(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111767406 B
(45)授权公告日 2022.08.30

(21)申请号 201910257222.4

(22)申请日 2019.04.01

(65)同一申请的已公布的文献号
申请公布号 CN 111767406 A

(43)申请公布日 2020.10.13

(73)专利权人 杭州电子科技大学
地址 310018 浙江省杭州市下沙高教园区二号路1158号

(72)发明人 邬惠峰 秦飞巍 安雅萌 陈佰平

(74)专利代理机构 北京同立钧成知识产权代理有限公司 11205
专利代理师 朱颖 刘芳

(51)Int.Cl.
G06F 16/36(2019.01)

(56)对比文件
US 2006173868 A1,2006.08.03
US 2013204910 A1,2013.08.08
US 2016292304 A1,2016.10.06
weibin dai等.“Transformation from PLC to Distributed Control using Ontology Mapping”.《IEEE 10th International Conference on Industrial Informatics》.2012,

审查员 袁冠群

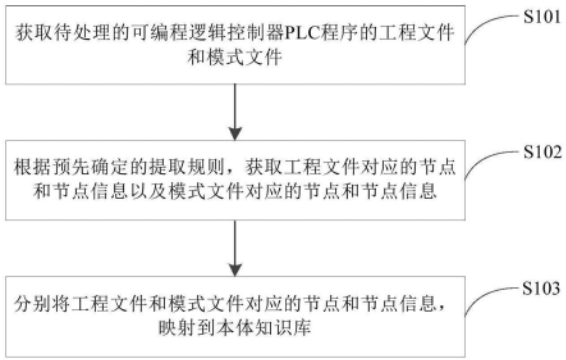
权利要求书2页 说明书14页 附图1页

(54)发明名称

PLC工程的知识表示方法及装置

(57)摘要

本发明实施例提供一种PLC工程的知识表示方法及装置,该方法包括:获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件,并根据预先确定的提取规则,获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息,分别将所述工程文件和所述模式文件对应的节点和节点信息,映射到本体知识库,实现了对不同平台PLC程序进行统一的知识表示,为PLC程序在不同硬件平台间的转移和重复使用的便利性提供了可能。



CN 111767406 B

1.一种PLC工程的知识表示方法,其特征在于,应用于终端设备,所述方法包括:

获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件;

根据预先确定的提取规则,获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息;所述节点信息包括节点属性和节点层级关系;

分别将所述工程文件和所述模式文件对应的节点和节点信息,映射到本体知识库,所述本体知识库用于表示和存储待处理的PLC程序的知识;

构建统一的PLC初始模型,用于为不同知识库提供交互的中介,其中构建过程包括:列举PLC工程领域的重要术语和词汇;确定PLC本体的类及类之间的层次关系,定义类之间的层次关系由自顶向下的方法实现;定义PLC本体中类的属性,OWL语言包括对象属性和数据属性两种属性,一个类包含多个属性;定义PLC本体中属性的限定,一个属性具有属性取值的类型、容许的取值、取值基数多个限定;

所述分别将所述工程文件和所述模式文件对应的节点和节点信息,映射到本体知识库,包括:

将所述工程文件和所述模式文件对应的节点和节点信息,使用网络本体语言OWL进行表示和存储。

2.根据权利要求1所述的方法,其特征在于,将所述工程文件和所述模式文件对应的节点和节点信息,使用OWL进行表示和存储,包括:

通过Manchester OWL对所述工程文件和所述模式文件对应的节点和节点信息进行描述和保存;

再将Manchester OWL转换到OWL。

3.根据权利要求1所述的方法,其特征在于,所述根据预先确定的提取规则,获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息,包括:

通过正则表达式在所述工程文件和所述模式文件中进行匹配,获取所述工程文件和所述模式文件的节点和节点信息;

或者,

通过文档对象模型DOM获取所述工程文件和所述模式文件的节点和节点信息。

4.一种PLC工程的知识表示装置,其特征在于,所述装置包括:

获取模块,用于获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件;

解析模块,用于获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息;所述节点信息包括节点属性和节点层级关系;

处理模块,用于分别将所述工程文件和所述模式文件对应的节点和节点信息,映射到本体知识库,所述本体知识库用于表示和存储待处理的PLC程序的知识;

构建统一的PLC初始模型,用于为不同知识库提供交互的中介,其中构建过程包括:列举PLC工程领域的重要术语和词汇;确定PLC本体的类及类之间的层次关系,定义类之间的层次关系由自顶向下的方法实现;定义PLC本体中类的属性,OWL语言包括对象属性和数据属性两种属性,一个类包含多个属性;定义PLC本体中属性的限定,一个属性具有属性取值的类型、容许的取值、取值基数多个限定;

所述处理模块具体用于将所述工程文件和所述模式文件对应的节点和节点信息,使用网络本体语言OWL进行表示和存储。

5.根据权利要求4所述的装置,其特征在于,所述处理模块具体用于:

通过Manchester OWL对所述工程文件和所述模式文件对应的节点和节点信息进行描述和保存;

再将Manchester OWL转换到OWL。

6.根据权利要求4所述的装置,其特征在于,所述获取模块具体用于:

通过正则表达式在所述工程文件和所述模式文件中进行匹配,获取所述工程文件和所述模式文件的节点和节点信息;

或者,

通过文档对象模型DOM获取所述工程文件和所述模式文件的节点和节点信息。

7.一种终端设备,其特征在于,包括:至少一个处理器和存储器;

所述存储器存储计算机执行指令;

所述至少一个处理器执行所述存储器存储的计算机执行指令,使得所述至少一个处理器执行如权利要求1至3任一项所述的PLC工程的知识表示方法。

8.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如权利要求1至3任一项所述的PLC工程的知识表示方法。

PLC工程的知识表示方法及装置

技术领域

[0001] 本发明实施例涉及工业控制领域，尤其涉及一种可编程逻辑控制器(Programmable Logic Controller,PLC)工程的知识表示方法及装置。

背景技术

[0002] 随着工业控制领域的不断发展，自动化程度越来越高，工业领域的控制系统也越来越复杂，在同一个系统中集成有多个来自不同供应商的产品，并且，在不同的工作阶段和不同的生产需求下，需要使用不同的开发软件。

[0003] 目前，不同厂家生产的可编程逻辑控制器(Programmable Logic Controller，PLC)，通常是基于不同的软件平台开发的，各产品间不具备可交互性。

发明内容

[0004] 本发明实施例提供一种PLC工程的知识表示方法及装置，以实现对基于不同平台开发的PLC程序进行统一的语义表示，为PLC程序在不同硬件平台间的转移和重复使用的便利性提供了可能。

[0005] 第一方面，本发明实施例提供一种PLC工程的知识表示方法，应用于终端设备，所述方法包括：

[0006] 获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件；

[0007] 根据预先确定的提取规则，获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息；所述节点信息包括节点属性和节点层级关系；

[0008] 分别将所述工程文件和所述模式文件对应的节点和节点信息，映射到本体知识库，所述本体知识库用于表示和存储待处理的PLC程序的知识。

[0009] 可选的，在所述分别将所述工程文件和所述模式文件对应的节点和节点信息，映射到本体知识库，包括：

[0010] 将所述工程文件和所述模式文件对应的节点和节点信息，使用网络本体语言(Ontology Wed Language,OWL)进行表示和存储。

[0011] 具体的，将所述工程文件和所述模式文件对应的节点和节点信息，使用OWL语言进行表示和存储，包括：

[0012] 通过Manchester OWL对所述工程文件和所述模式文件对应的节点和节点信息进行描述和保存；

[0013] 再将Manchester OWL转换到OWL。

[0014] 可选的，所述根据预先确定的提取规则，获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息，包括：

[0015] 通过正则表达式在所述工程文件和所述模式文件中进行匹配，获取所述工程文件和所述模式文件的节点和节点信息；

[0016] 或者，

[0017] 通过文档对象模型DOM获取所述工程文件和所述模式文件的节点和节点信息。

[0018] 第二方面,本发明实施例提供一种PLC工程的知识表示装置,所述装置包括:

[0019] 获取模块,用于获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件;

[0020] 解析模块,用于根据预先确定的提取规则,获取所述工程文件对应的节点和节点信息以及所述模式文件对应的节点和节点信息;所述节点信息包括节点属性和节点层级关系;

[0021] 处理模块,用于分别将所述工程文件和所述模式文件对应的节点和节点信息,映射到本体知识库,所述本体知识库用于表示和存储待处理的PLC程序的知识。

[0022] 可选的,所述处理模块具体用于将所述工程文件和所述模式文件对应的节点和节点信息,使用OWL进行表示和存储。

[0023] 在一种具体的实现方式中,所述处理模块具体用于:

[0024] 通过Manchester OWL对所述工程文件和所述模式文件对应的节点和节点信息进行描述和保存;

[0025] 再将Manchester OWL转换到OWL。

[0026] 在一种具体的实现方式中,所述获取模块具体用于:

[0027] 通过正则表达式在所述工程文件和所述模式文件中进行匹配,获取所述工程文件和所述模式文件的节点和节点信息;

[0028] 或者,

[0029] 通过文档对象模型DOM获取所述工程文件和所述模式文件的节点和节点信息。

[0030] 第三方面,本发明实施例提供一种终端设备,包括:至少一个处理器和存储器;

[0031] 所述存储器存储计算机执行指令;

[0032] 所述至少一个处理器执行所述存储器存储的计算机执行指令,使得所述至少一个处理器执行如第一方面所述的PLC工程的知识表示方法。

[0033] 第四方面,本发明实施例提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如第一方面所述的PLC工程的知识表示方法。

[0034] 本发明实施例提供一种PLC工程的知识表示方法及装置,该方法通过获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件,并根据预先确定的提取规则,获取工程文件对应的节点和节点信息以及模式文件对应的节点和节点信息,分别将该工程文件和模式文件对应的节点和节点信息,映射到本体知识库,该本体知识库用于表示和存储待处理的PLC程序的知识,实现了对不同平台PLC程序进行统一的知识表示,为PLC程序在不同硬件平台间的转移和重复使用的便利性提供了可能。

附图说明

[0035] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其它的附图。

[0036] 图1为本发明实施例提供的PLC工程的知识表示方法实施例一的流程示意图;

[0037] 图2为本发明实施例提供的PLC工程的知识表示装置的结构示意图；

[0038] 图3为本发明实施例提供的终端设备的硬件结构示意图。

具体实施方式

[0039] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例，都属于本发明保护的范围。

[0040] 可编程逻辑控制器(Programmable Logic Controller,PLC)作为设备和装置的控制器，除了传统的逻辑控制、顺序控制、运动控制、安全控制功能之外，还承担着工业4.0和智能制造赋予的新任务，例如应用软件编程的平台化、进一步发展工程设计的自动化和智能化等。而传统的工业控制编程语言在不同的控制需求和工作阶段要用不同的开发软件，频繁的软件切换，将导致人力资源成本高、效率低下。

[0041] 本方案中的PLC工程的知识表示方法，主要应用于终端设备，包括个人计算机PC、笔记本、服务器等具备处理能力的终端设备，该方法通过构建PLC本体模型实现对不同PLC程序的归一化表示，为多种PLC产品以及PLC程序能够更加便利的交互使用，提供了可能。下面通过几个具体实施例对该方案进行详细说明。

[0042] 图1为本发明实施例提供的PLC工程的知识表示方法实施例一的流程示意图，如图1所示，该方法包括：

[0043] S101：获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件。

[0044] 本方案提供的PLC工程的知识表示方法，需要将待处理的PLC程序进行统一的知识表示，具体的，包括对PLC程序中的工程文件和模式文件的知识表示。

[0045] 在本步骤中，在待处理的PLC程序文件中遍历查找，以获PLC程序的工程文件和模式文件。

[0046] 在一种具体的实现方式中，模式文件为XML Schema(即XSD)格式，工程文件为XML格式，则在待处理的PLC程序文件中遍历查找，获取XSD和XML格式的文件。

[0047] S102：根据预先确定的提取规则，获取工程文件对应的节点和节点信息以及模式文件对应的节点和节点信息。

[0048] 其中，节点信息包括节点属性和节点层级关系。提取规则为预先制定的规则，用于实现对工程文件或模式文件的节点和节点信息的提取，具体的实现形式可以通过正则表达式进行匹配获取或者通过文档对象模型(Document Object Model,DOM)获取。

[0049] 具体的，获取工程文件对应的节点和节点信息以及模式文件对应的节点和节点信息的方式包括：

[0050] 通过正则表达式在工程文件和模式文件中进行匹配，获取工程文件和模式文件的节点和节点信息。

[0051] 或者，

[0052] 通过文档对象模型(Document Object Model,DOM)获取工程文件和模式文件的节点和节点信息。

[0053] S103：分别将工程文件和模式文件对应的节点和节点信息，映射到本体知识库。

[0054] 将步骤S102中获取的工程文件和模式文件对应的节点和节点信息映射到本体知识库，该本体知识库用于表示和存储待处理的PLC程序的知识，具体的，将模式文件映射至本体知识库中的模式层，将工程文件映射至本体知识库中的个体(也称作实体)。

[0055] 本实施例提供的一种PLC工程的知识表示方法，获取待处理的PLC程序的工程文件和模式文件，并根据预先确定的提取规则，分别获取该工程文件和模式文件对应的节点和节点信息，分别将该工程文件和模式文件对应的节点和节点信息，映射到本体知识库，通过本体知识库可表示和存储待处理的PLC程序的知识，实现了对不同平台PLC程序进行统一的知识表示，为PLC程序在不同硬件平台间的转移和重复使用的便利性提供了可能。

[0056] 在上述实施例的基础上，在一种具体的实现方式中，将工程文件映射到本体知识库ABox层，将工程文件的节点映射到本体知识库的个体，节点属性映射到本体知识库的属性和属性的限定。

[0057] 例如，以下为一个XML格式的PLC工程文件的实例片段：

```
<block localId="10006" typeName="OR">
  <position x="0" y="0" />
  <inputVariables>
    <variable formalParameter="In1">
    <connectionPointIn>
      <connection refLocalId="10002" formalParameter="Out1" />
    </connectionPointIn>
  </variable>
  <variable formalParameter="In2">
    <connectionPointIn>
```

[0058]

```
      <connection refLocalId="10005" formalParameter="Out1" />
    </connectionPointIn>
  </variable>
</inputVariables>
<inOutVariables />
<outputVariables>
  <variable formalParameter="Out1">
    <connectionPointOut />
  </variable>
  </outputVariables>
</block>
```

[0059] 该段代码描述了一个OR功能块，包含两个输入变量In1和In2，及一个输出变量Out1。其中In1连接localId为10002的功能块的Out1变量，In2连接localId为10005的功能块的Out1变量。使用OWL本体语言对其进行描述，对应的代码如下：

```
<owl:NamedIndividual rdf:about="#block_10006">
```

[0060]

```
<rdf:type rdf:resource=" #block"/>
<rdf:type>
```

[0061]

```
            <owl:Restriction>
              <owl:onProperty rdf:resource="has_block_typeName"/>
              <owl:hasValue>OR</owl:hasValue>
            </owl:Restriction>
          </rdf:type>
          <rdf:type>
            <owl:Restriction>
              <owl:onProperty rdf:resource=" #has_localId"/>
              <owl:hasValue rdf:datatype="#integer">10006</owl:hasValue>
            </owl:Restriction>
          </rdf:type>
        </owl:NamedIndividual>
        <owl:NamedIndividual rdf:about=" #position_1006">
          <rdf:type rdf:resource="#block"/>
          <rdf:type rdf:resource="#position"/>
          <rdf:type>
            <owl:Restriction>
              <owl:onProperty rdf:resource="#has_position_x"/>
              <owl:hasValue rdf:datatype="#integer">0</owl:hasValue>
            </owl:Restriction>
        </rdf:type>
        <rdf:type>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#has_position_y"/>
            <owl:hasValue rdf:datatype="#integer">0</owl:hasValue>
          </owl:Restriction>
        </rdf:type>
      </owl:NamedIndividual>
      <owl:NamedIndividual rdf:about="#variable_10006_In1">
        <rdf:type rdf:resource="#block"/>
```

<rdf:type rdf:resource="#inputVariable"/>
<rdf:type rdf:resource="#variable"/>
<rdf:type>
<owl:Restriction>
<owl:onProperty rdf:resource="#has_formalParameter"/>
<owl:hasValue>Out1</owl:hasValue>
</owl:Restriction>
</rdf:type>
<rdf:type>
<owl:Restriction>
<owl:onProperty rdf:resource="#has_refLocalId"/>
<owl:hasValue rdf:datatype="#integer">10002</owl:hasValue>
</owl:Restriction>
</rdf:type>
</owl:NamedIndividual>

[0062]

<owl:NamedIndividual rdf:about="#variable_10006_In2">
<rdf:type rdf:resource="#block"/>
<rdf:type rdf:resource="#inputVariable"/>
<rdf:type rdf:resource="#variable"/>
<rdf:type>
<owl:Restriction>
<owl:onProperty rdf:resource="#has_formalParameter"/>
<owl:hasValue>Out1</owl:hasValue>
</owl:Restriction>
</rdf:type>
<rdf:type>
<owl:Restriction>
<owl:onProperty rdf:resource="#has_refLocalId"/>
<owl:hasValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">10005</owl:hasV

```
alue>
        </owl:Restriction>
      </rdf:type>
    </owl:NamedIndividual>
    <owl:NamedIndividual rdf:about="#variable_10006_Out1">
      <rdf:type rdf:resource="#block"/>
      <rdf:type rdf:resource="#outputVariable"/>
      <rdf:type rdf:resource="#variable"/>
      <rdf:type>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#has_formalParameter"/>
          <owl:hasValue>Out1</owl:hasValue>
        </owl:Restriction>
      </rdf:type>
    </owl:NamedIndividual>
```

[0063]

[0064] 将模式文件映射到本体知识库TBox层，将模式文件的节点映射到本体知识库中的概念，节点属性映射到本体知识库的属性和属性的限定。

[0065] 例如，以下为一个XML Schema文件的实例片段：

```
<xsd:element name="coordinateInfo">
  <xsd:element name="fbdScaling">
    <xsd:complexType>
      <xsd:attribute name="x" type="xsd:decimal" use="required"/>
      <xsd:attribute name="y" type="xsd:decimal" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:element>
```

[0066]

[0067] 该段代码规定了XML Schema文件中节点coordinateInfo具有子节点fbdScaling，子节点fbdScaling具有两个分别名为“x”和“y”的“xsd:decimal”类型的必需属性。使用OWL本体语言对其进行描述，对应的代码如下：

[0068] <owl:Class rdf:about="#coordinateInfo"/>

```
<owl:Class rdf:about="#fbdScaling">
  <rdfs:subClassOf rdf:resource="#coordinateInfo"/ >
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="# has_fbdScaling_x"/>
      <owl:qualifiedCardinality  rdf:datatype="#  nonNegativeInteger">1</
            owl:qualifiedCardinality>
      <owl:onDataRange rdf:resource="#decimal"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#has_fbdScaling_y"/>
      <owl:qualifiedCardinality    rdf:datatype="#nonNegativeInteger">1</
            owl:qualifiedCardinality>
      <owl:onDataRange rdf:resource="#decimal"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

[0069]

[0070] 本方案提供的PLC工程的知识表示方法，分别将工程文件和模式文件对应的节点和节点信息，映射到本体知识库，形成本体模型，具体包括以下实现方式：将工程文件和模式文件对应的节点和节点信息，使用OWL语言进行表示和存储。

[0071] 在一种具体的实现方式中，本实施例包括：通过Manchester OWL语言对工程文件和模式文件对应的节点和节点信息进行描述和保存，再将Manchester OWL语言转换到OWL语言。

[0072] 如下示例为对模式文件对应的节点和节点信息的提取和映射规则：

[0073] 1、对于一个形如<xsd:element name＝“NAME1”>的节点，将其转换为：Class:Name1。

[0074] 2、若节点<xsd:element name＝“NAME2”>是NAME1的子节点，则将该节点转换为：Class:Name2SubClassOf:Name1

[0075] 3、若节点<xsd:attribute name＝“NAME3”type＝“ppx:type”use＝“required”>是节点NAME2的子节点，则将该节点转换为：

[0076] ObjectProperty:has_NAME2_NAME3

[0077] Domain:NAME2

[0078] Range:NAME2_NAME3

[0079] Datatype:ppx:type

[0080] DataProperty:hasType

[0081] Domain:NAME2_NAME3

[0082] Range:ppx:type

[0083] Class:NAME2_NAME 3

[0084] SubClassOf:NAME 2

[0085] SubClassOf:hasType exactly 1type

[0086] Class:NAME 2

[0087] SubClassOf:has_NAME2_NAME3exactly 1NAME2_NAME3

[0088] 如下示例为使用OWL对工程文件对应的节点和节点信息的表示的规则：

[0089] 根据OWL命名规范，规定本体知识库中的ObjectProperty的命名形式为has_DomainName_AttributeName。

[0090] 1、对于包含name属性的节点<element1name＝“NAME4”>，将该节点转换为：

[0091] Individual:element1_NAME4

[0092] Types:element1

[0093] 2、对于包含localId属性的节点<element2localId＝“ID1”>，将该节点转换为：

[0094] Individual:element2_ID1

[0095] Types:element2

[0096] 若节点<element3attrName1＝“VALUE1”>是该节点的子节点，则将其子节点转换为：

[0097] Individual:element3_ID1

[0098] Types:element3,element2

[0099] Facts:has_element3_attrName1VALUE2

[0100] 3、对于包含formalParameter属性且该属性是第一个属性的节点<element4formalParameter＝“PARAM1”>，将该节点转换为：

[0101] Individual:element4_PARAM1

[0102] Types:element4

[0103] 4、若节点<element5attrName2＝“VALUE2”>是element1的子节点，且属性attrName2不是name或formalParameter，则将该节点转换为：

[0104] Individual:element1_element5

[0105] Types:element5,_has element5attrName2value VALUE2

[0106] 5、若节点<element6refLocalId＝“VALUE3”>是element1的孙节点，该节点可选择性包含formalParameter属性，则对individual:element1增加以下内容：

[0107] Types:has_refLocalId value VALUE3

[0108] 本实施例中，通过将所述工程文件和所述模式文件对应的节点和节点信息，使用OWL进行表示和存储，以形成用于表示和存储PLC程序语义的本体知识库。

[0109] 在一种具体的实现方式中，本方案提供的PLC工程的知识表示方法还包括，构建统一的PLC初始模型，用于为不同知识库提供交互的中介。构建过程主要包括以下四个步骤：1.列举PLC工程领域的重要术语和词汇，例如“功能块”和“方法”等；2.确定PLC本体的类及

类之间的层次关系，定义类之间的层次关系可由自顶向下的方法实现，从PLC工程领域最常见的概念开始，随后逐步对概念进行细化。例如，从定义最顶层的“Project”类开始，然后为其创建一些子类，如“FileHeader”、“Types”和“Instances”，接下来，再对子类做进一步地分类，如“Instance”类包含子类“Configuration”和“Resource”，判断类的层次即判断一个类是否为另一个类的父类或子类，可以通过判断一个类的实例是否也必须是其他类的实例来决定；3.定义PLC本体中类的属性，OWL语言包括对象属性和数据属性两种属性，一个类可以包含多个属性，例如“Link”具有对象属性“hasLink”和“hasParameter”，及数据属性“hasLinkComment”；4.定义PLC本体中属性的限定，一个属性可以具有多个限定，如属性取值的类型、容许的取值、取值基数等。“LD”的对象属性“hasRung”具有取值基数“min 1”，即“hasRung min 1Rung”。

[0110] 在一种具体的实现方式中，本方案提供的PLC工程的知识表示方法可用于语义查询与推理。本体知识库包含本体的概念、实体、属性以及关系等，还包含语义规则，规则可使用描述逻辑(Description Logics，DL)语言或语义网规则语言(Semantic Web Rule Language，SWRL)编写。应用规则，一方面可以对本体知识库中的概念进行一致性检测以保证准确性，另一方面可以查询概念之间潜在的关系。具体的，可使用SPARQL查询语言可以对本体知识库进行查询，例如查询上述实施例中的OR功能块所连接到的功能块有哪些：

[0111]
```
SELECT ?x
WHERE
{
  ?s rdf:type block
  ?s :localId 10006
  ?s :has_refLocalId ?p
  ?p :has_block_typeName ?x
}
```

[0112] 查询结果如表1所示：

[0113]

x
“GT”
“GT”

[0114] 表1

[0115] 通过SPARQL可对本体知识库进行查询操作，如子查询、聚合操作等。对本体知识库的查询和推理可以使用本体推理机来实现，例如Pellet、FaCT++等。

[0116] 图2为本发明实施例提供的PLC工程的知识表示装置的结构示意图，如图2所示，该装置10包括：

[0117] 获取模块11：用于获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件；

[0118] 解析模块12：用于根据预先确定的提取规则，获取所述工程文件对应的节点和节

点信息以及所述模式文件对应的节点和节点信息;所述节点信息包括节点属性和节点层级关系;

[0119] 处理模块13:用于分别将所述工程文件和所述模式文件对应的节点和节点信息,映射到本体知识库,所述本体知识库用于表示和存储待处理的PLC程序的知识。

[0120] 本实施例提供的PLC工程的知识表示装置,包括获取模块、解析模块和处理模块,通过获取待处理的PLC程序的工程文件和模式文件,并根据预先确定的提取规则,分别获取该工程文件和模式文件对应的节点和节点信息,分别将该工程文件和模式文件对应的节点和节点信息,映射到本体知识库,通过本体知识库可表示和存储待处理的PLC程序的知识,实现了对不同平台PLC程序进行统一的知识表示,为PLC程序在不同硬件平台间的转移和重复使用的便利性提供了可能。

[0121] 在一种可能的设计中,所述处理模块13具体用于将所述工程文件和所述模式文件对应的节点和节点信息,使用OWL进行表示和存储。

[0122] 在一种可能的设计中,所述处理模块13具体用于:

[0123] 通过Manchester OWL对所述工程文件和所述模式文件对应的节点和节点信息进行描述和保存;

[0124] 再将Manchester OWL转换到OWL。

[0125] 在一种可能的设计中,所述获取模块具体用于:

[0126] 通过正则表达式在所述工程文件和所述模式文件中进行匹配,获取所述工程文件和所述模式文件的节点和节点信息;

[0127] 或者,

[0128] 通过文档对象模型DOM获取所述工程文件和所述模式文件的节点和节点信息。

[0129] 本实施例提供的装置,可用于执行上述任一方法实施例的技术方案,其实现原理和技术效果类似,本实施例此处不再赘述。

[0130] 图3为本发明实施例提供的终端设备的硬件结构示意图。如图3所示,本实施例的终端设备60包括:处理器601以及存储器602;其中

[0131] 存储器602,用于存储计算机执行指令;

[0132] 处理器601,用于执行存储器存储的计算机执行指令,以实现上述实施例中终端设备所执行的各个步骤。具体可以参见前述方法实施例中的相关描述。

[0133] 可选地,存储器602既可以是独立的,也可以跟处理器601集成在一起。

[0134] 当存储器602独立设置时,该终端设备还包括总线603,用于连接所述存储器602和处理器601。

[0135] 本发明实施例还提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上所述的PLC工程的知识表示方法。

[0136] 在本发明所提供的几个实施例中,应该理解到,所揭露的设备和方法,可以通过其它的方式实现。例如,以上所描述的设备实施例仅仅是示意性的,例如,所述模块的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个模块可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或模块的间接耦合或通信连

接,可以是电性,机械或其它的形式。

[0137] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0138] 另外,在本发明各个实施例中的各功能模块可以集成在一个处理单元中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个单元中。上述模块成的单元既可以采用硬件的形式实现,也可以采用硬件加软件功能单元的形式实现。

[0139] 上述以软件功能模块的形式实现的集成的模块,可以存储在一个计算机可读取存储介质中。上述软件功能模块存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或处理器(英文:processor)执行本申请各个实施例所述方法的部分步骤。

[0140] 应理解,上述处理器可以是中央处理单元(英文:Central Processing Unit,简称:CPU),还可以是其他通用处理器、数字信号处理器(英文:Digital Signal Processor,简称:DSP)、专用集成电路(英文:Application Specific Integrated Circuit,简称:ASIC)等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合发明所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0141] 存储器可能包含高速RAM存储器,也可能还包括非易失性存储NVM,例如至少一个磁盘存储器,还可以为U盘、移动硬盘、只读存储器、磁盘或光盘等。

[0142] 总线可以是工业标准体系结构(Industry Standard Architecture,ISA)总线、外部设备互连(Peripheral Component,PCI)总线或扩展工业标准体系结构(Extended Industry Standard Architecture,EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示,本申请附图中的总线并不限定仅有一根总线或一种类型的总线。

[0143] 上述存储介质可以是由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0144] 一种示例性的存储介质耦合至处理器,从而使处理器能够从该存储介质读取信息,且可向该存储介质写入信息。当然,存储介质也可以是处理器的组成部分。处理器和存储介质可以位于专用集成电路(Application Specific Integrated Circuits,简称:ASIC)中。当然,处理器和存储介质也可以作为分立组件存在于电子设备或主控设备中。

[0145] 本领域普通技术人员可以理解:实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时,执行包括上述各方法实施例的步骤;而前述的存储介质包括:ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0146] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进

行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

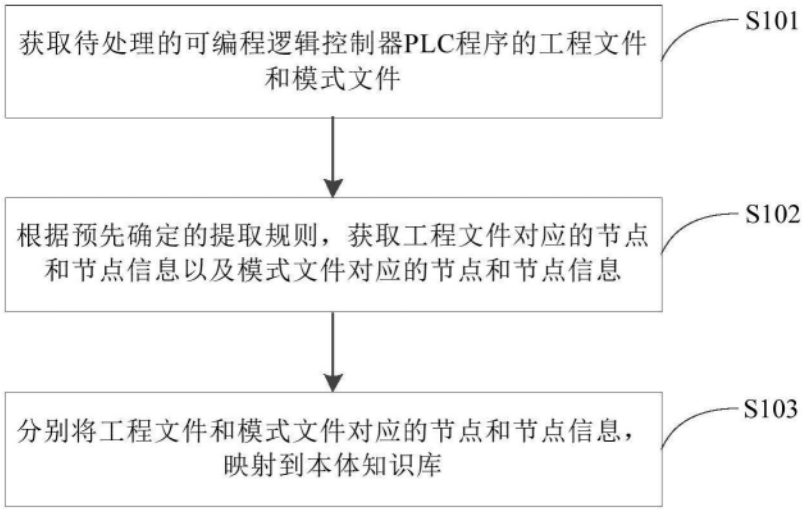

获取待处理的可编程逻辑控制器PLC程序的工程文件和模式文件
S101
根据预先确定的提取规则，获取工程文件对应的节点和节点信息以及模式文件对应的节点和节点信息
S102
分别将工程文件和模式文件对应的节点和节点信息，映射到本体知识库
S103

图1

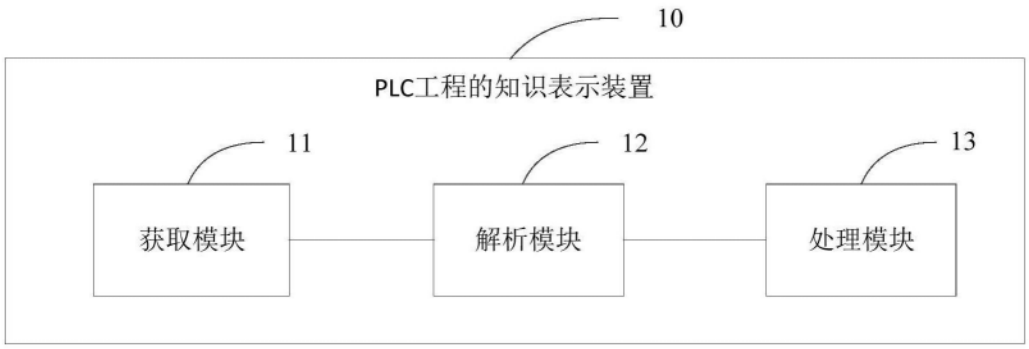

10
PLC工程的知识表示装置
11
获取模块
12
解析模块
13
处理模块

图2

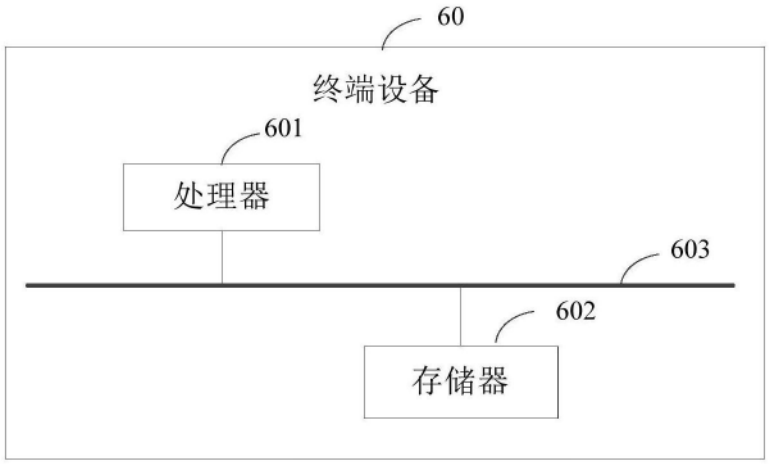

60
终端设备
601
处理器
603
602
存储器

图3