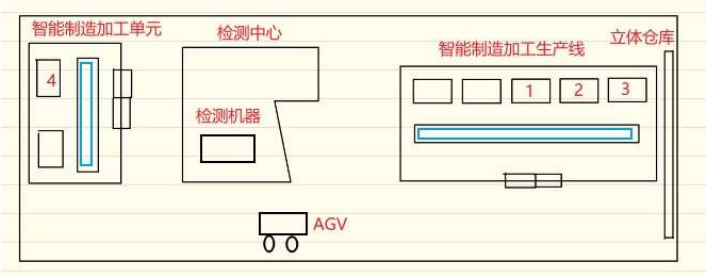


“检测”在xml中用英文check来表示  
“加工完成”在xml中用英文complete来表示

零件分类（不按产线，按流程）  
A类：machine1: 5-machine2: 7-加工完成  
B类：machine3: 6-machine4: 2-检测-加工完成  
C类：machine1: 3-检测-machine3: 7-加工完成  
AGV block

maxNum (integer): AGV上最多可放多少“中间件”  
numA0(integer): 记录AGV上放了多少个A0  
numB0(integer): 记录AGV上放了多少个B0  
... ..

A0: A类订单的毛坯  
A1: A0经过第一道工序之后产生的“中间件”  
A2: A0经过2道工序之后产生的“中间件”；或者A1经过第二道工序之后产生的“中间件”  
... ..  
B0 B1 ... ..  
C0 C1 ... ..



```
string s = "numA0";
string temp = s.substr(3,4);
cout << temp; //A0
```

订单举例：  
订单A的内容记录在workflowA里，比如用户动态定义了订单A的内容如下：  
machine1:5; machine2:7; complete  
该订单有3道工序（把加工完成当成最后一道工序），每道工序对应一个String类型的属性，我对应的界面的block如下：



numA内容：“A\_0: 5”  
numB内容：“B\_0: 4”

上述block对应的xml内容如下：

```
<elementModel name = "WorkflowA" type = "discrete">
  <state name = "state" type = "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>
```

创建订单A，则对应block名称为WorkflowA；创建订单B，则对应block名称为WorkflowB ... ..  
每类订单的第一道工序取名为process1,第二道工序process2 ... ..