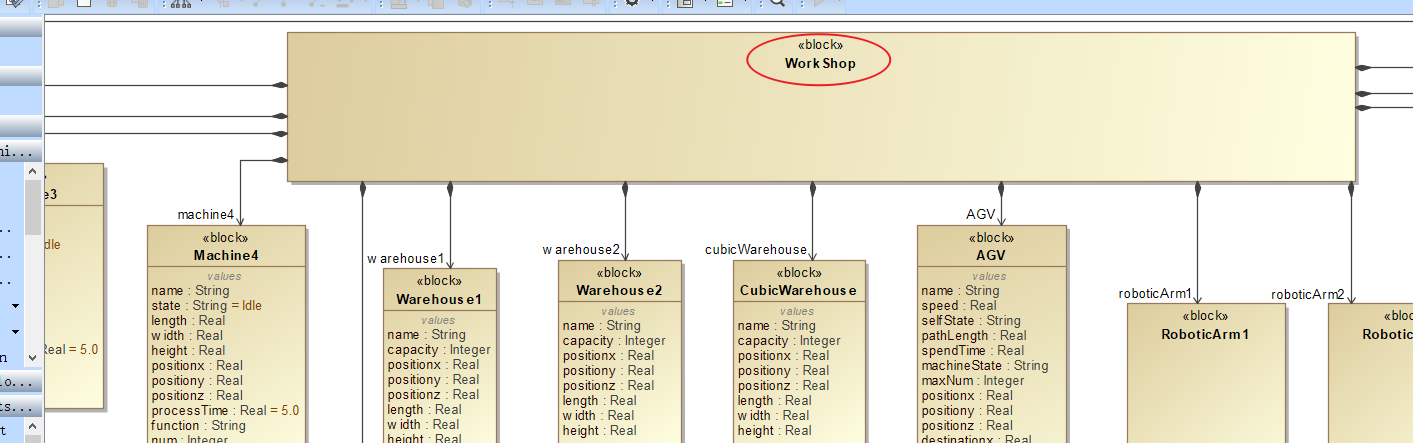
# 一、创建模型

创建模型时生成的整体xml框架如下，比如用户给项目命名为" firstModel "，仿真结束时间设置为27.0，顶层模块命名为"WorkShop"，experiment关键字下其他属性的设置允许用户选填，用户不设计的话则为默认值。



<?xml version = "1.0" encoding = "UTF-8"?>

<project name = " firstModel ">

<model>

<primaryModel>

<compositionModel name = " WorkShop ">

<subModel>

</subModel>

</compositionModel>

<structs/>

<signals>

</signals>

</primaryModel>

</model>

<initiation>

<instance name = "" init = "" template = ""/>

<inputElement>

</inputElement>

<outputElement>

</outputElement>

</initiation>

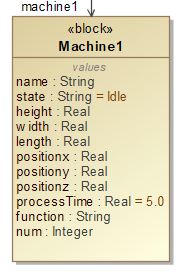
<experiment recordingStep = "0.01" lcoatorTolerance = "DBL\_MAX" solverTolerance = "1e-8" endTime = "27.0" step = "0.01"/>

</project>

# 二、拖拽实体组件

## 机器类实体组件

### 1.1.1 Machine1



拖拽一个Machine到界面，命名为Machine1，生成如下xml内容。

<elementModel name = "Machine1" discrete = "true">

<state name = "state" discrete = "true">

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "state" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "processTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "function" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "num" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

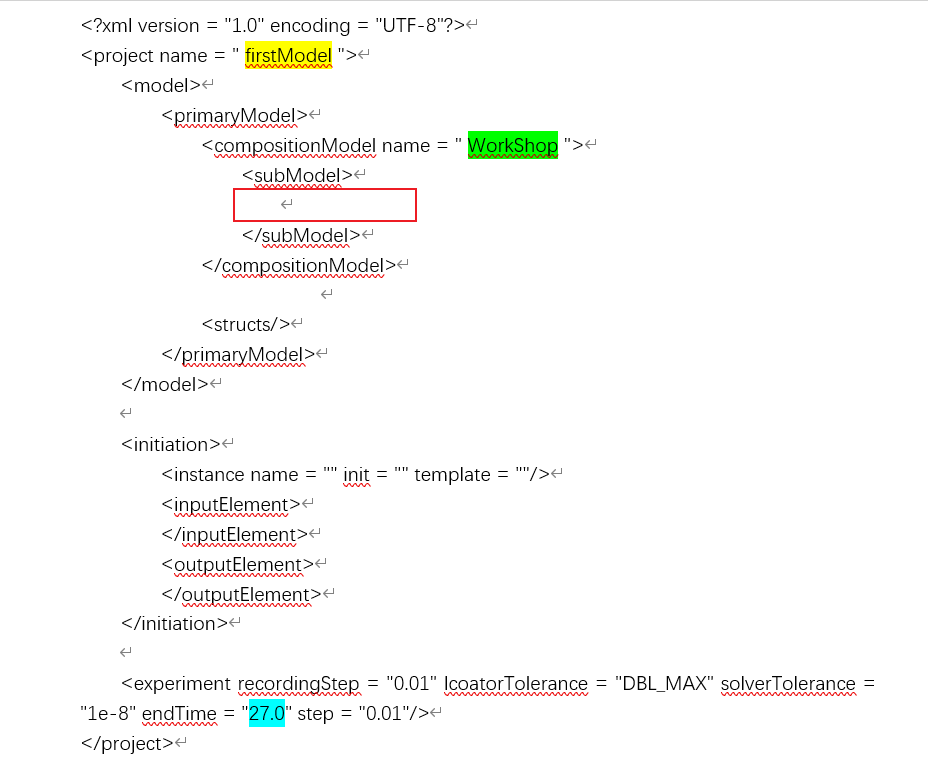
</state>

</elementModel>

同时，增加如下代码。（补充：将拖过来的实体的名字，这里就是Machine1，首字母小写）

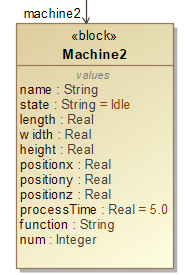
<instance name = "machine1" template = "Machine1"/>

这行代码应该放置的位置如红框所示。



下面所有增加的instance代码都放在该位置，之后再出现将不再说明需要放置的位置。

### 1.1.2 Machine2



拖拽一个Machine到界面，命名为Machine2，生成如下xml内容。

<elementModel name = "Machine2" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "state" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "processTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "function" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "num" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

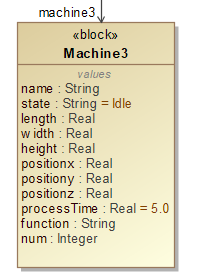
</elementModel>

同时，增加如下代码。

<instance name = "machine2" template = "Machine2"/>

放置位置见Machine1的说明。

### 1.1.3 Machine3



拖拽一个Machine到界面，命名为Machine3，生成如下xml内容。

<elementModel name = "Machine3" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "state" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "processTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "function" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "num" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

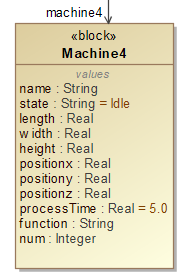
</elementModel>

同时，增加如下代码。

<instance name = "machine3" template = "Machine3"/>

放置位置见Machine1的说明。

### 1.1.4 Machine4



拖拽一个Machine到界面，命名为Machine4，生成如下xml内容。

<elementModel name = "Machine4" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "state" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "processTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "function" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "num" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

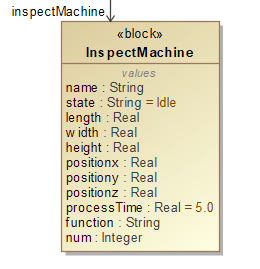
</elementModel>

同时，增加如下代码。

<instance name = "machine4" template = "Machine4"/>

放置位置见Machine1的说明。

### 1.1.5 InspectMachine



拖拽一个Machine到界面，命名为InspectMachine，生成如下xml内容。

<elementModel name = "InspectMachine" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "state" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "processTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "function" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "num" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

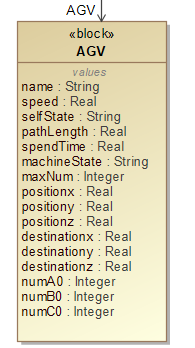
</elementModel>

同时，增加如下代码。

<instance name = " inspectMachine " template = " InspectMachine "/>

放置位置见Machine1的说明。

## 1.2 AGV



拖拽一个AGV到界面，命名为AGV，生成如下xml内容。

<elementModel name = "AGV" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "speed" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "selfState" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "pathLength" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "spendTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "machinState" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "preparing" discrete = "true" isStatic="false" multiciplity="">

<call name = "bool" value = ""/>

</state>

<state name = "maxNum" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "destinationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " destinationy " discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " destinationz " discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

</elementModel>

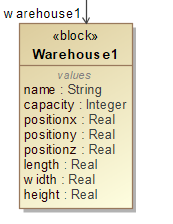
同时，增加如下代码。（说明：为了好看，将agv全部小写）

<instance name = "agv" template = "AGV"/>

放置位置见Machine1的说明。

## 1.3 仓库类实体组件

### 1.3.1 Warehouse1



<elementModel name = " Warehouse2" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "capacity" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

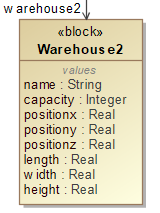
</elementModel>

同时，增加如下代码。

<instance name = " warehouse1" template = "Warehouse1"/>

放置位置见Machine1的说明。

### 1.3.2 Warehouse2



<elementModel name = "Warehouse2" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "capacity" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

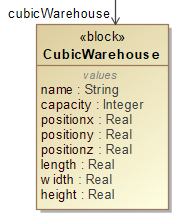
</elementModel>

同时，增加如下代码。

<instance name = " warehouse2" template = "Warehouse2"/>

放置位置见Machine1的说明。

### 1.3.3 CubicWarehouse



<elementModel name = "CubicWarehouse" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "capacity" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "height" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "preparing" discrete = "true" isStatic="false" multiciplity="">

<call name = "bool" value = "false"/>

</state>

</elementModel>

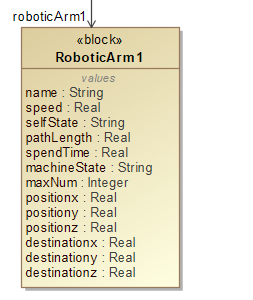
同时，增加如下代码。

<instance name = " cubicWarehouse " template = "CubicWarehouse"/>

放置位置见Machine1的说明。

## 1.4 机器臂

### 1.4.1 RoboticArm1



拖拽一个RoboticArm到界面，命名为RoboticArm1，生成如下xml内容。

<elementModel name = " RoboticArm1" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "speed" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "selfState" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "pathLength" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "spendTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "machinState" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "maxNum" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "destinationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " destinationy " discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " destinationz " discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

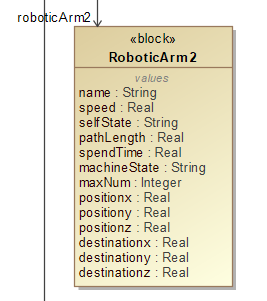
</elementModel>

同时，增加如下代码。

<instance name = "roboticArm1" template = " RoboticArm1"/>

放置位置见Machine1的说明。

### 1.4.2 RoboticArm2



拖拽一个RoboticArm到界面，命名为RoboticArm2，生成如下xml内容。

<elementModel name = " RoboticArm2" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "speed" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "selfState" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = " &quot;Idle &quot;"/>

</state>

<state name = "pathLength" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "spendTime" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "machinState" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "maxNum" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = "positionx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positiony" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "positionz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "destinationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " destinationy " discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " destinationz " discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

</elementModel>

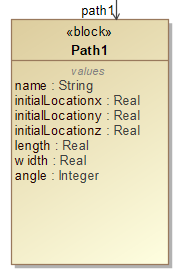
同时，增加如下代码。

<instance name = "roboticArm2" template = " RoboticArm2"/>

放置位置见Machine1的说明。

## 1.5 路径

### 1.5.1 Path1



拖拽一个Path到界面，命名为Path1，生成如下xml内容。

<elementModel name = " Path1" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "initialLocationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationy" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "angle" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

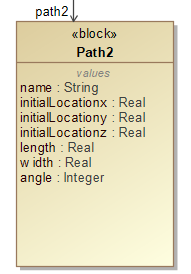
</elementModel>

同时，增加如下代码。

<instance name = " path1" template = "Path1"/>

放置位置见Machine1的说明。

### 1.5.2 Path2



拖拽一个Path到界面，命名为Path2，生成如下xml内容。

<elementModel name = " Path2" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "initialLocationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationy" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "angle" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

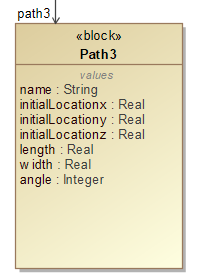
</elementModel>

同时，增加如下代码。

<instance name = " path2" template = "Path2"/>

放置位置见Machine1的说明。

### 1.5.3 Path3



拖拽一个Path到界面，命名为Path3，生成如下xml内容。

<elementModel name = " Path3" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "initialLocationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationy" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "angle" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

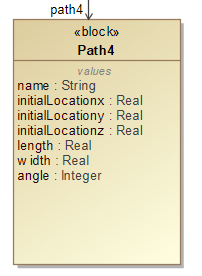
</elementModel>

同时，增加如下代码。

<instance name = " path3" template = "Path3"/>

放置位置见Machine1的说明。

### 1.5.4 Path4



拖拽一个Path到界面，命名为Path4，生成如下xml内容。

<elementModel name = " Path4" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "name" discrete = "true" isStatic="false" multiciplity="">

<call name = "string" value = ""/>

</state>

<state name = "length" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "width" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "initialLocationx" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationy" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = " initialLocationz" discrete = "true" isStatic="false" multiciplity="">

<call name = "double" value = ""/>

</state>

<state name = "angle" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

</elementModel>

同时，增加如下代码。

<instance name = " path4" template = "Path4"/>

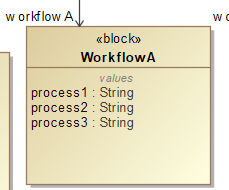
放置位置见Machine1的说明。

# 三、设计订单流程

## 3.1 添加A类订单

* 流程

machine1:5 – machine2:7 - complete



生成如下xml

<elementModel name = "WorkflowA" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "process1" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;machine1:5 &quot;"/>

</state>

<state name = "process2" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;machine2:7 &quot;"/>

</state>

<state name = "process3" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;complete &quot;"/>

</state>

</elementModel>

同时，增加如下代码。

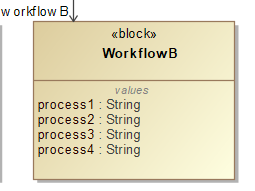
<instance name = " workflowA " template = "WorkflowA"/>

放置位置见Machine1的说明。

## 3.2 添加B类订单

* 流程

machine3:6 – machine4:2 -check - complete



生成如下xml

<elementModel name = "WorkflowB" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "process1" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;machine3:6 &quot;"/>

</state>

<state name = "process2" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;machine4:2 &quot;"/>

</state>

<state name = "process3" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;check &quot;"/>

</state>

<state name = "process4" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;complete &quot;"/>

</state>

</elementModel>

同时，增加如下代码。

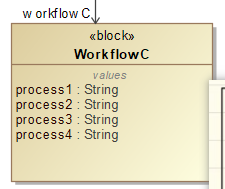
<instance name = " workflowB " template = "WorkflowB"/>

放置位置见Machine1的说明。

## 3.3 添加C类订单

* 流程

machine1:3 – check – machine3:7 - complete



生成如下xml

<elementModel name = "WorkflowC" discrete = "true">

<state name = "state" discrete = "true" >

<call name = "int" value = "0"/>

</state>

<state name = "process1" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;machine1:3 &quot;"/>

</state>

<state name = "process2" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;check &quot;"/>

</state>

<state name = "process3" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;machine3:7 &quot;"/>

</state>

<state name = "process4" discrete = "" multiplicity = "" isStatic = "false">

<call name = "string" value = " &quot;complete &quot;"/>

</state>

</elementModel>

同时，增加如下代码。

<instance name = " workflowC " template = "WorkflowC"/>

放置位置见Machine1的说明。

# 四、设计每类订单对应毛坯数

这里的意思是，用户设计好了每个订单的数目，比如A:5 – B:4 – C:3，会同时增加/更改4.1基础设计以及4.2收发数据设计里关于xml内容的描述。

## 4.1 基础设计

A:5 – B:4 – C:3

定位到AGV所在的elementModel内，为AGV新增属性numA0、numB0、numC0，对应的xml代码如下。（这里先不赋值，等到AGV到达立体仓库之后再赋值）

<state name = " numA0" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = " numB0" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

<state name = " numC0" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = ""/>

</state>

用户设计完成之后更改CubicWarehouse里的preparing属性值为true。更改后的部分属性代码如下。

<state name = "preparing" discrete = "true" isStatic="false" multiciplity="">

<call name = "bool" value = "true"/>

</state>

## 4.2 收发数据设计

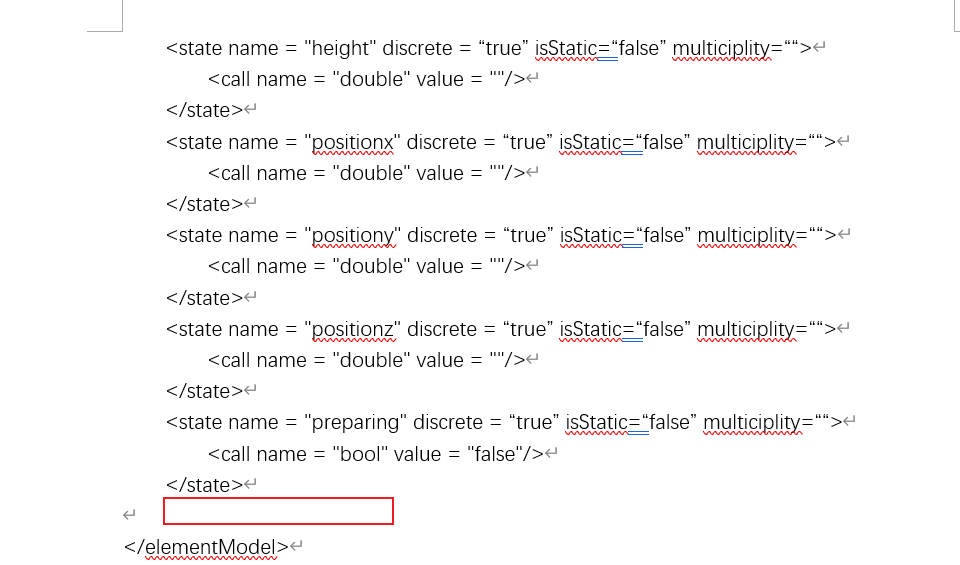
三部分组成：①增加端口②建立连接③创建信号

1.增加端口

为CubicWarehouse增加output端口，为AGV增加input端口，代码分别如下：

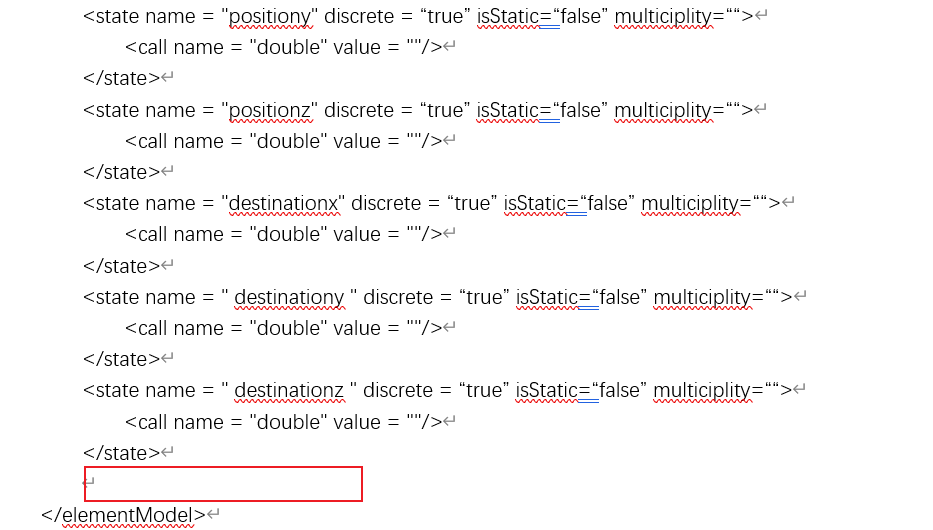
<output name="out1" discrete="true"/>

在CubicWarehouse的elementModel里，其放置位置如下



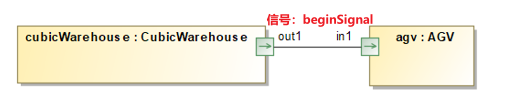
<input name="in1" discrete="true"/>

在AGV的elementModel里，其放置位置如下



2.建立连接

需要CubicWarehouse向AGV发送数据，告诉AGV你可以出发去立体仓库了，端口名字及连线如下。



需要生产的xml代码如下。

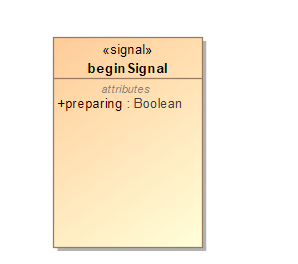
<couple from = "cubicWarehouse" fromOutput = "out1" to = "agv" toInput = "in1"/>

放置位置如下图红色方框所示。



3.创建信号

已经把两个端口连接起来了，现在需要定义发送的内容，需要写在信号里面，block如下图所示。



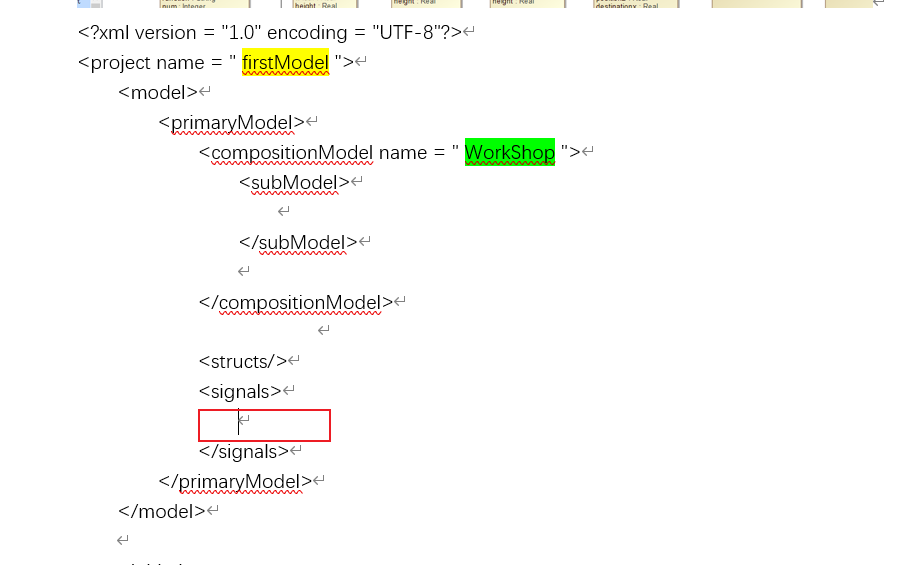
对应需要生成的xml代码如下图所示。

<signal name = "beginSignal">

<call name = "preparing" type = "bool"/>

</signal>

该部分xml放置的位置如下图红色方框所示。



# 五、流程开始

## 5.1 AGV接收到信号，准备前往立体仓库

定位到AGV所在的elementModel内，为AGV属性numA0、numB0、numC0赋值，对应的xml代码如下。

<state name = " numA0" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = "5"/>

</state>

<state name = " numB0" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = "4"/>

</state>

<state name = " numC0" discrete = "true" isStatic="false" multiciplity="">

<call name = "int" value = "3"/>

</state>