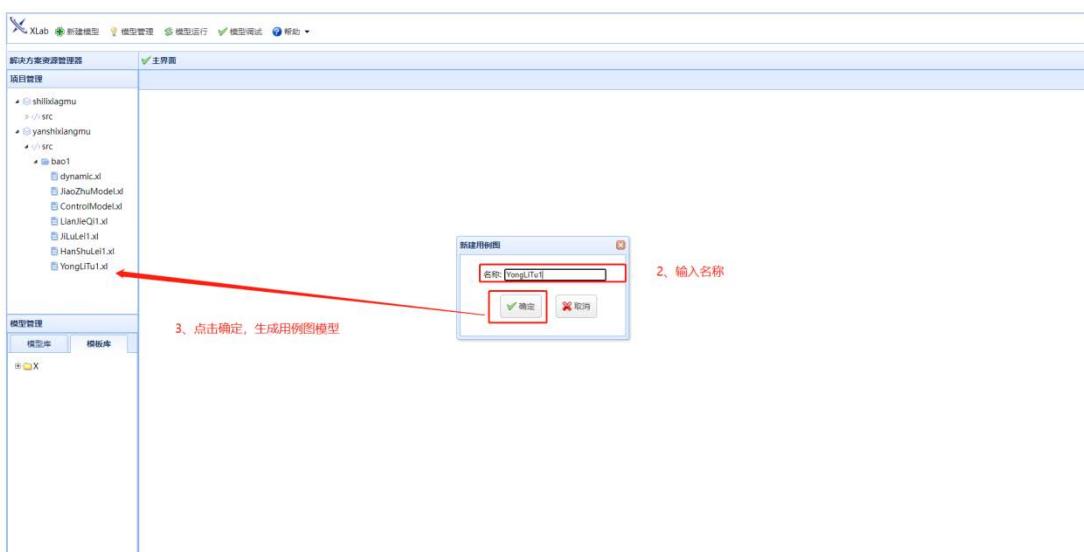
其他绩效评价材料



1.51. 新建用例图



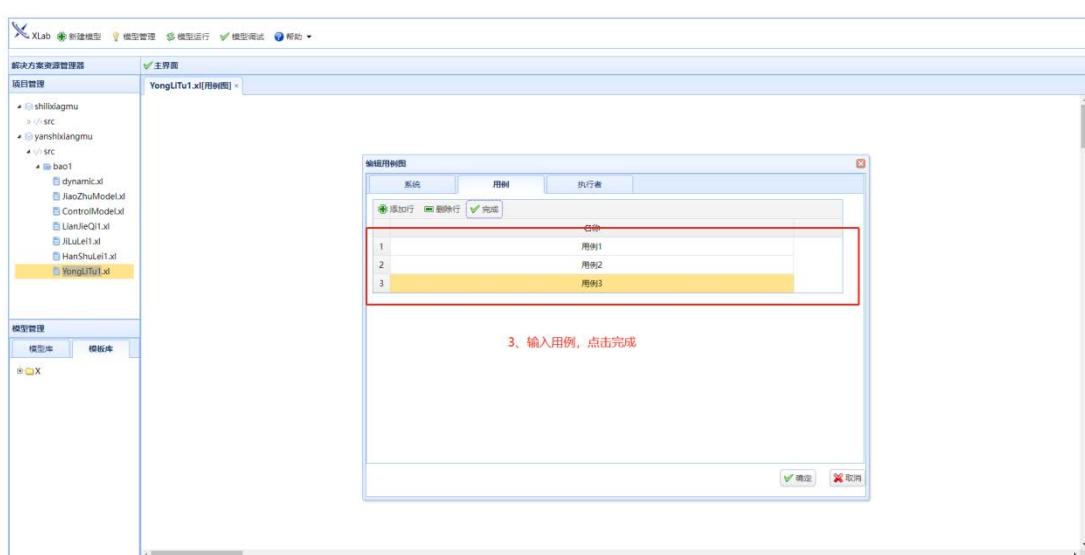
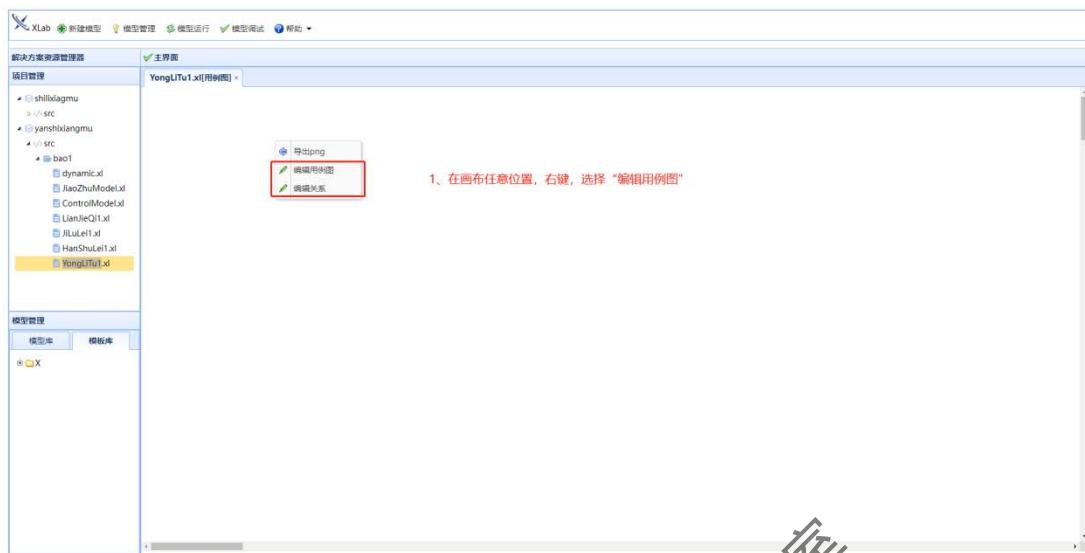
其他绩效评价材料



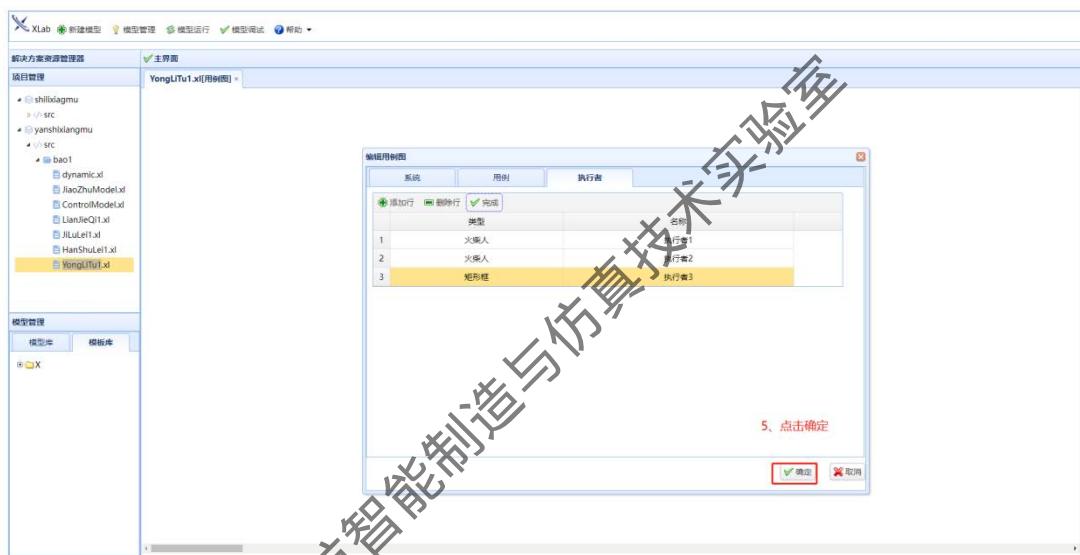
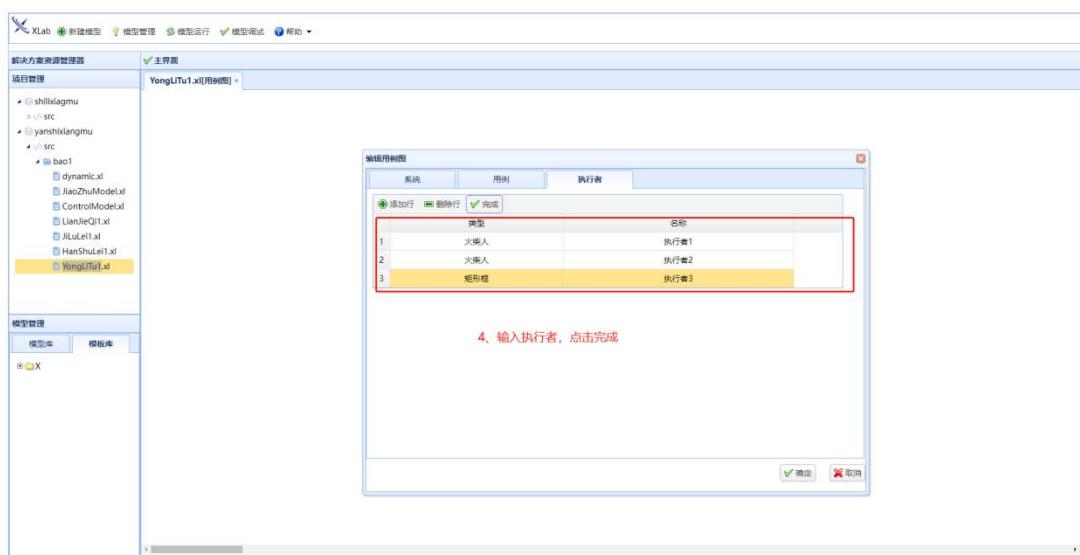
1.52. 打开用例图



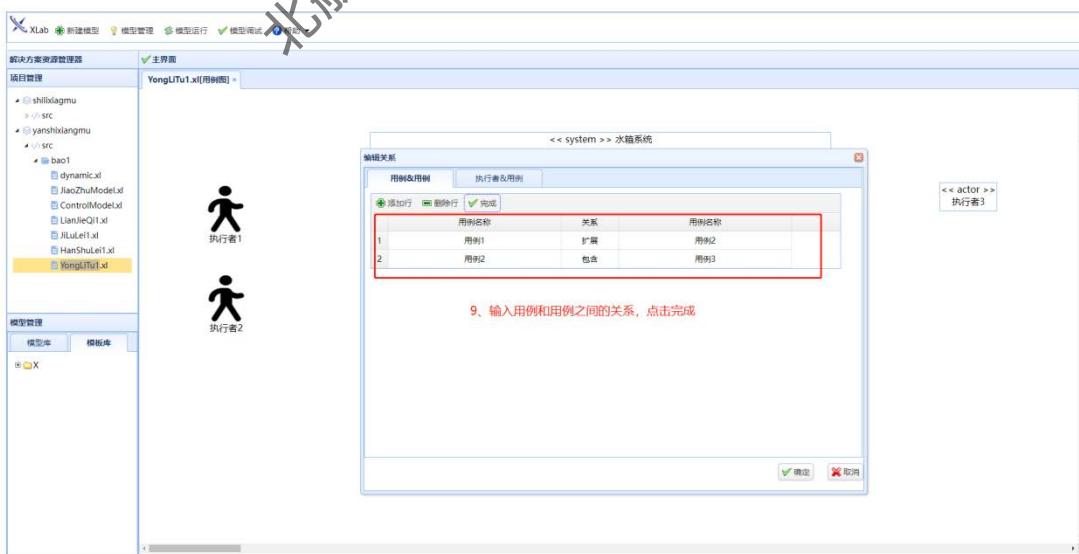
1.53. 编辑用例图



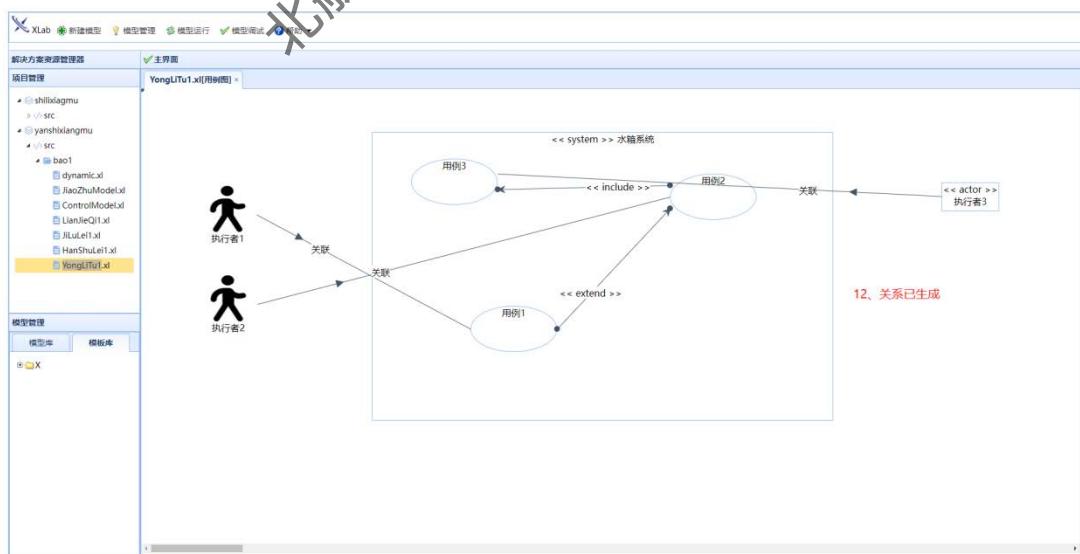
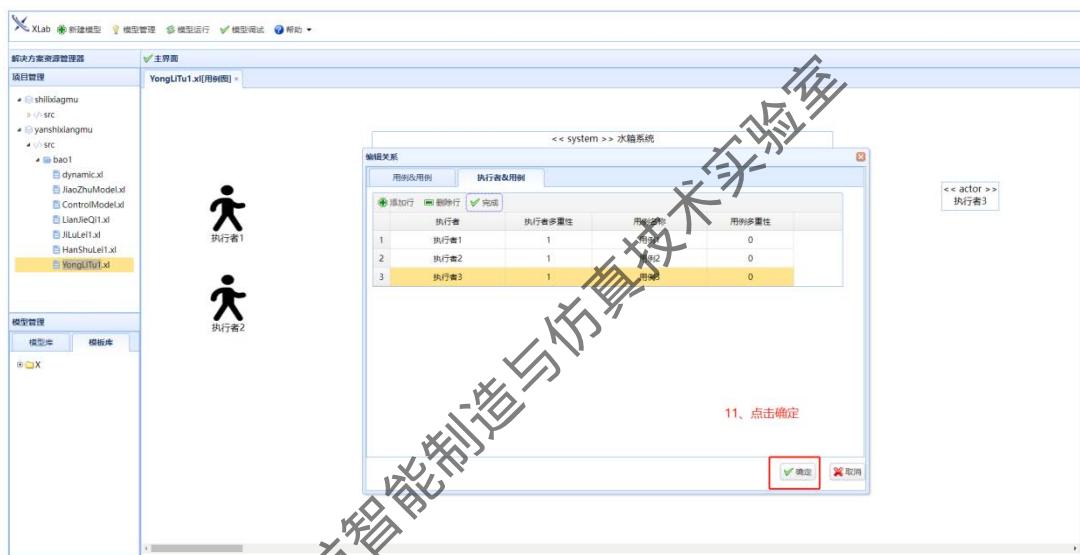
其他绩效评价材料



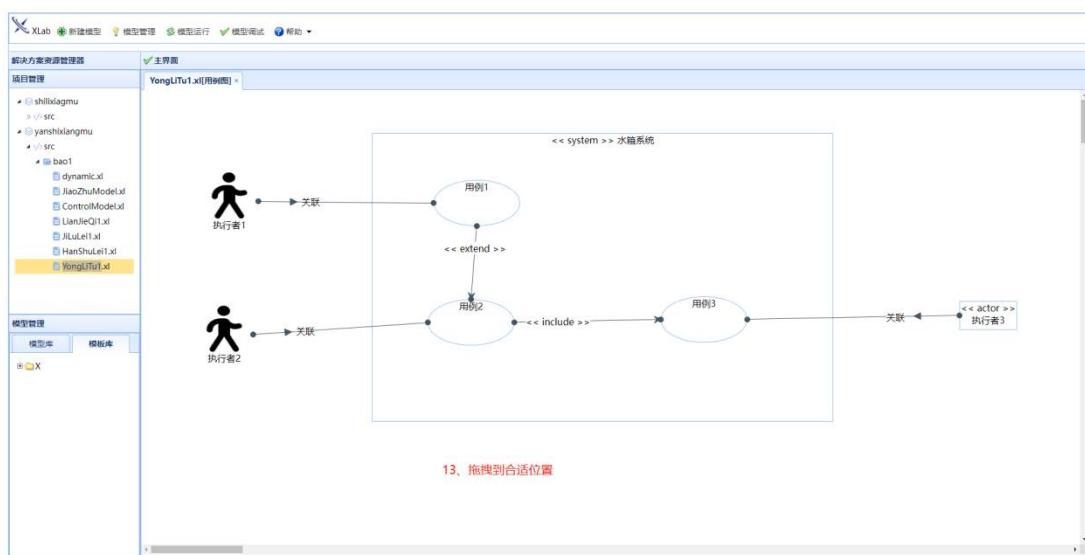
其他绩效评价材料



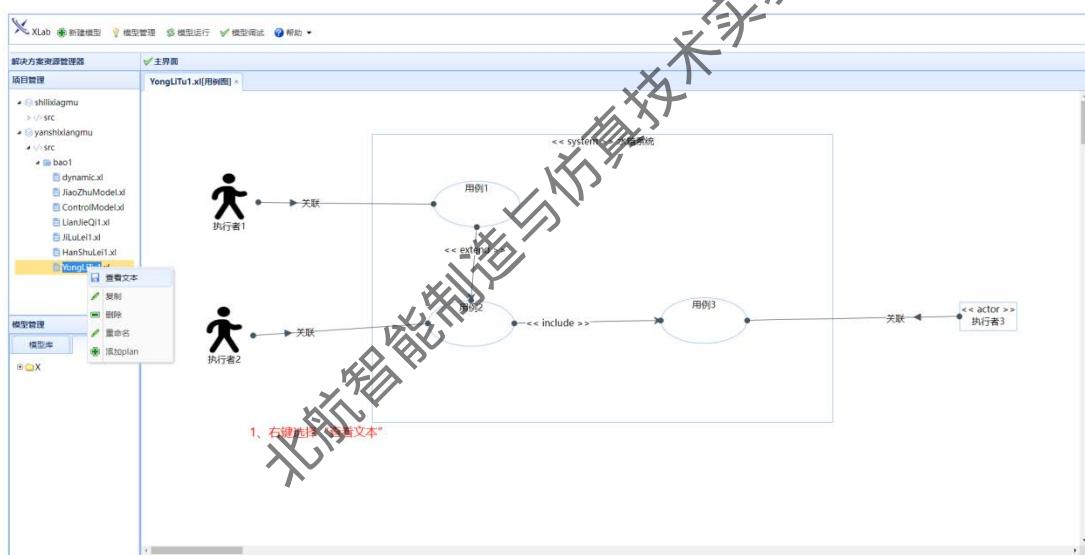
其他绩效评价材料



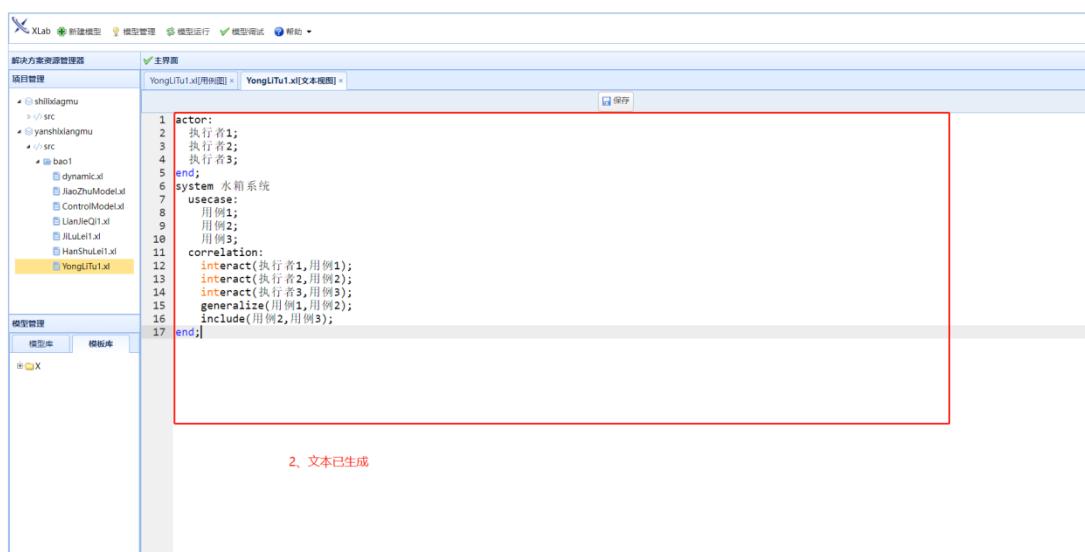
其他绩效评价材料



1.54. 用例图-查看文本（图形到文本）



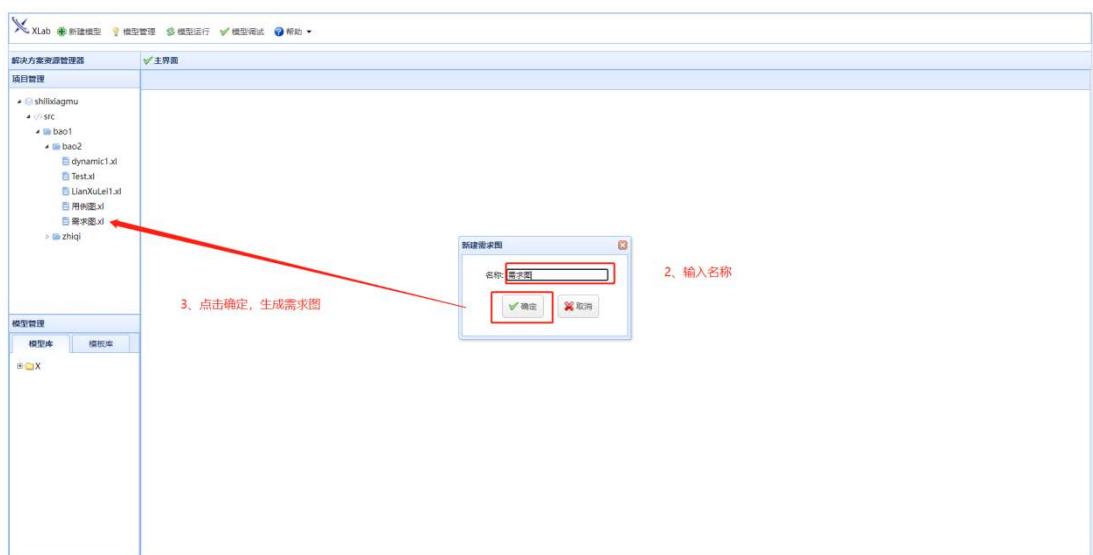
其他绩效评价材料



1.55. 新建需求图



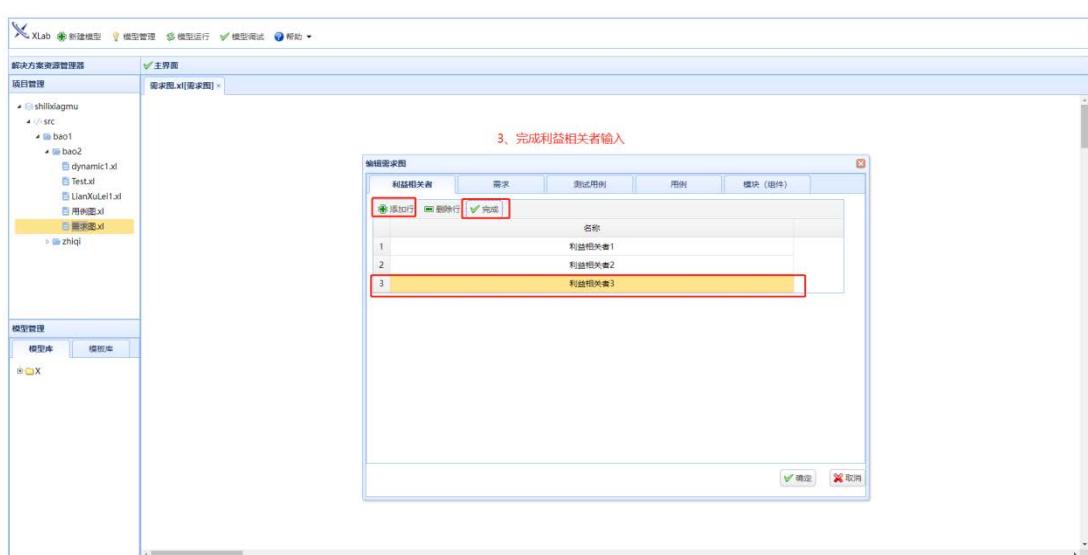
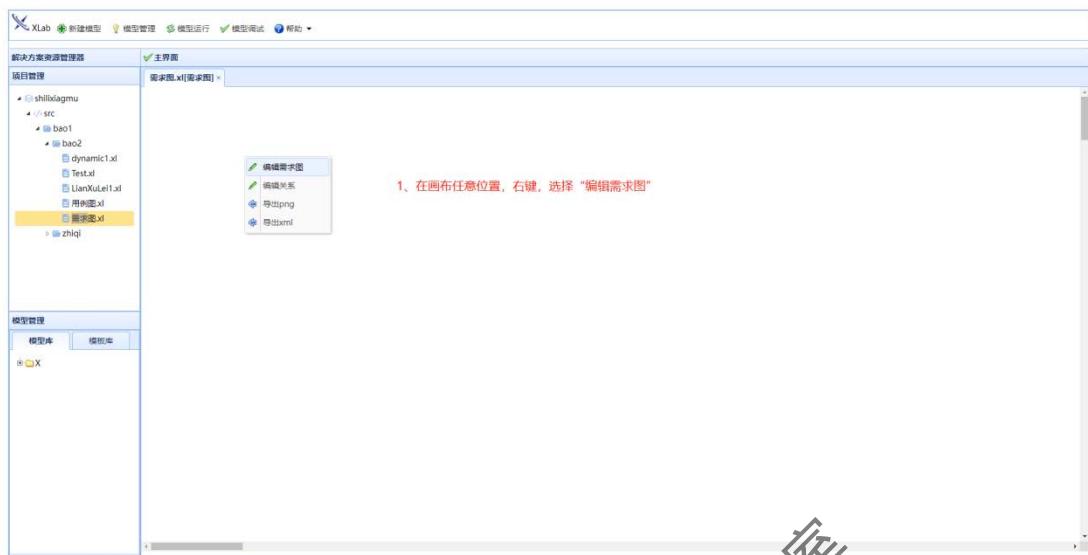
其他绩效评价材料



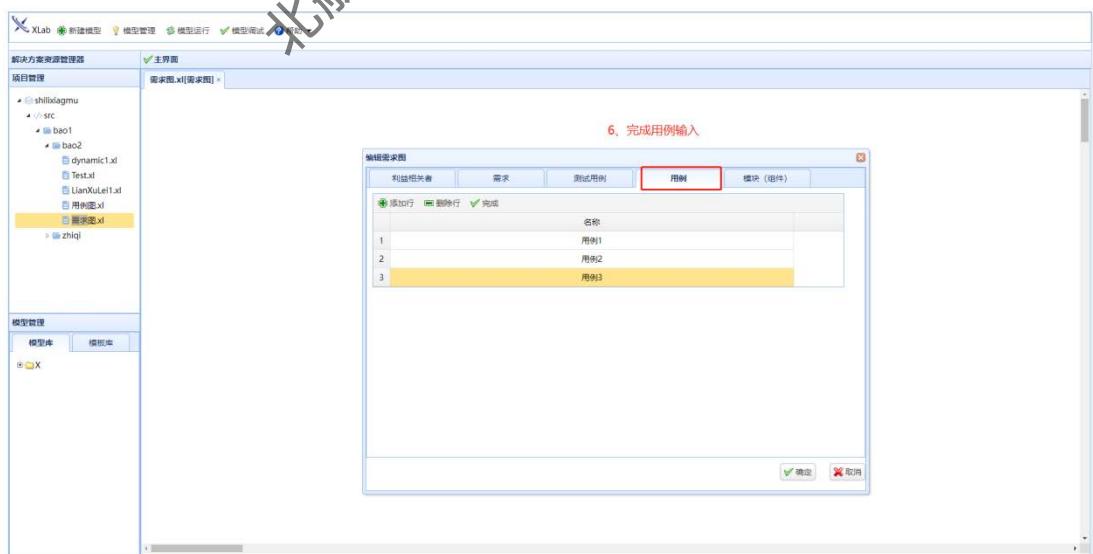
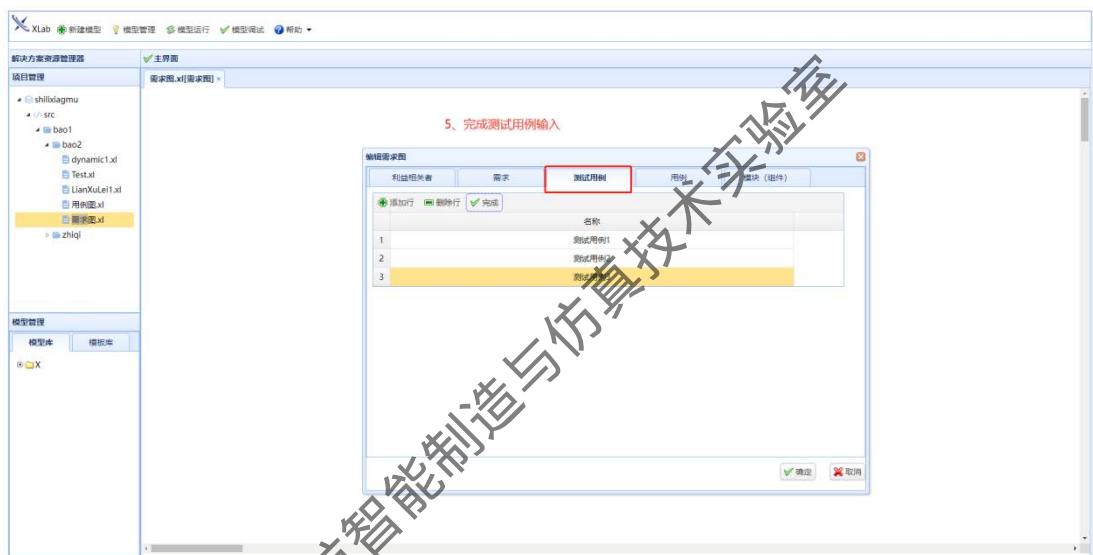
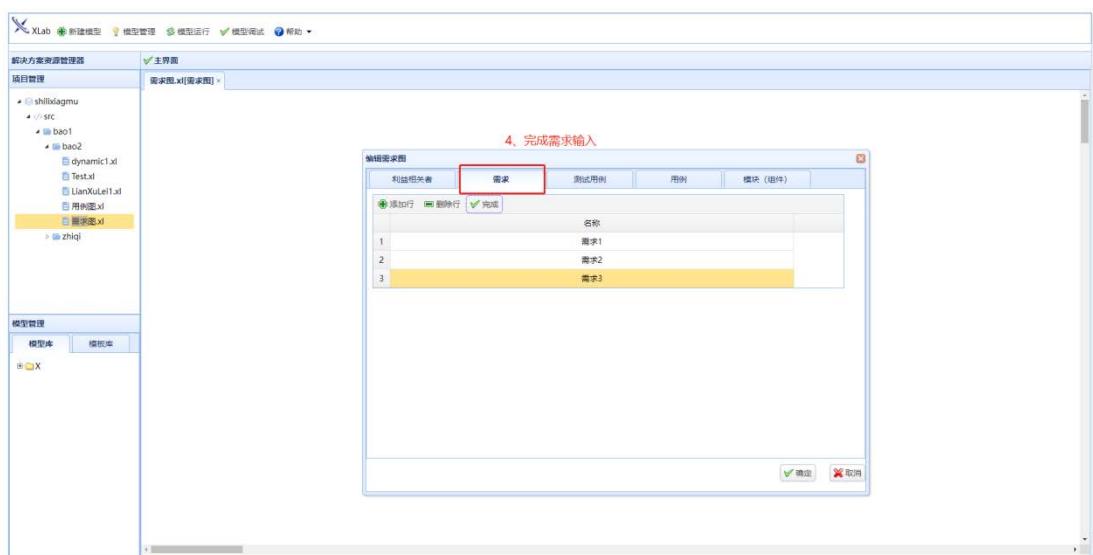
1.56. 打开需求图



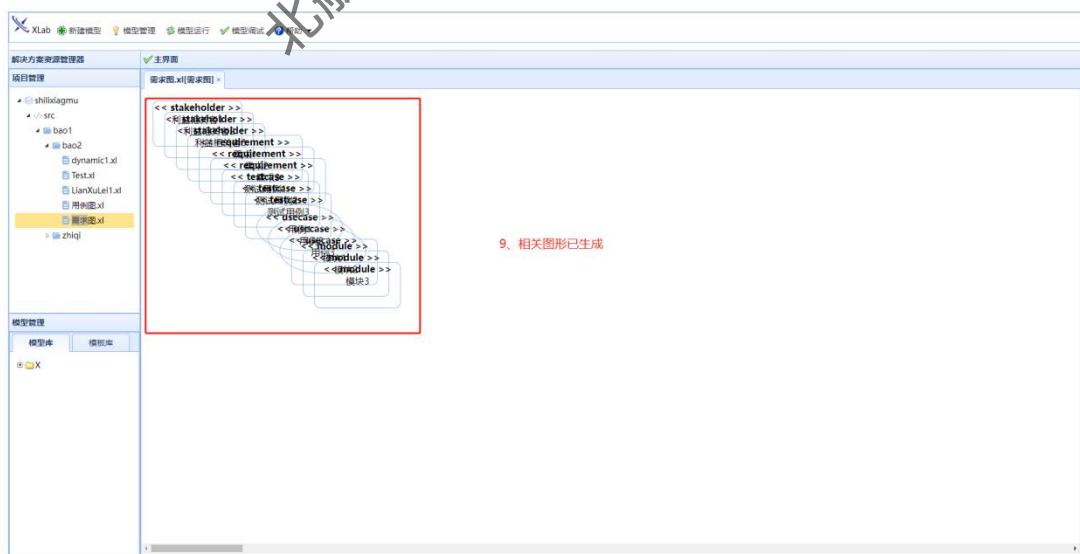
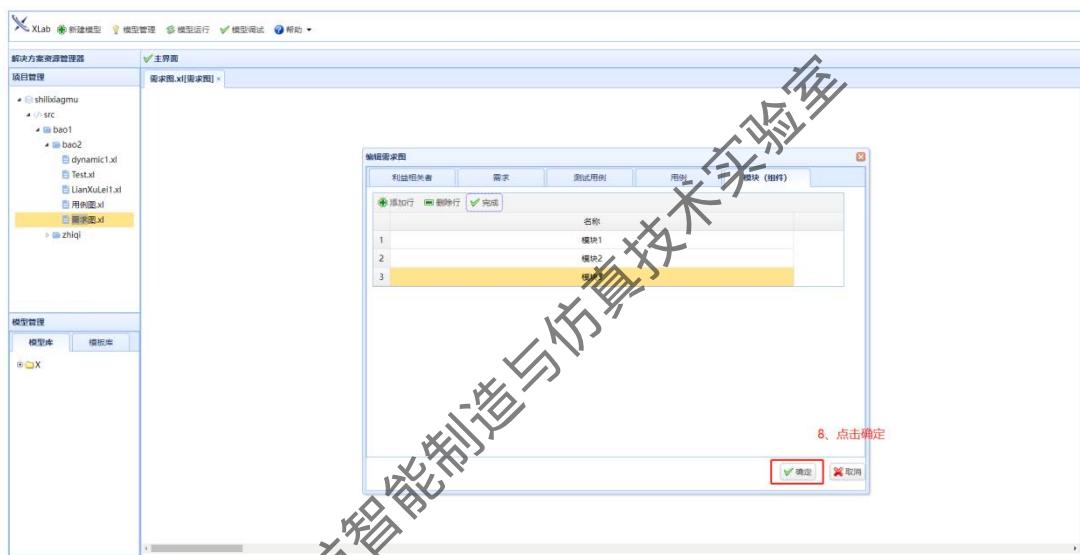
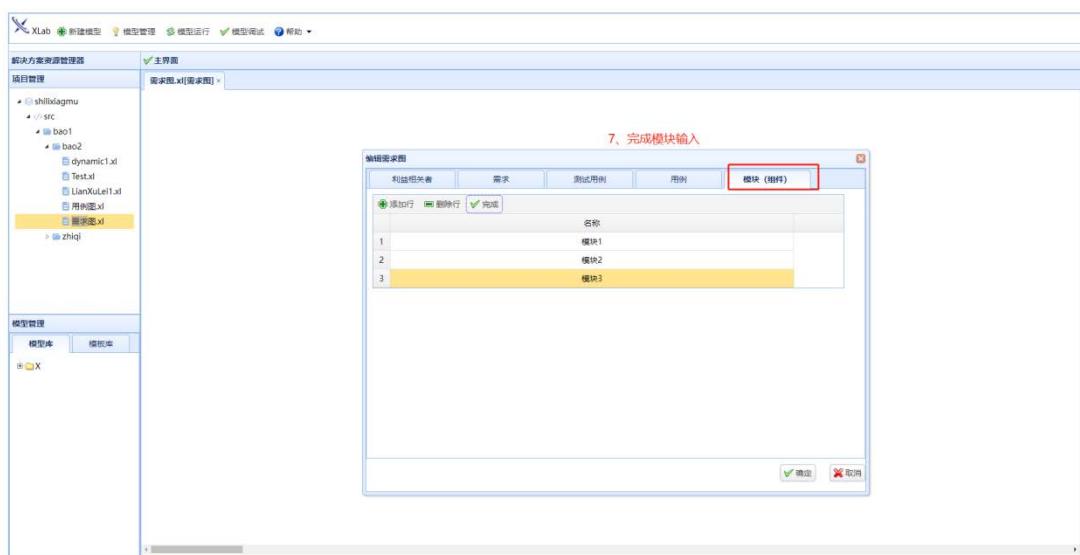
1.57. 编辑需求图



其他绩效评价材料



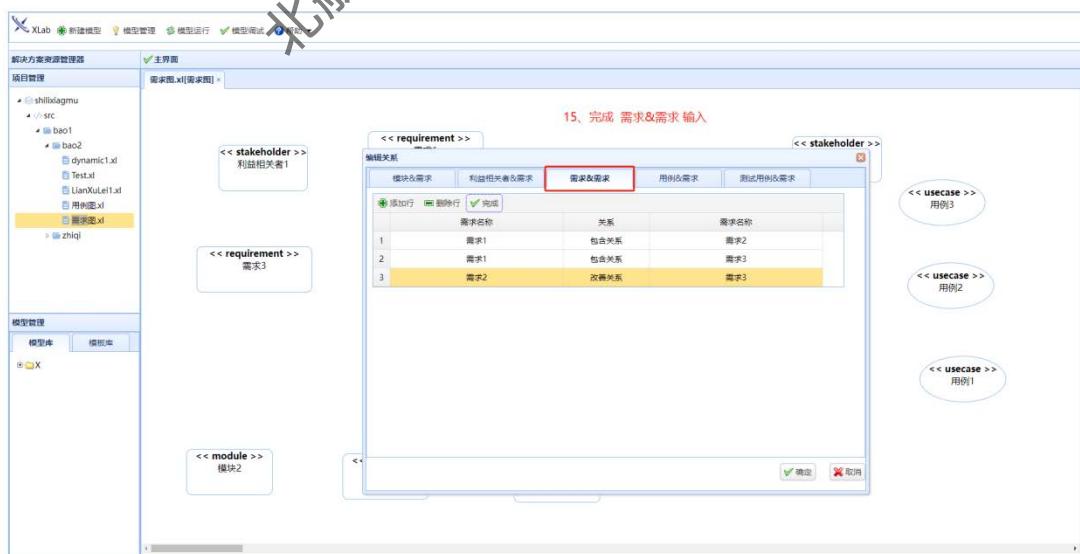
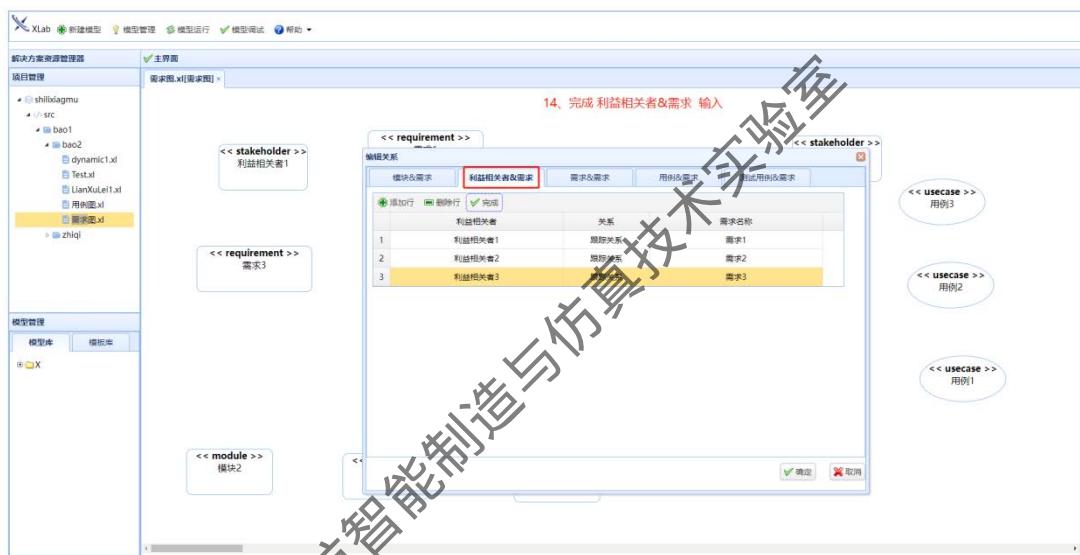
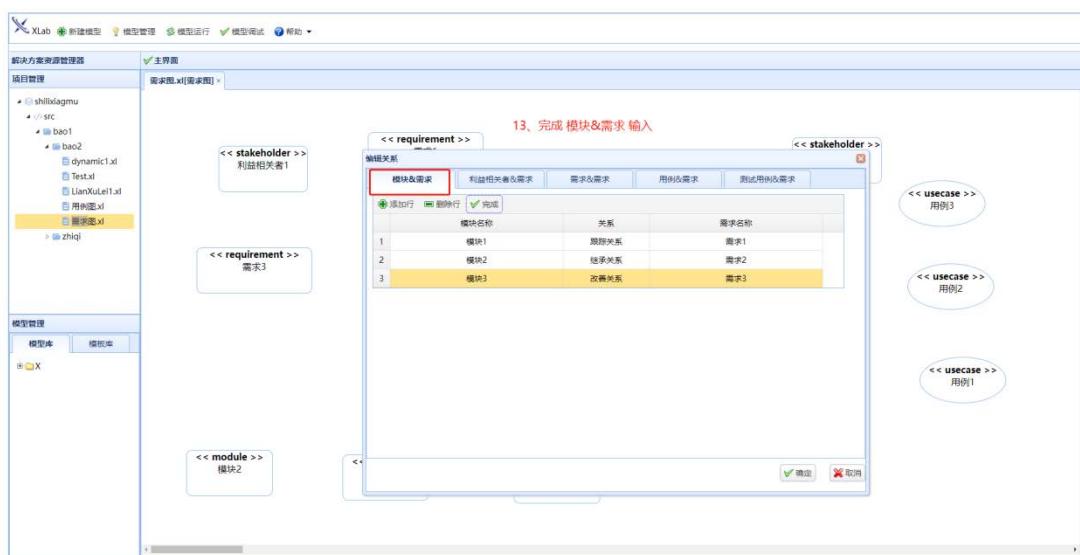
其他绩效评价材料



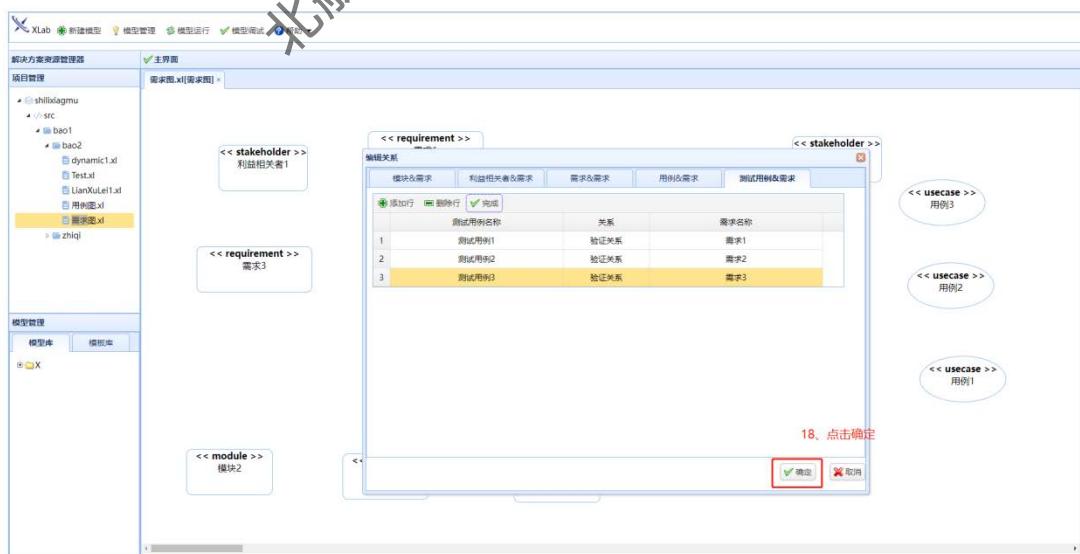
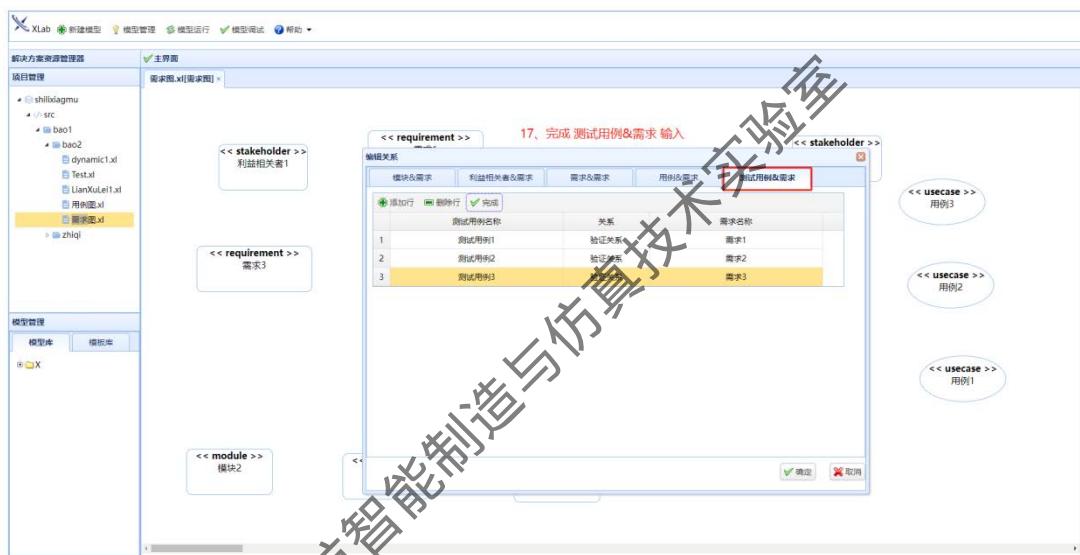
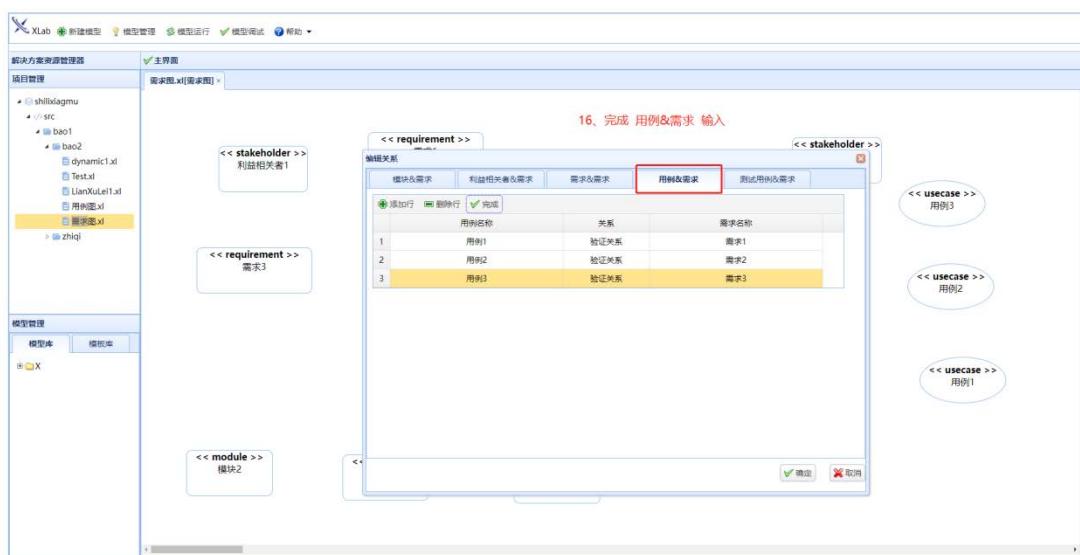
其他绩效评价材料



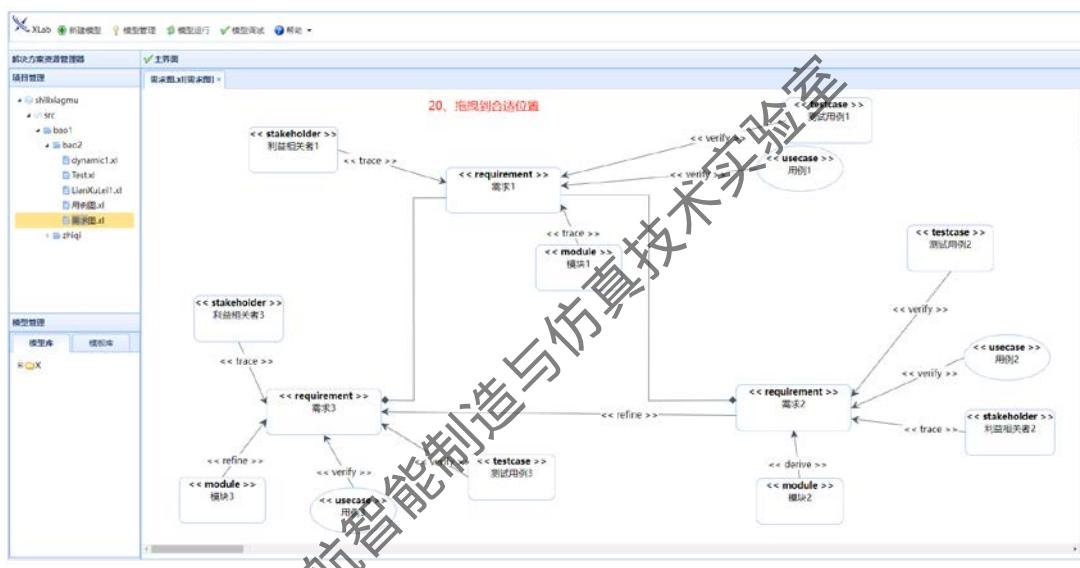
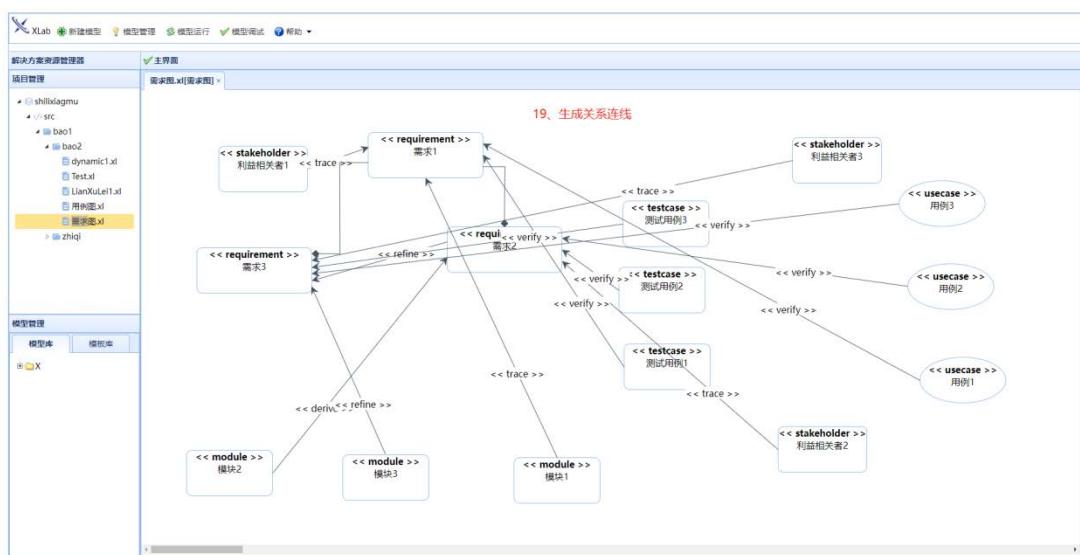
其他绩效评价材料



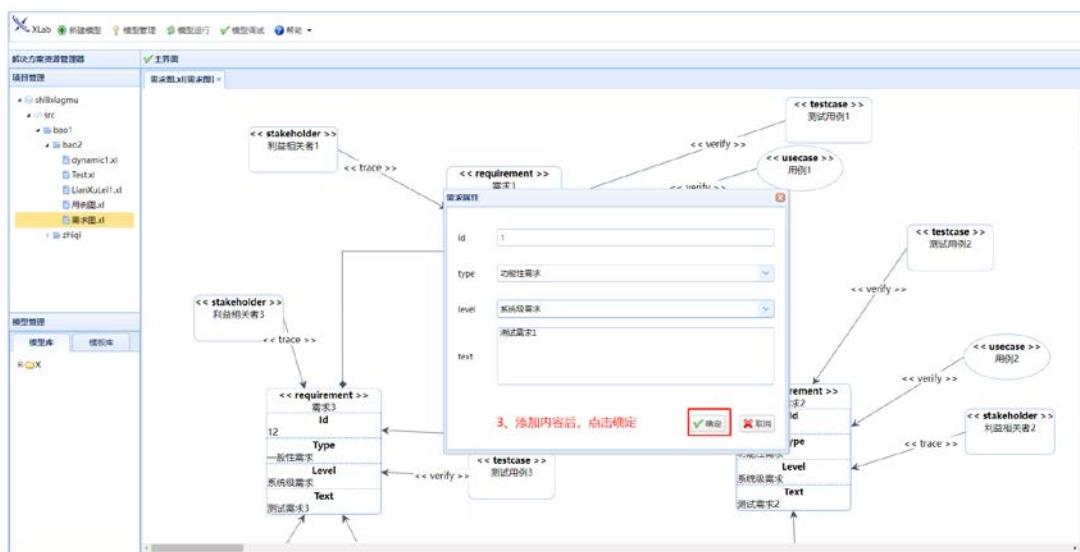
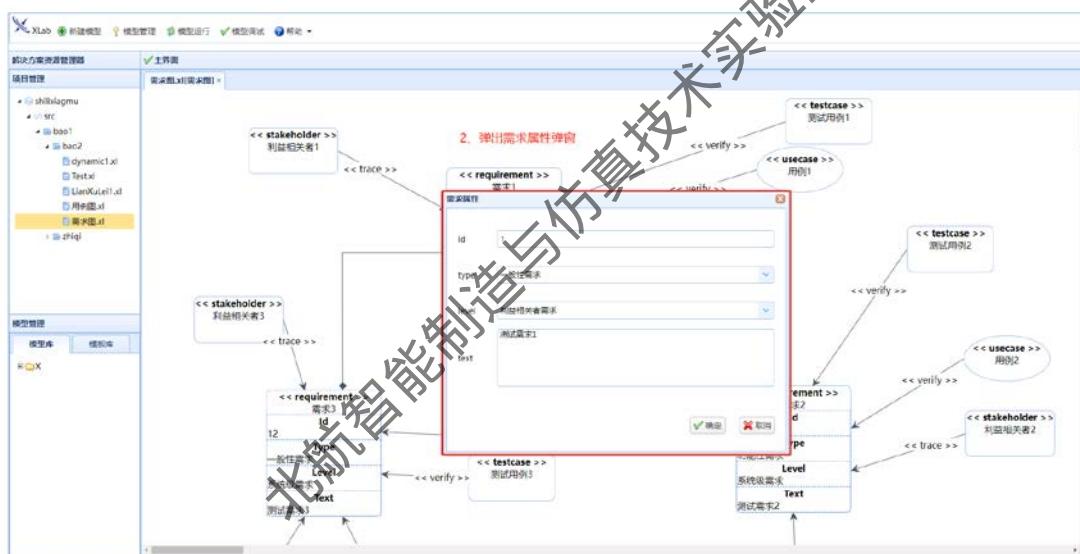
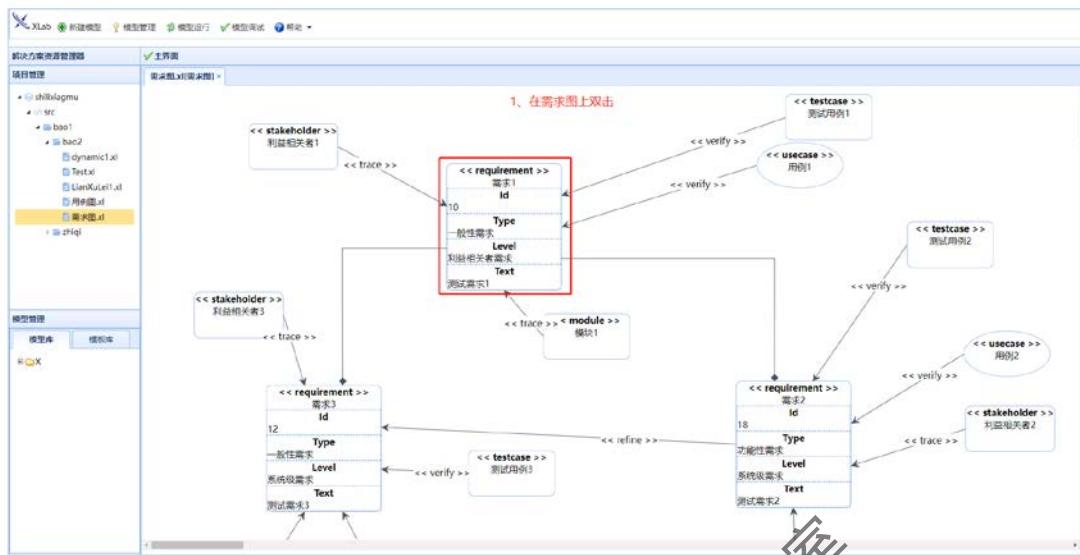
其他绩效评价材料



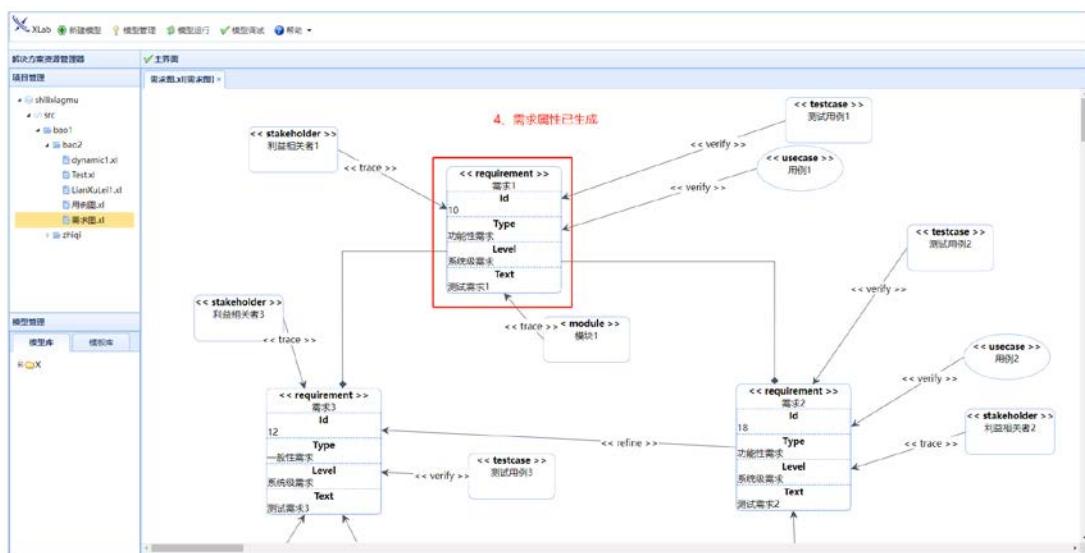
其他绩效评价材料



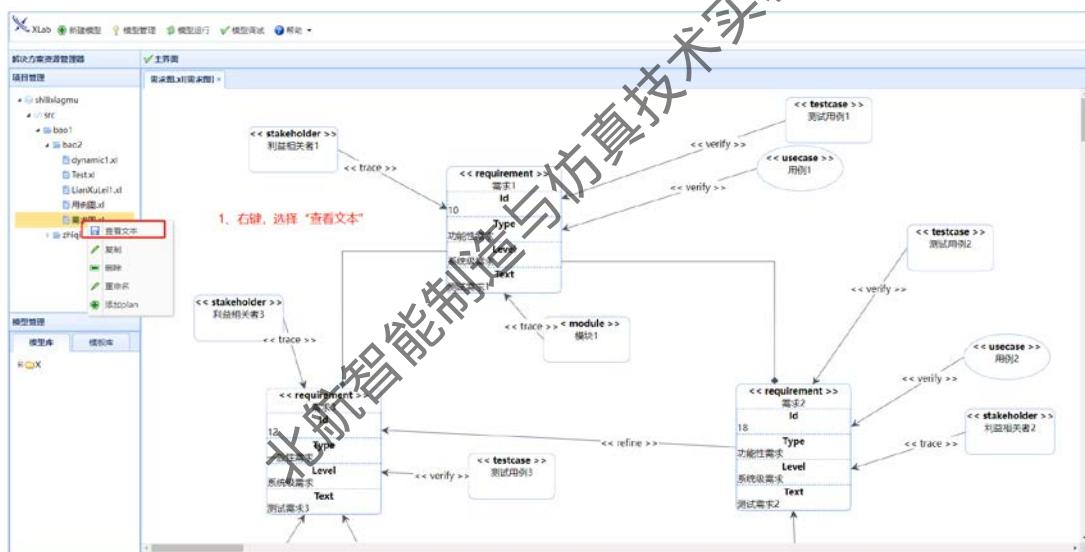
1.58. 编辑需求属性



其他绩效评价材料



1.59. 需求图-查看文本（图形到文本）



其他绩效评价材料

XLab 新建模型 模型管理 模型运行 模型测试 帮助

解决方案资源管理器

项目管理 需求图.x[需求图] 需求图.x[文本视图]

模型管理

模型库 模块库

shilixiagmu

- src
- bao1
- bao2
- dynamictl.xls
- Test.xls
- LianXulei.xls
- 用例图.xls
- 需求图.xls
- zhiqj

需求图.xls

```
1 requirement 需求1
2 id: 10;
3 type: 功能性需求;
4 level: 系统级需求;
5 text: 测试需求1;
6 end;
7 requirement 需求2
8 id: 18;
9 type: 功能性需求;
10 level: 系统级需求;
11 text: 测试需求2;
12 end;
13 requirement 需求3
14 id: 12;
15 type: 功能性需求;
16 level: 系统级需求;
17 text: 测试需求3;
18 end;
19 stakeholder: 利益相关者1;
20 end;
```

2、文本已生成

XLab 新建模型 模型管理 模型运行 模型测试 帮助

解决方案资源管理器

项目管理 需求图.x[需求图] 需求图.x[文本视图]

模型管理

模型库 模块库

shilixiagmu

- src
- bao1
- bao2
- dynamictl.xls
- Test.xls
- LianXulei.xls
- 用例图.xls
- 需求图.xls
- zhiqj

需求图.xls

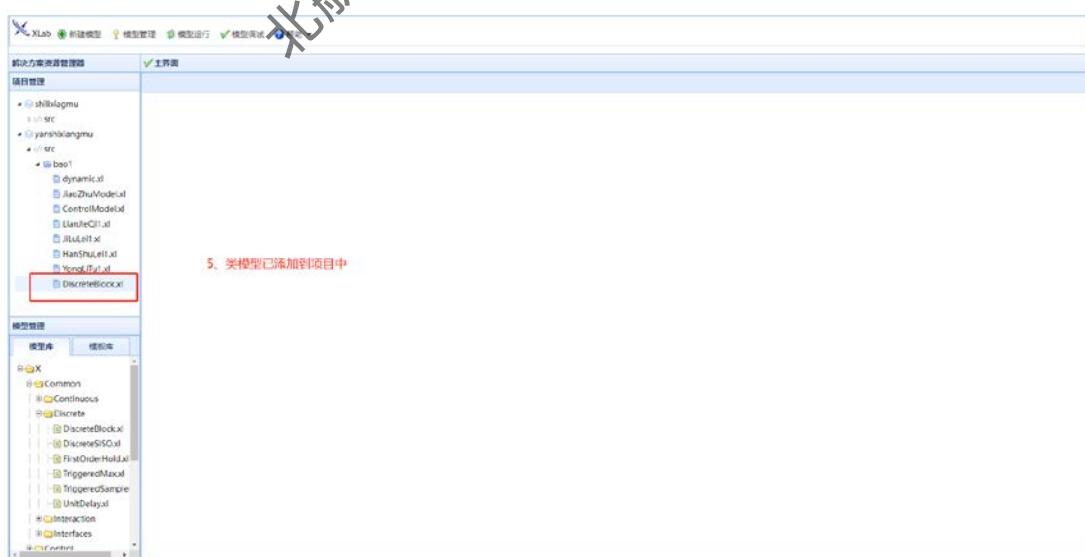
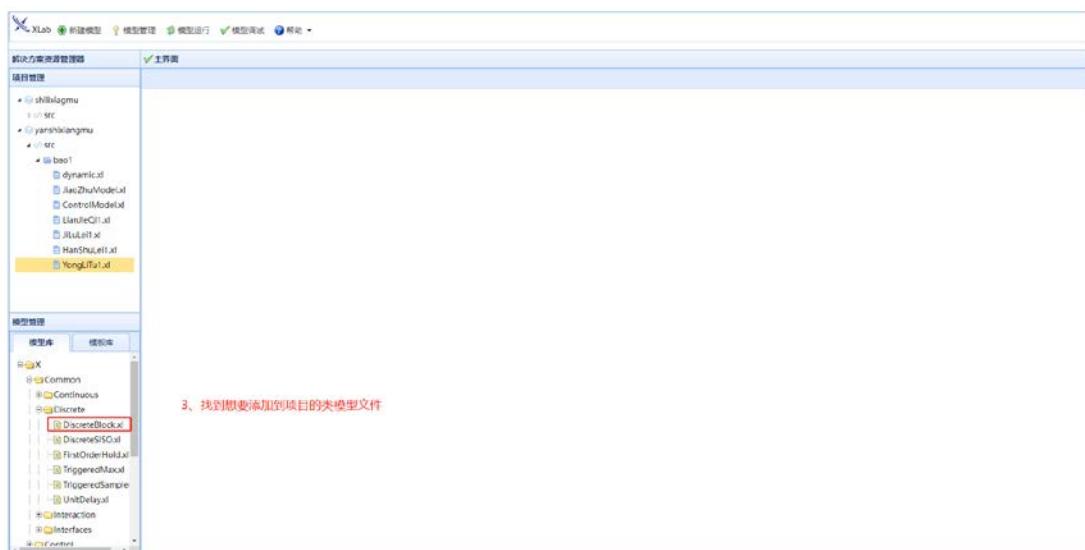
```
51 end;
52 usecase: 模块2;
53 end;
54 usecase: 模块3;
55 end;
56 testcase: 测试用例1;
57 end;
58 testcase: 测试用例2;
59 end;
60 testcase: 测试用例3;
61 end;
62 correlation: trace(模块1,需求1);
63 derive(模块2,需求2);
64 refine(模块3,需求3);
65 include(需求1,需求2);
66 include(需求1,需求3);
67 include(需求2,需求3);
68 refine(需求2,需求3);
69 verify(用例1,需求1);
70 refine(用例2,需求2);
71 trace(利益相关者1,需求1);
72 trace(利益相关者2,需求2);
73 trace(利益相关者3,需求3);
74 include(用例1,需求2);
75 include(用例1,需求3);
76 refine(用例2,需求3);
77 verify(用例1,需求1);
78 verify(用例2,需求2);
79 verify(用例3,需求3);
80 verify(测试用例1,需求1);
81 verify(测试用例2,需求2);
82 verify(测试用例3,需求3);
83 end;
```

2. 模型管理

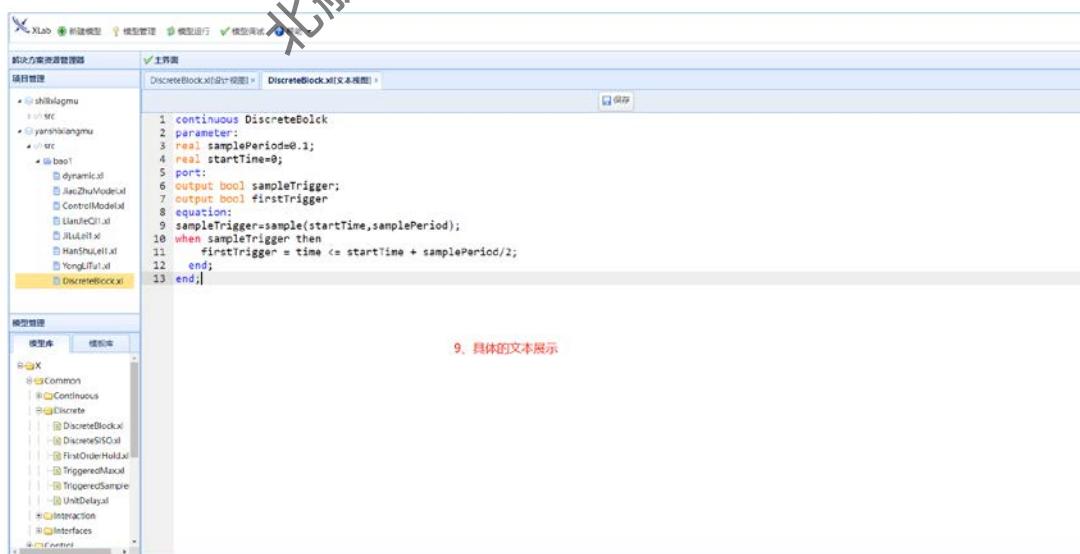
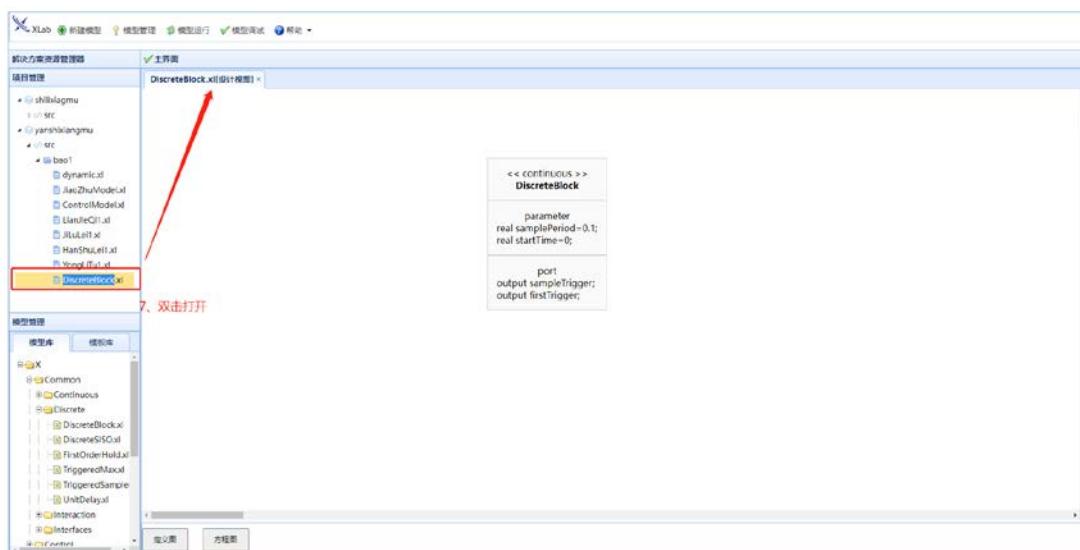
2.1. 模型库



其他绩效评价材料

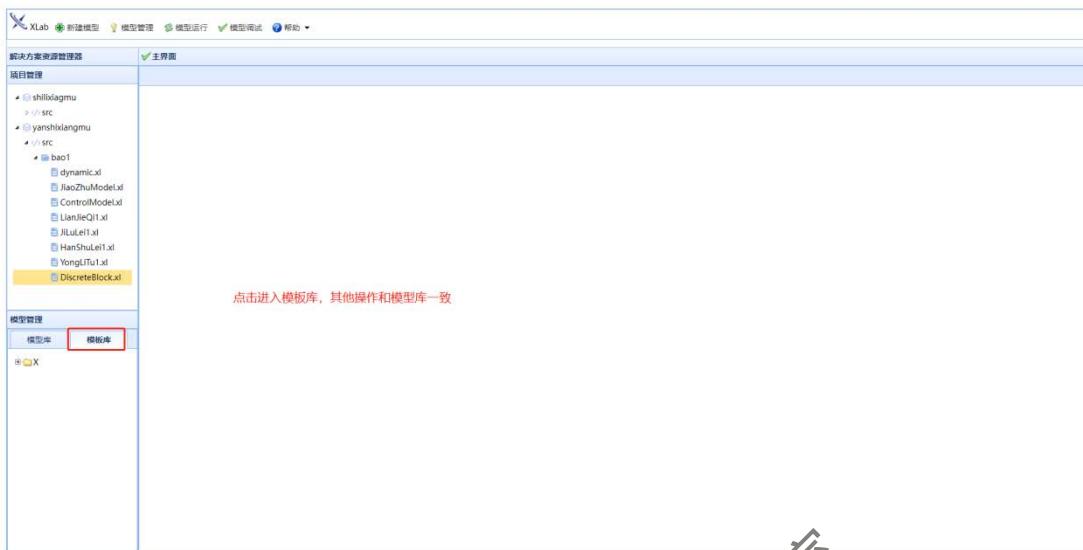


其他绩效评价材料



其他绩效评价材料

2.2. 模板库



3. 帮助

3.1. 文件下载



其他绩效评价材料

