# 机器人编程及可视化解决方案

1. **发明背景及最接近的现有技术**
   1. 发明背景（简述，可以结合附图加以说明）：

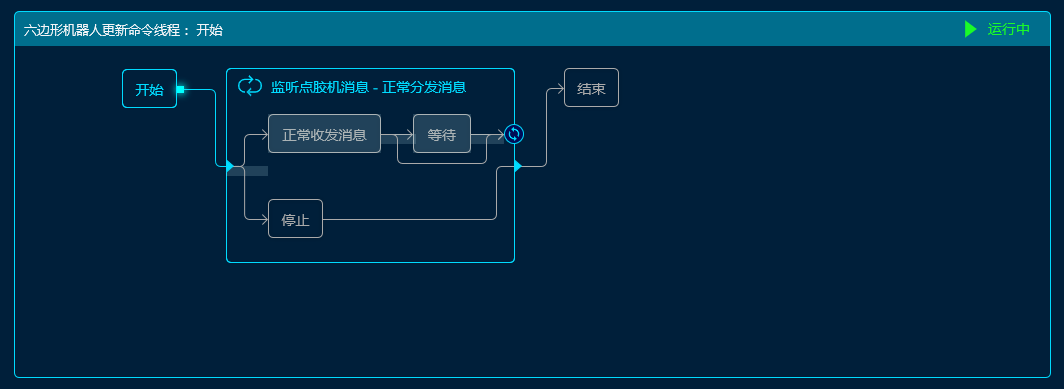
现阶段用户在使用机器人时除了自己编写机器人程序外还需要另外实现GUI来展示机器人的运行状态，并且在编码构思的过程中只能通过其他绘图软件描述机器人状态机，这个状态机和机器人程序又是独立存在的。

现有最接近技术

暂无查到接近资料

* 1. 现有技术的缺点

1. **发明技术方案的详细阐述（即发明内容，建议结合结构图、流程图、原理框图、电路图、时序图等进行说明）：**

2.1 本发明所要解决的技术问题（发明目的）：

1) 传统状态机无法转化为机器人运行代码。

2) 编程平台无法查看机器人的状态和工艺流程状态。

3) 不同的机器人厂商所采用的编程语言不一样，用户学习成本很高。

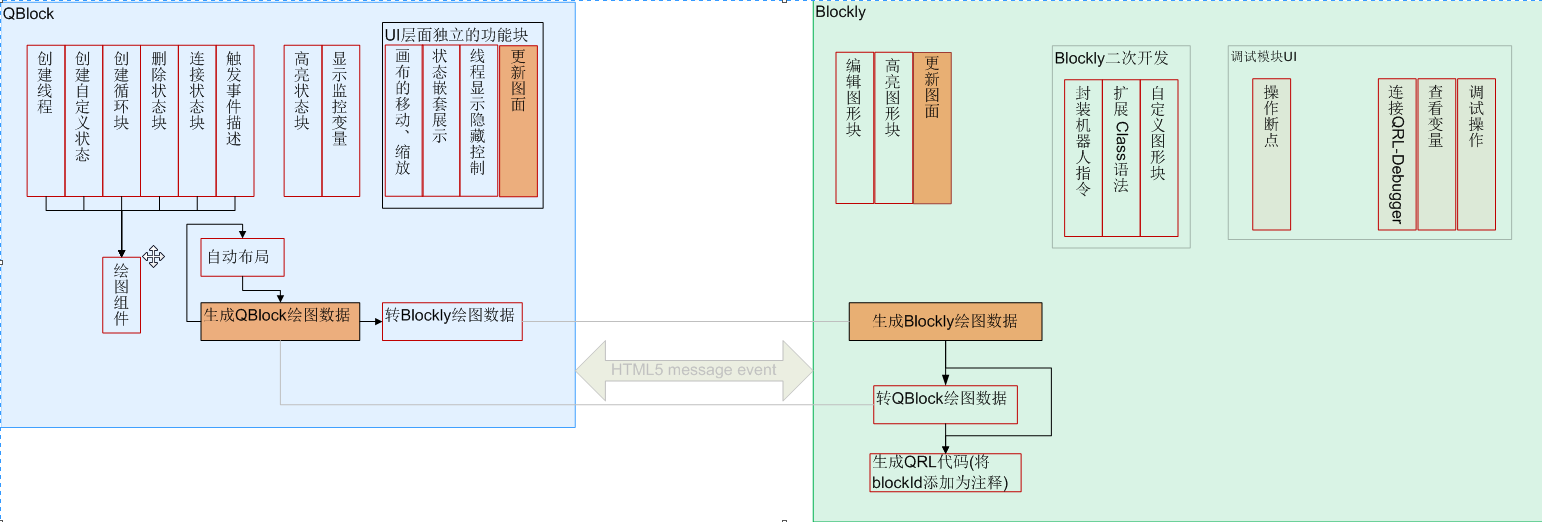
4) Blockly图形化编程中Class语法块。

5）对Blockly界面中的部分图形块分类排列，方便查看和调试。

6）对状态图实现自动布局。

7）可对接其他厂商的机器人实现离线编程。

2.2 本发明完整的技术方案（需详细阐述）

本发明采用浏览器/服务器架构模式，将应用处理程序和web资源都放在机器人端，以web服务器的形式提供给用户访问，用户在浏览器端输入机器人IP地址即可连接机器人。本发明提供了状态图与Blockly两种图形编辑功能，状态图以连线的形式表示机器人各个状态间的关系，相较传统的图形程序，状态图提供了更加清晰的表示逻辑关系的方法，状态图不仅能转化成图形程序（图形程序再转成代码）而且能实时显示当前程序所处的状态，状态图的详细技术介绍参考要点1说明。

要点1说明：

本发明基于Vue.js框架实现绘制状态图的界面，状态图大致分为绘图工具区域和画布区域，绘图区域有线程、状态、循环三类组件，利用html5的drag事件将绘图组件拖拽到画布区域，在画布元素上侦听drog事件，当侦听到有组件在画布上释放时创建一个该组件的实例。创建2个状态分别命名为“状态1”和“状态2”，在状态1的输出节点上按下鼠标开始绘制连线，在状态2的输入节点上松开鼠标，调用javascript在画布中创建svg线条组件。鼠标双击状态可以对其重命名，在线条上点击鼠标右键选择添加事件描述，这样就可生成对整个工艺流程的状态描述。以线程为单位的所有组件最终就形成了状态图。因为采用了Vue框架，画布中所有元素都是由数据驱动完成绘制的，所以保存这些组件的json（文中json特指状态图数据）数据再次加载时就可以恢复画布中的元素。json的数据结构大致如下：

|  |  |  |
| --- | --- | --- |
| 对象名称 | 属性名称 | 属性描述 |
| 线程 | name | 线程名称 |
| width | 宽度 |
| height | 高度 |
| stateAry | 状态集合 |
| lineAry | 连线集合 |
| 状态 | name | 状态名称 |
| stateId | 状态ID |
| stateType | 状态类型 |
| x | 状态在线程组件中的水平位置 |
| y | 状态在线程组件中的垂直位置 |
| inputAry | 进入该状态的触发事件集合 |
| outputAry | 离开该状态的触发事件集合 |
| children | 子状态集合 |
| 连线 | lineId | 连线ID |
| d | 连线的绘制路径 |
| startPoint | 连线的起点坐标 |
| endPoint | 连线的终点坐标 |
| startState | 连线的上一状态（触发事件发生的状态） |
| endState | 连线的下一状态（触发事件发生后进入的状态） |

Blockly图面中有着和状态图一一对应的组件，切换到Blockly图面时，将这份json数据转为相应的xml（文中xml特指Blockly图面数据）数据，然后加载xml数据就能在Blockly图面生成对应状态图所描述的逻辑结构，此时，在Blockly界面可定义所有状态组件，往这些状态组件里面拖拽恰当的图形块就形成了完整的程序。Blockly界面中的每个图形块都有其固定结构的代码。最终生成代码时将动态的代码信息填充到固定结构中形成完整的程序。

在Blockly界面中调整状态组件的结构，再切换到状态图时，对比Blockly图面的xml数据和状态图的json数据，对应xml中的改动部分更新json数据， 状态图会自动同步更新，由于状态图和Blockly图面中组件位置没有逻辑关系，所以在切换到状态图时需要采用我们的自动布局算法对状态组件进行排列，然后可手动调整，手动调整后QBlock会将位置信息记录到json数据中。

要点2说明：

在状态图中所有的状态描述都是由用户自定义的，这些状态的内部代码就是一些控制机器人的图形程序，在每一个状态块的代码中都有一个钩子函数用于输出这个状态块的id，当程序执行到某个状态时，将这个id反馈到浏览器端，然后通过javascript高亮界面上对应的图形，这样就可以展示出机器人的状态，配合丰富渐变的动画效果也就是整个工艺流程的形象呈现。

要点3说明：

Blockly图形化编程通过图形化编程完成程序设计，在Blockly中有一个类似语言转换器的工具箱，可以将图形化编程语言转化成多种编程语言代码。用图形化编程方式去理解多种程序语言。将常用的逻辑结构转化成图形块，给每个图形块定义多种语言的代码结构，屏蔽了语言本身的语法，使编程变得简洁、形象。

要点4说明：

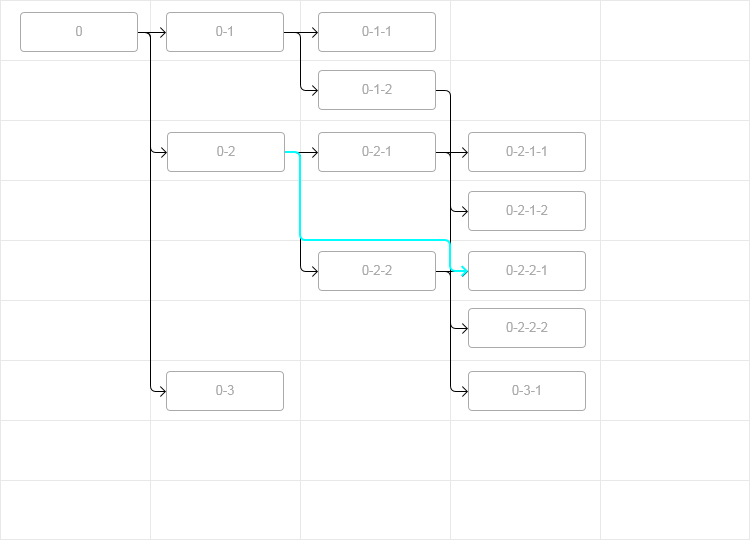
在Blockly图形化编程中，Google本身不提供Class语法块,也不提供局部变量。我们会创建一个Class定义块和Class调用块，这种块有点类似于函数块，不同的是这个Class调用块会根据用户所选的属性或方法动态改变其内部的输入项，比如Person类有jobs和hobbys2个属性，选择jobs后则后面的可选项是teacher和farmer，选择hobbys后则可选项动态变为singing和dancing。实现Class语法后将会为用户编码带来极大的方便。

要点5说明：

Google Blockly提供的图形界面是无序的堆在一起的，用户查找某个图形块时只能凭编辑时的记忆去寻找，这是非常不方便的。在QBlock软件中，我们对此进行了优化，将状态的定义、函数的定义等放到一个列表block中，并按分类横向排列，这样就可以方便的查找和调试，也避免了一些无序图形块重叠在一起。

要点6说明：

将整个画布划分为矩形表格，如下图1-1所示：



1‑1

现在状态块的宽度和高度是设置的固定值，所以可以将这个单元格的大小确定，我们将叶子节点的状态高度设为1，每个状态的高度等于其所有子节点高度之和，这样就可以在遍历时确定每个状态的纵坐标（依据状态的高度），根据节点的深度确定状态的横坐标。连接2个状态时，根据目标状态的纵坐标查询其左边是否有其他状态会遮挡连线（查询当前所有纵坐标等于目标状态纵坐标的状态，如果这些状态的横坐标介于开始状态的横坐标与目标状态的横坐标之间，则说明此状态会遮挡最短连线），如果有遮挡的状态则在垂直方向先画到目标状态的纵坐标，然后沿水平方向画线，最后连接至目标状态，如图高亮的连线所示。

要点7说明：

QBlock 资源是放在机器人端然后通过浏览器进行访问的，也可以提供离线版，将QBlock资源放在客户端，然后通过浏览器访问。因为Blockly本身是支持JavaScript、Python、PHP、Lua、Dart等编程语言的，我们也可以在此基础上再扩展其他语言，对其他厂商的机器人指令进行封装，以图形块的形式提供出来，这样就实现了对接其他厂商机器人编程的功能。

2.3、本发明带来的有益效果：

此编程软件与其他图形化编程软件相比，既能作为编程软件使用也能作为可视化运维的平台，机器人的状态、工艺流程会以生动的动画展示给用户，当前是否有问题，问题是否严重等一目了然。因为我们的状态图都是自定义描述，这样就能保证准确的表示工艺流程，当程序执行到对应的状态时，调试器会返回组件的id等信息，web端更新相应的DOM节点及其样式就能达到高亮当前状态，给用户视觉反馈。这样用户不必再去额外的实现一套GUI软件。

2.4、替代方案（是否有替代方案能达到本发明效果）：

暂无

2.5、本发明关键点及预保护点：

**三、其他有助于理解本发明的资料**

[CN110494843A](https://share-analytics.zhihuiya.com/view/1e3875d8-70e0-45cc-bd02-c6c9fb036358)

[US8640100](https://share-analytics.zhihuiya.com/view/98d0cab4-fd53-4da8-b99c-04eecb4ec87f)

[JPWO2019003524A1](https://share-analytics.zhihuiya.com/view/548b9a3d-0327-497d-821e-8a4573a767f8)

[CN110494843A](https://share-analytics.zhihuiya.com/view/1e3875d8-70e0-45cc-bd02-c6c9fb036358)