



(12) 发明专利

(10) 授权公告号 CN 113039498 B

(45) 授权公告日 2024. 08. 09

(21) 申请号 201980074908.8

(22) 申请日 2019.11.11

(65) 同一申请的已公布的文献号  
申请公布号 CN 113039498 A

(43) 申请公布日 2021.06.25

(30) 优先权数据  
18206305.7 2018.11.14 EP

(85) PCT国际申请进入国家阶段日  
2021.05.13

(86) PCT国际申请的申请数据  
PCT/EP2019/080902 2019.11.11

(87) PCT国际申请的公布数据  
W02020/099340 EN 2020.05.22

(73) 专利权人 ABB瑞士股份有限公司  
地址 瑞士巴登

(72) 发明人 罗兰·布劳恩  
弗朗西斯科·门多萨

迪尔克·舒尔茨  
海科·科齐奥勒克  
安德里亚斯·伯格

(74) 专利代理机构 北京市金杜律师事务所  
11256  
专利代理师 李辉 范有余

(51) Int.Cl.  
G05B 19/418 (2006.01)

(56) 对比文件  
US 2007250180 A1, 2007.10.25  
US 2017034308 A1, 2017.02.02  
Heiko Koziolk et al..Self-Commissioning Industrial IoT-Systems in Process Automation: A Reference Architecture.《2018 IEEE International Conference on Software Architecture (ICSA)》.2018,第196-205页.

审查员 陈跃燕

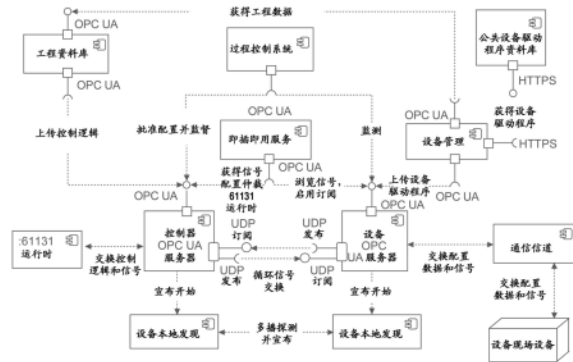
权利要求书2页 说明书10页 附图7页

(54) 发明名称

在工业系统网络中调试现场设备的方法

(57) 摘要

本发明涉及一种在工业系统网络中调试现场设备的方法,方法包括:将现场设备连接到工业系统网络;提供关于现场设备的信息,信息包括用于现场设备的功能信息;利用用于现场设备的功能信息来获得与现场设备在自动化应用中的角色相关的信息;利用与现场设备类型相关的能力信息和与现场设备在自动化应用中的角色相关的信息来获得针对现场设备的参数集,以供在自动化应用中操作;以及将参数集下载到所述现场设备。



1. 一种在工业系统网络中调试现场设备的方法,其中,所述方法由即插即用系统实施,且所述方法包括:
  - a) 将现场设备连接到工业系统网络;
  - c) 由所述现场设备提供关于所述现场设备的信息,所述信息包括用于所述现场设备的功能信息;
  - d) 利用用于所述现场设备的所述功能信息,从与所述现场设备分离的服务器获得与所述现场设备在自动化应用中的角色相关的信息;
  - e) 利用由所述现场设备所提供的关于所述现场设备的所述信息,从与所述现场设备分离的服务器获得与所述现场设备的类型相关的能力信息中的至少一些;
  - f) 利用与所述现场设备的类型相关的能力信息和与所述现场设备在所述自动化应用中的所述角色相关的信息,从与所述现场设备分离的服务器获得针对所述现场设备的参数集,以供在所述自动化应用中操作;以及
  - g) 将所述参数集下载到所述现场设备。
2. 根据权利要求1所述的方法,其中用于所述现场设备的所述功能信息包括所述现场设备的身份。
3. 根据权利要求2所述的方法,其中所述现场设备的所述身份从所述工业系统网络被推断出。
4. 根据权利要求2至3中任一项所述的方法,其中所述现场设备的所述身份根据以下一项或多项被确定:所述现场设备在所述工业系统网络中的位置;与所述现场设备邻近的一个或多个设备的身份;人工输入。
5. 根据权利要求1至3中任一项所述的方法,其中用于所述现场设备的所述功能信息包括所述现场设备的至少一种能力。
6. 根据权利要求1至3中任一项所述的方法,包括步骤b),向所述现场设备提供地址。
7. 根据权利要求6所述的方法,其中所述地址是网络地址。
8. 根据权利要求6所述的方法,其中步骤b)包括收集针对所述现场设备以及针对所述工业系统网络的至少一部分的元数据,以生成随后被提供给所述现场设备的所述地址。
9. 根据权利要求1至3中任一项所述的方法,其中在步骤c)中,由所述现场设备所提供的关于所述现场设备的所述信息包括所述能力信息中的至少一些能力信息。
10. 根据权利要求1所述的方法,其中所述参数集包括与相同设备类型的不同设备相关的数据。
11. 根据权利要求10所述的方法,其中所述方法包括:将所述参数集映射到所述现场设备。
12. 根据权利要求11所述的方法,其中所述映射包括转换所述参数集的所述数据中的至少一些数据的过程。
13. 根据权利要求1至3中任一项所述的方法,包括步骤h),实施浏览过程以将控制器与所述现场设备匹配,所述匹配包括针对所述现场设备的与控制相关的功能性的利用。
14. 根据权利要求13所述的方法,其中与控制相关的所述功能性来自在所述现场设备中所实施的信息模型。
15. 根据权利要求13所述的方法,其中针对所述现场设备的与控制相关的所述功能性

被包括在由所述设备所提供的所述功能信息内。

## 在工业系统网络中调试现场设备的方法

[0001] 相关申请的交叉引用

[0002] 本申请要求于2018年11月14日提交的欧洲专利申请第18 206 305.7号的申请日的权益,该案的全部内容以引用的方式并入本文。

### 技术领域

[0003] 本发明涉及一种在工业系统网络中调试现场设备的方法。

### 背景技术

[0004] 工业系统网络中现场设备的调试需要人工干预,这会导致误差,并花费时间和精力。

### 发明内容

[0005] 因此,具有减少或去除用于在工业系统网络中调试现场设备的人工干预的装置将是有利的。本发明的目的通过独立权利要求的主题被解决,其中其他实施例被并入从属权利要求中。

[0006] 在一方面,提供了一种在工业系统网络中调试现场设备的方法。

[0007] 方法包括:

[0008] a) 将现场设备连接到工业系统网络;

[0009] c) 提供关于现场设备的信息,该信息包括用于现场设备的功能信息;

[0010] d) 利用用于现场设备的功能信息来获得与现场设备在自动化应用中的角色相关的信息;

[0011] f) 利用与现场设备类型相关的能力信息(capability information)和与现场设备在自动化应用中的角色相关的信息来获得针对现场设备的参数集,以供在自动化应用中操作;以及

[0012] g) 将参数集下载到现场设备。

[0013] 在示例中,用于现场设备的功能信息包括现场设备的身份。

[0014] 在示例中,现场设备的身份从工业系统网络被推断出。

[0015] 在示例中,现场设备的身份根据以下一项或多项被确定:现场设备在工业系统网络中的位置;与现场设备邻近的一个或多个设备的身份;人工输入。

[0016] 在示例中,用于现场设备的功能信息包括现场设备的至少一种能力。

[0017] 在示例中,步骤c) 包括由现场设备提供关于该设备的信息。

[0018] 在示例中,方法包括步骤b),向现场设备提供地址。

[0019] 在示例中,地址是网络地址。

[0020] 在示例中,步骤b) 包括收集针对现场设备以及针对工业系统网络的至少一部分的元数据,以便生成随后被提供给现场设备的地址。

[0021] 以该方式,DNS名称可以被生成并被提供给现场设备,该DNS名称将与应用相关的

语义并入现场设备的站名中。因此,使用现场设备(DNS)名称帮助操作者更有效且更轻松地获得/标识设备。

[0022] 在示例中,方法包括步骤e),利用由现场设备所提供的关于该现场设备的信息来获得能力信息中的至少一些。

[0023] 在示例中,能力信息中的至少一些从与现场设备分离的服务器被获得。

[0024] 在示例中,在步骤c)中,由现场设备所提供的关于该现场设备的信息包括能力信息中的至少一些。

[0025] 在示例中,在步骤d)中,与现场设备的角色相关的信息从与现场设备分离的服务器被获得。

[0026] 在示例中,在步骤f)中,参数集从与现场设备分离的服务器被获得。

[0027] 在示例中,参数集包括与相同设备类型的不同设备相关的数据。

[0028] 在示例中,方法包括将参数集映射到现场设备。

[0029] 在示例中,映射包括转换参数集的数据中的至少一些的过程。

[0030] 以该方式,参数集可以用于相同设备系列的较旧设备,并且该较旧参数集可以被映射到较新的现场设备,在必要时涉及一些数据转换。以该方式,设备替换被简化。

[0031] 在示例中,方法包括步骤h),实施浏览过程以将控制器与现场设备匹配,该匹配包括针对现场设备的与控制相关的功能性的利用。

[0032] 在示例中,与控制相关的功能性来自在现场设备中所实施的信息模型。

[0033] 在示例中,针对现场设备的与控制相关的功能性被包括在由设备所提供的功能信息内。

## 附图说明

[0034] 示例性实施例将在下文参考以下图式被描述:

[0035] 图1示出了实施在工业系统网络中调试现场设备的方法的示例的架构的详细静态视图;

[0036] 图2示出了在工业系统网络中调试现场设备的方法的示例内的设备发现和参数化的详细示例(UML交互图);

[0037] 图3示出了在工业系统网络中调试现场设备的方法的示例内的信号匹配的详细示例(UML交互图);

[0038] 图4至5示出了在工业系统网络中调试现场设备的方法的示例内的组件与硬件节点的部署的详细示例;

[0039] 图6描绘了设备OPC UA服务器的OPC UA地址空间的摘录;以及

[0040] 图7示出了HMI服务的示例性控制流程图。

## 具体实施方式

[0041] 图1至7涉及一种在工业系统网络中调试现场设备的方法。在工业系统网络中调试现场设备的方法包括:

[0042] a) 将现场设备连接到工业系统网络;

[0043] c) 提供关于现场设备的信息,该信息包括用于现场设备的功能信息;

- [0044] d) 利用用于现场设备的功能信息来获得与现场设备在自动化应用中的角色相关的信息;
- [0045] f) 利用与现场设备类型相关的能力信息和与现场设备在自动化应用中的角色相关的信息来获得针对现场设备的参数集,以供在自动化应用中操作;以及
- [0046] g) 将参数集下载到现场设备。
- [0047] 根据示例,用于现场设备的功能信息包括现场设备的身份。
- [0048] 根据示例,现场设备的身份从工业系统网络被推断出。
- [0049] 根据示例,现场设备的身份根据以下一项或多项被确定:现场设备在工业系统网络中的位置;与现场设备邻近的一个或多个设备的身份;人工输入。
- [0050] 根据示例,用于现场设备的功能信息包括现场设备的至少一种能力。
- [0051] 根据示例,步骤c) 包括由现场设备提供关于该设备的信息。
- [0052] 根据示例,方法包括步骤b),向现场设备提供地址。
- [0053] 根据示例,地址是网络地址。
- [0054] 根据示例,步骤b) 包括收集针对现场设备以及针对工业系统网络的至少一部分的元数据,以生成随后被提供给现场设备的地址。
- [0055] 根据示例,方法包括步骤e),利用由现场设备所提供的关于该现场设备的信息来获得能力信息中的至少一些。
- [0056] 根据示例,能力信息中的至少一些从与现场设备分离的服务器被获得。
- [0057] 根据示例,在步骤c) 中,由现场设备所提供的关于该现场设备的信息包括能力信息中的至少一些。
- [0058] 根据示例,在步骤d) 中,与现场设备的角色相关的信息从与现场设备分离的服务器被获得。
- [0059] 根据示例,在步骤f) 中,参数集从与现场设备分离的服务器被获得。
- [0060] 根据示例,参数集包括与相同设备类型的不同设备相关的数据。
- [0061] 根据示例,方法包括将参数集映射到现场设备。
- [0062] 根据示例,映射包括转换参数集的数据中的至少一些的过程。
- [0063] 根据示例,方法包括步骤h),实施浏览过程以将控制器与现场设备匹配,该匹配包括针对现场设备的与控制相关的功能性的利用。
- [0064] 根据示例,与控制相关的功能性来自在现场设备中所实施的信息模型。
- [0065] 根据示例,针对现场设备的与控制相关的功能性被包括在由设备所提供的功能信息内。
- [0066] 利用附图继续,实施在工业系统网络中调试现场设备的方法的即插即用系统被示出并且现在被详细描述。
- [0067] 图1描绘了参考架构的静态视图,该架构侧重于ANSI/ISA-95 2级和1级(level)。控制器和现场设备经由OPC UA通过以太网进行通信,以进行调试和操作。现有的现场总线和模拟连接可以通过具有在不同网络之间进行仲裁的IO网关节点而被包括。使用OPC UA服务器,控制器和现场设备成为IIoT设备,这些设备在需要时可以被工厂所有者从因特网访问,并且能够自我描述。
- [0068] 具体来说,图1示出了存储控制逻辑程序和控制器状态信息的控制器OPC UA服务

器。还存在为连接的现场设备提供数据和服务的设备OPC UA服务器。虽然经典的客户端/服务器连接管理控制器和现场设备的自调试,但是用于闭环控制的循环的实时通信被OPC UA发布/订阅连接(IEC 62541-14)支持。

[0069] 使用TSN启用的网络交换机,确定性的实时控制环成为可能。发布/订阅通信可以是基于代理的(例如经由AMQP或MQTT)或是无代理的。在后一种情况下,发布者经由订阅者侦听的UDP将数据集发送到多播IP地址。使用较低级别的网络协议IGMP或MMRP,网络业务可以被减少,以使得网络交换机仅转发订阅者已被注册的网络分组。

[0070] 为了发现连接的OPC UA服务器,参考架构建议根据IEC62541-12的OPC UA控制器本地发现服务器和设备本地发现服务器。资源受限的设备可以备选地使用多播DNS(mDNS)代替本地发现服务器,例如,这也被Apple Bonjour或Spotify Connect实施。mDNS可以使用DNS SRV记录将针对OPC UA服务器的发现URL通告给客户端。

[0071] 参考架构中的控制器提供了可以执行PLCopen控制逻辑程序(R5)的IEC 61131运行时。它可以经由控制器OPC UA服务器从工程资料库中导入与PLCopen兼容的控制逻辑。控制逻辑揭露了需要例如来自传感器(诸如级(level)指示器)的连接的输入信号的功能块。随后,控制算法计算例如用于致动器(诸如用于热交换器的阀)的输出值。控制算法循环运行,例如每100ms运行一次,并且针对特殊应用甚至每1ms运行一次。61131运行时与实际的现场设备解耦,并且可以在不同的部署目标上运行。

[0072] OPC UA服务器揭露了在参考架构中关于现场设备的信息。如果这种服务器具有适当的CPU能力和存储器,那么该服务器可以在专用网络节点上或直接在现场设备上运行。在专用节点(诸如IO网关)上,服务器可以管理针对大量现场设备的数据和服务。现场设备经由通信信道被连接到OPC UA服务器。这可能会封装现场总线或模拟通信,在这种情况下,其需要被适当地配置。

[0073] 设备管理负责为通信信道提供驱动程序,并为现场设备提供参数化。它从公共设备驱动程序资料库(例如从特定供应者或其他组织)获取驱动程序。例如,Fieldcom Group处于为FDI设备包设置这种服务器的过程中。参数化可以依赖于导入的工程数据,并且如果需要微调,该参数化还可以经由来自OPC UA客户端的用户输入被更改。然而,为了使设备具有基本特征,导入的工程数据就足够了。设备管理将参数化上传到现场设备。

[0074] 最后,架构包含查找连接的控制器并处置与现场设备匹配的信号的即插即用服务。该即插即用服务具有内置的OPC UA客户端,以从控制器获得所需的信号名称,并针对匹配信号名称浏览用于现场设备的OPC UA服务器。其知道不同的语义标准,诸如设备的OPC UA、IEC 61987或NAMUR NE131,这有助于执行信号匹配。例如,其可以被部署在工业PC或控制器上。

[0075] 设备参数化遵循图2中的程序。首先,在现场设备经由附加网络连接被连接的情况下,设备OPC UA服务器可选地扫描其连接的通信信道。否则,该步骤可以被省略。随后,该设备OPC UA服务器在网络上宣布自己,其被设备管理组件获得。

[0076] 设备管理获得连接的现场设备的设备ID,这些ID可以由制造者名称、产品类型ID和序列号构成(请参阅IEC 62541-100)。

[0077] 如果设备OPC UA服务器未提供ID,或如果所提供的ID对于设备管理来说是未知的或格式未知,那么它可以尝试从上下文信息推断出设备身份。设备管理可以例如分析设备

的网络地址并确定同一子网中的周围设备。如果这些设备对于设备管理是已知的,并且工程资料库包括在网络上尚未被找到的同一区域中的设备的规范,那么它可以预测哪个设备已被连接。设备还可以携带其他信息,例如地理位置、设备类型特性的已知性质或对其他设备的引用,这可以帮助推断其身份。如果设备管理无法使用所描述的方法中的任何一个确定设备身份,那么它可以询问人类用户以进行输入,并且因此获得设备ID。

[0078] 该设备管理使用ID来针对设备驱动程序查询公共设备驱动程序资料库和工程资料库参数化数据。如果设备没有有意义的主机名,那么设备管理可以通过并入来自设备的OPC UA服务器的信息来生成这种名称。例如在设备已被较新版本替换的情况下,工程资料库可能包含较旧类型的设备的参数化数据。在该情况下,工程资料库可以提供一种机制,如果设备属于同一系列或至少具有明显的功能性重叠,那么该机制允许将参数从一种设备类型映射到另一种设备类型。映射可以涉及更改参数名称,改变参数类型(例如将32位编码转换为64位编码),和甚至改变参数值(例如将预定义的参数值范围转换为另一范围)。

[0079] 其将信息上传到OPC UA服务器,该服务器将其转发到实际的现场设备。最后,OPC UA服务器更新其内容,并向设备管理发出准备就绪的信号。在该更新过程之后,现场设备可以容易地被参数化。

[0080] 现在,信号匹配可以开始,如图3中所示出。即插即用服务不断搜索被连接到网络的控制器OPC UA服务器。该即插即用服务连接到这种服务器,并获得执行配置的控制逻辑程序所需的输入和输出信号的列表。随后,服务将连接到在工厂调试期间已经被连接或正在被连接的任何设备OPC UA服务器。

[0081] 即插即用服务浏览OPC UA服务器的地址空间,并尝试为已配置的设备查找匹配的标签名称。随后,该即插即用服务搜索带有指定输入信号(主要用于致动器)和输出信号(主要用于传感器)的PLCopen功能块。一旦找到,该即插即用服务尝试查找与来自控制器的信号名称的匹配。在控制器和设备使用不同的信号名称(由于它们已经被独立配置)的情况下,匹配也可能依赖于其他信息,诸如OPC UA地址空间中的物理位置和网络配置。

[0082] 在匹配的情况下,即插即用服务设置控制器对输出信号的订阅。这可以是随后需要在设备OPC UA服务器上处置客户端会话的OPC UA客户端/服务器连接,或其可以是经由多播UDP/IP或代理的OPC UA发布/订阅连接。针对快速循环控制,控制器OPC UA服务器可以绕过将信号值写入其地址空间中,而直接将其转发到61131运行时以运算控制逻辑输出。

[0083] 在设置订阅之后,即插即用服务与设备OPC UA服务器断开连接,并在控制器OPC UA服务器中设置旗标,以针对最终批准询问人类工程师。已找到的配置可以被中继给工程师,该工程师可以在实际开始过程之前执行最终检查。

[0084] 组件与硬件节点的部署是灵活的,并允许解决不同的应用上下文和资源约束。图4和5示出了两个不同的变体,其中图4示出了经典设置(UML部署图),并且其中图5示出了具有智能现场设备的设置(UML部署图)。

[0085] 图4中的经典部署以控制器设备、具有连接的现场设备的I/O网关和设备管理服务为特征。后者还托管即插即用服务,该服务随后可以为被连接到以太网网络链路的不同控制器和I/O网关服务。该部署的益处在于投资保护。该部署不需要IIoT启用的现场设备,这是因为I/O网关可以管理所需的IIoT通信。带有现场总线或模拟连接的现有安装可以通过对硬件进行少量添加被更新。



[0086] 具有智能IIoT现场设备的部署(图5)不包括I0网关设备。现场设备具有足够的计算能力和存储器以及网络接口,以托管其自己的OPC UA服务器并将其揭露在以太网网络链路上。作为部署变体,控制器设备托管即插即用服务,并且因此成为具有在给定的网络访问的情况下对其自己进行配置的能力的独立自主代理。

[0087] 可以想到许多其他形式的部署,例如,将61131运行时托管在功能强大的致动器、工厂侧服务器或云平台上。这些部署取决于特定的应用上下文、业务约束和可用的硬件。

[0088] 图6描绘了设备OPC UA服务器的OPC UA地址空间的摘录。地址空间遵循针对设备的OPC UA规范,并在其对象树中包含一个设备集(DeviceSet)。所包括的IoT设备(IoTDevice)是根据PLCopen以及设备本身的特殊CtrlConfiguration。因此,它揭露了重要的标识信息,诸如序列号、制造者和所需的通信协议。

[0089] 如该参考架构所建议,IoTDevice包含特殊功能块,该功能块揭露了其输入和输出变量,其通常仅针对控制器被规定。设备上的该功能块独立于其上运行的任何应用或控制功能。即插即用服务使用该信息来匹配来自控制器的信号名称。

[0090] 以下内容提供了关于在工业系统网络中调试现场设备的方法的细节的其他信息。

[0091] PnP服务规范

[0092] 以下部分指定了来自图1的即插即用服务(PnPService)的不同服务,这些服务被用于满足即插即用功能性。服务以本文档的读者应该能够实施它们为重点被描述。针对每种服务,详细说明被给出以及服务的输入和输出。除一些核心服务外,还提供了所产生的实施方式的伪代码抽象。

[0093] 寻找服务器(FindServers)

[0094] findServers功能性允许标识网络中可用的OPC UA服务器,并使它们可访问以在PnPService中进行进一步处理。

[0095] findServers服务返回服务器或发现服务器已知的服务器。发现服务器的详细行为例如在L.Dürkop等人发表在2013年7月第11届IEEE国际会议的工业信息学(INDIN)中的第248至253页的Using OPC-UA for the Autoconfiguration of real-time Ethernet Systems(将OPC-UA用于实时以太网系统的自动配置)的论文中被详细描述。附加地,例如,如果设备内存在资源约束,那么相同的行为可以通过使用多播DNS(mDNS)协议被实施。PnPService使用发现方法来标识网络内所有可用的OPC UA服务器,以便支持自我调试步骤。因此,PnPService还过滤已利用的OPC UA服务器,以便避免设备的重复处理。

[0096] 该服务的mDNS版本不需要特定的参数。

[0097] 浏览

[0098] PnP服务提供了浏览功能性,以扫描OPC UA信息模型并检查可以被彼此匹配的信号。

[0099] 浏览服务分析发现的OPC UA服务器的信息模型。一旦findServers服务标识了OPC UA端点,浏览服务就开始连接到该服务器并浏览其信息模型。从而,服务开始浏览类型,这些类型被PLCopen[3]类型继承。一旦这些类型被标识出,它们就被服务存储,以用于浏览过程的第二阶段。在该第二阶段中,浏览服务查找收集PLCopen类型的实例。在两阶段浏览过程期间,服务仅通过使用深度优先搜索算法(DFS)来遵循信息模型的层次结构引用。在每个节点处,浏览服务调用匹配服务以查找控制逻辑的标识出的信号与浏览的设备的潜在输入

信号和输出信号之间的可能匹配。通过使用该匹配服务,浏览服务也可以被用于非PLCopen一致性信息模型。以下伪代码片段使浏览服务的功能性可视化。该伪代码片段还包括针对每个节点的匹配服务调用。

[0100] 列表1浏览服务的实施方式

```
int PnPService::browse(NodeId node_id, list<MatchingRule*> m_rules, int browseDir)
```

```

for MatchingRule m_rule in m_rules do
    if m_rule->match(node_id, pMyClient) then
        //tuple of node id and browse name of matched node
        t = tuple(m_nodeid, m_browseName);
        store tuple in result list
    end if
end for

```

[0101]

```
//expand method of DFS; follow hierarchical references and push them on stack
expand(node_id, browseDir)
```

```

while ! stack.empty() do
    NodeId new_node_id = stack.pop();
    //call browse service recursively
    browse(new_node_id, m_rules, browseDirection);
end while
//once stack is empty return
return 0;

```

[0102] 参数

[0103] 输入:

[0104] -当前节点的ID

[0105] -在浏览过程期间应被应用的匹配规则列表

[0106] -指示浏览方向的整数,遵循OPC UA规范以供参考

[0107] 输出:

[0108] -正匹配的节点被存储在全局列表中

[0109] 匹配

[0110] 只要两个信号满足所谓的匹配规则对象中所描述的匹配准则,该服务便将它们彼此匹配。

[0111] 匹配服务被用于标识设备的哪个输入信号适合控制逻辑的输出信号,且反之亦然。服务本身只能与浏览服务一起被使用。它不能被自身调用。因此,所谓的匹配规则对象需要被实施并被添加到浏览服务。这些匹配规则对象在每个节点处检查输入/输出信号是否可以适合控制逻辑的信号。如果这些输入/输出信号适合,那么由于匹配规则对象的规范,PnPService会处理它,并订阅必要的多播地址以开始在特定多播组内发布信号。

[0112] 参数

[0113] 输入:

[0114] -当前节点的ID

[0115] -指向在必要时从当前节点获取更多信息的OPC UA客户端实例的指针

[0116] 输出:

[0117] -布尔值,指示正或负匹配结果

[0118] 匹配规则 (MatchingRule)

[0119] 匹配规则是一个对象,该对象继承自基础类别并提供了将两个给定信号作为 NodeId 彼此匹配的功能性。

[0120] 匹配规则概念允许使用不同的业务逻辑容易地扩展 PnPService,以供匹配控制器和设备的信号。由于该情况,这不是 PnPService 的明确服务,而是使 PnPService 灵活且易于扩展的核心功能性。以下伪代码示出了基础类别的结构及其功能性,该功能性需要被派生对象实施。

[0121] 列表 2 MatchingRule 基础类别的实施方式

```

class MatchingRule
{
public:
    MatchingRule();
    virtual ~MatchingRule();
    // core matching functions
    virtual bool match(NodeId &ua_node_id, OPCUAClient* opc_uaclient);

[0122]     // nodeId and browse name of matched signal; only to buffer the data after a match;
    // will be overwritten by the next match;
    NodeId m_nodeid;
    std::string m_browseName = "";
private:
    // possible identification member variables of the certain matching rule
    int mid;
    std::string name ;
};

```

[0123] 这些匹配对象在匹配服务中被用于标识哪些节点/信号应被彼此匹配。匹配指示取决于信号的类型(输入或输出信号)来订阅或发布在节点内所存储的该值。

[0124] 增加节点 (AddNodes)

[0125] addNodes 服务被设计用于将适当信号结构添加到控制器,从而存储必要的信息以供匹配输入信号和输出信号。如果控制器支持 PLCopen,那么该结构是不必要的。

[0126] addNodes 服务使得能够在运行时期间将节点添加到信息模型。PnPService 使用该功能性来支持不符合 PLCopen 的控制器或设备。这些设备通过将特定的节点结构添加到信息模型、存储关于其输入信号和输出信号、数据类型和端点信息的详细信息被支持。之后, PnPService 使用该数据来实现两个信号之间的匹配。

[0127] 参数

[0128] 输入:

[0129] -添加到信息模型的节点列表

[0130] -端点 url

[0131] 输出:

[0132] -状态旗标

[0133] 运行

[0134] 本小节描述了 PnPService 的运行时功能性。

[0135] 在启动时, PnP 服务会更新其匹配表。这意味着 PnP 服务从控制器获得所有输入信

号和输出信号,以为匹配服务提供输入。在运行时期间,PnP服务本身以不同的状态被组织。这些状态中的大多数是在内部设置的,例如以检查以前未被探索过的新设备是否可用(state\_new\_device\_is\_available)。但是,其中一些是在外部设置的,尤其根据PnP服务的其他功能性(如替换功能性)被设置(state\_matching\_mode)。在PnP服务处于匹配模式时,它等待以前未被浏览过的新设备。

[0136] 其发现服务器侦听器之间的通信经由消息队列被完成。一旦新的端点被接收到,PnP服务就更新信号匹配表,该信号匹配表包含来自控制器的信号以及从设备到控制器的已匹配信号。如果并非所有信号都被匹配,那么PnP服务调用浏览服务。如果在这种浏览过程之后,所有信号都被匹配,那么即使有新设备可用,PnP服务也将不再尝试匹配信号。下文的伪代码将大致了解PnP服务的过程流程。

[0137] 列表3PnP服务运行方法的实施方式

```

void *PnPServiceProvider::run(void* arg)
    //update signals matching table
    PnPService update signals from controller

    while true do
        if state_matching_mode then

            while ! state_new device_is_available do
                //blocks till new opc ua endpoint is send via message queue
                size = message queue receive new opc ua endpoint

                if size == -1 then
                    "message queue receive failed";
                else
                    state_new device_is_available = true
                end if

            end while
            //update signals matching table
            PnPService update signals from controller

            if ! state_all_signals_matched then
                state_new device_is_available = false
                PnPService call start browse service

            end if

            state_all_signals_matched = PnPService check if all signals are matched
            state_matching_mode = false

        end if
    end while

```

[0139] HMI

[0140] 图7示出了HMI服务的控制流。在演示程序实施方式中,PnPService提供了简单的命令行接口(CLI)。CLI被用于触发PnPService的不同服务/功能性

[0141] 大体上,HMI服务的控制流被分为两部分:调试和设备替换。调试阶段在启动之后立即开始。在该阶段,PnPService等待发现服务器/功能性所宣布的新设备。因此,用户按下“T”以侦听新设备。一旦出现新设备,PnPService就会浏览该设备并使用匹配服务以供标识可能要匹配的信号。如果信号被匹配并且并非控制器的所有信号都被匹配,那么HMI跳回到侦听新设备的状态。如果所有信号都被匹配,那么PnPService示出匹配表,并要求对该表进

行验证。如果用户按下“ok”，那么PnPService将触发控制器和所有其他设备开始生产。在另一情况下，匹配结果可能有缺陷，以使得手动修复信号匹配是必要的。

[0142] HMI服务的第二部分是设备替换的公告。在生产状态期间，该功能性通过按下“r”被触发。此后，PnPService列出了所有可以被替换的设备。用户需要从列表选择设备。在下一步骤中，受分析影响的将被PnPService完成并被呈现给用户。基于该分析，PnPService要求确认替换该设备并将所有受设备影响的设置成处于仿真模式。现在应该被替换的设备可以被拔出，并且新设备可以被插入。这样做，PnPService跳回侦听、标识新设备、开始浏览该新设备并将新信号与控制器中不匹配的信号进行匹配。再次，匹配表被呈现并且需要被用户验证。

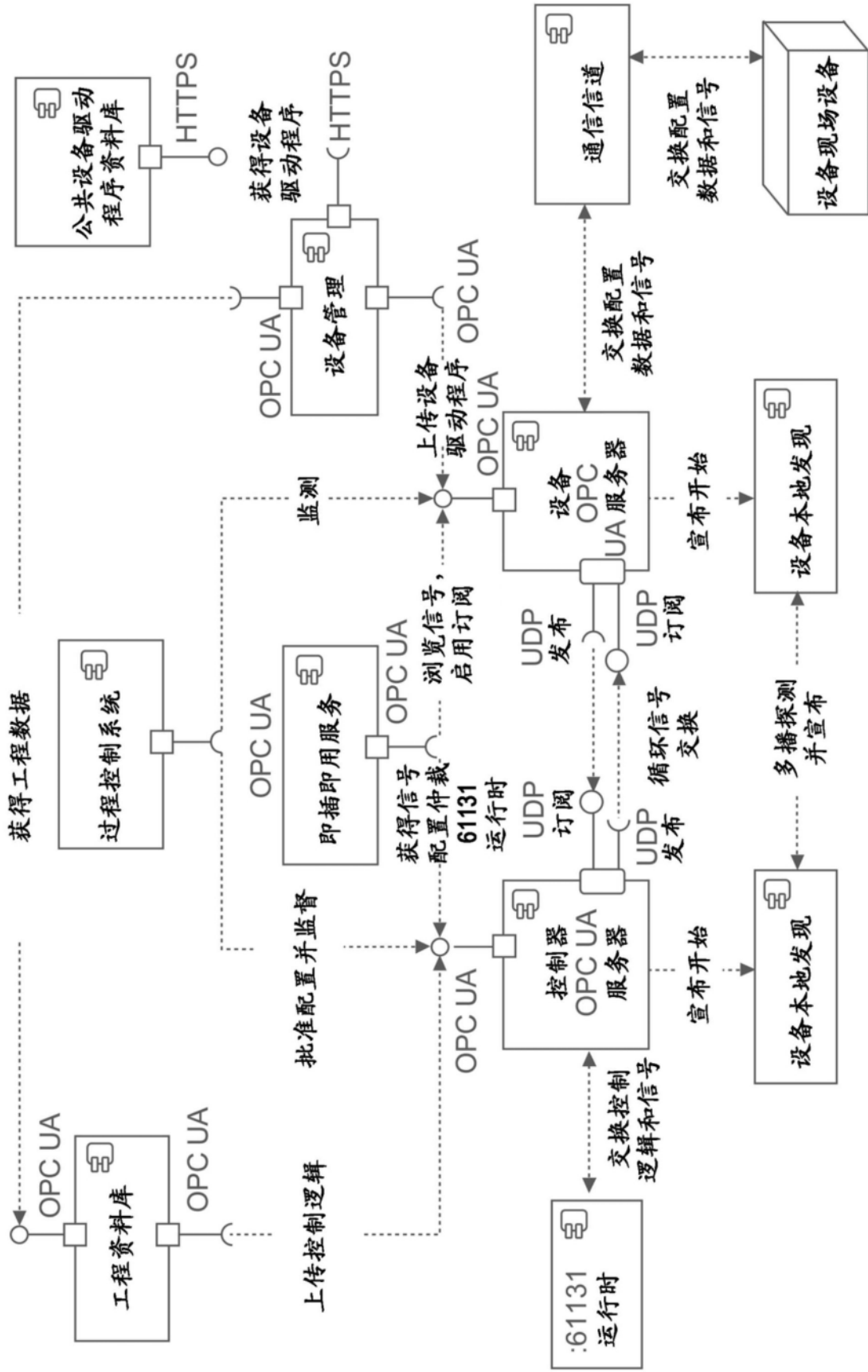


图1

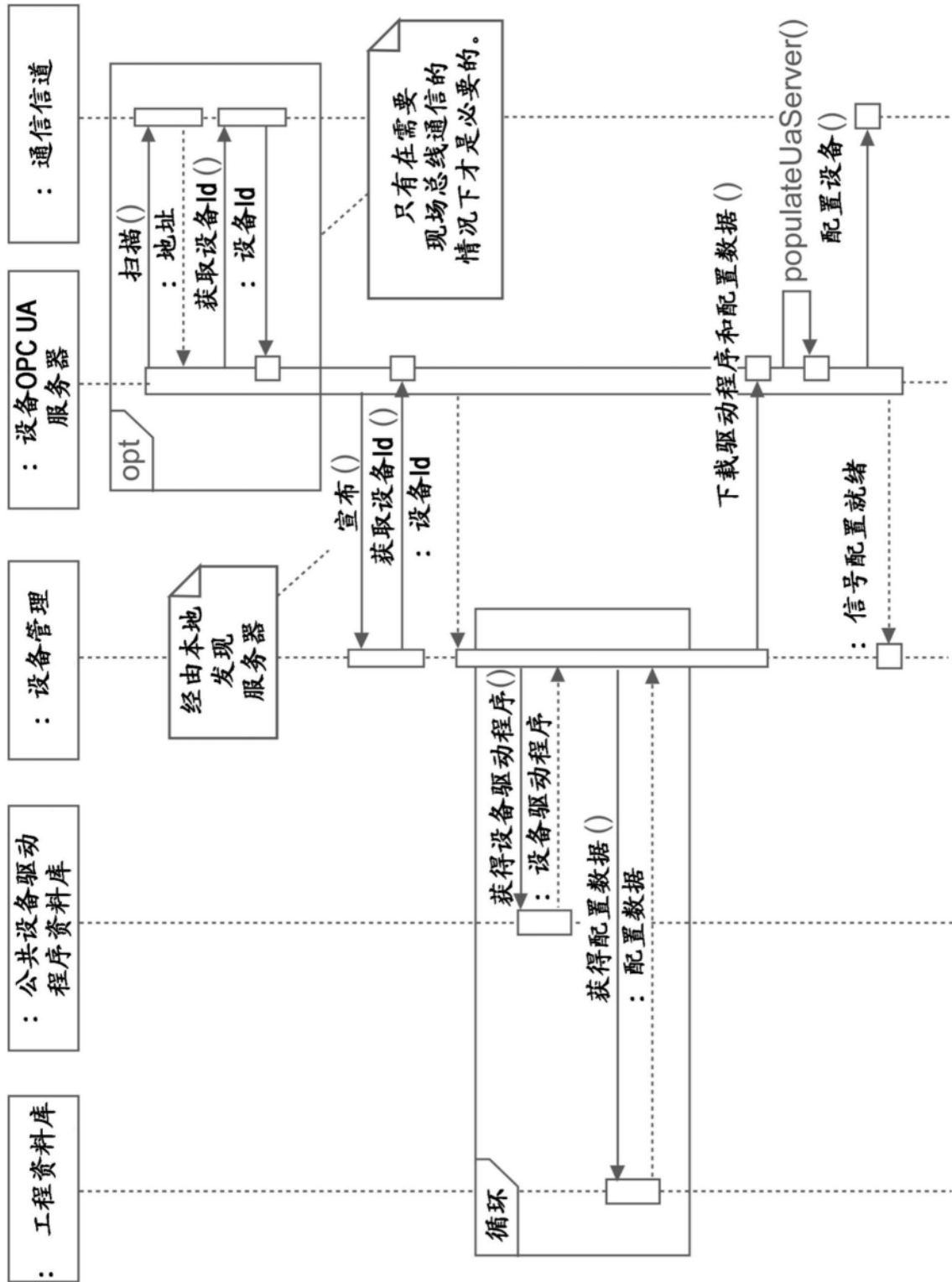


图2

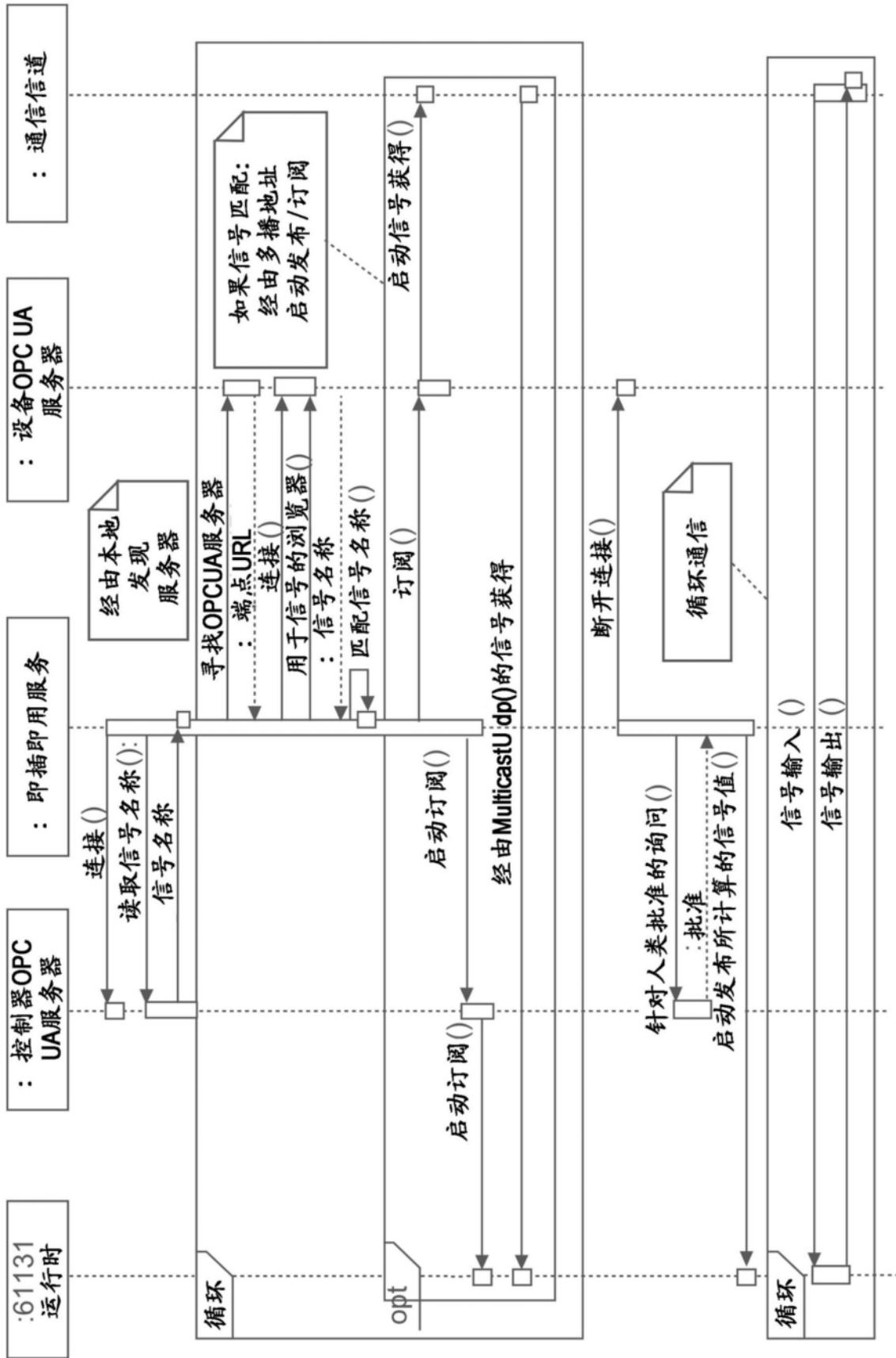


图3



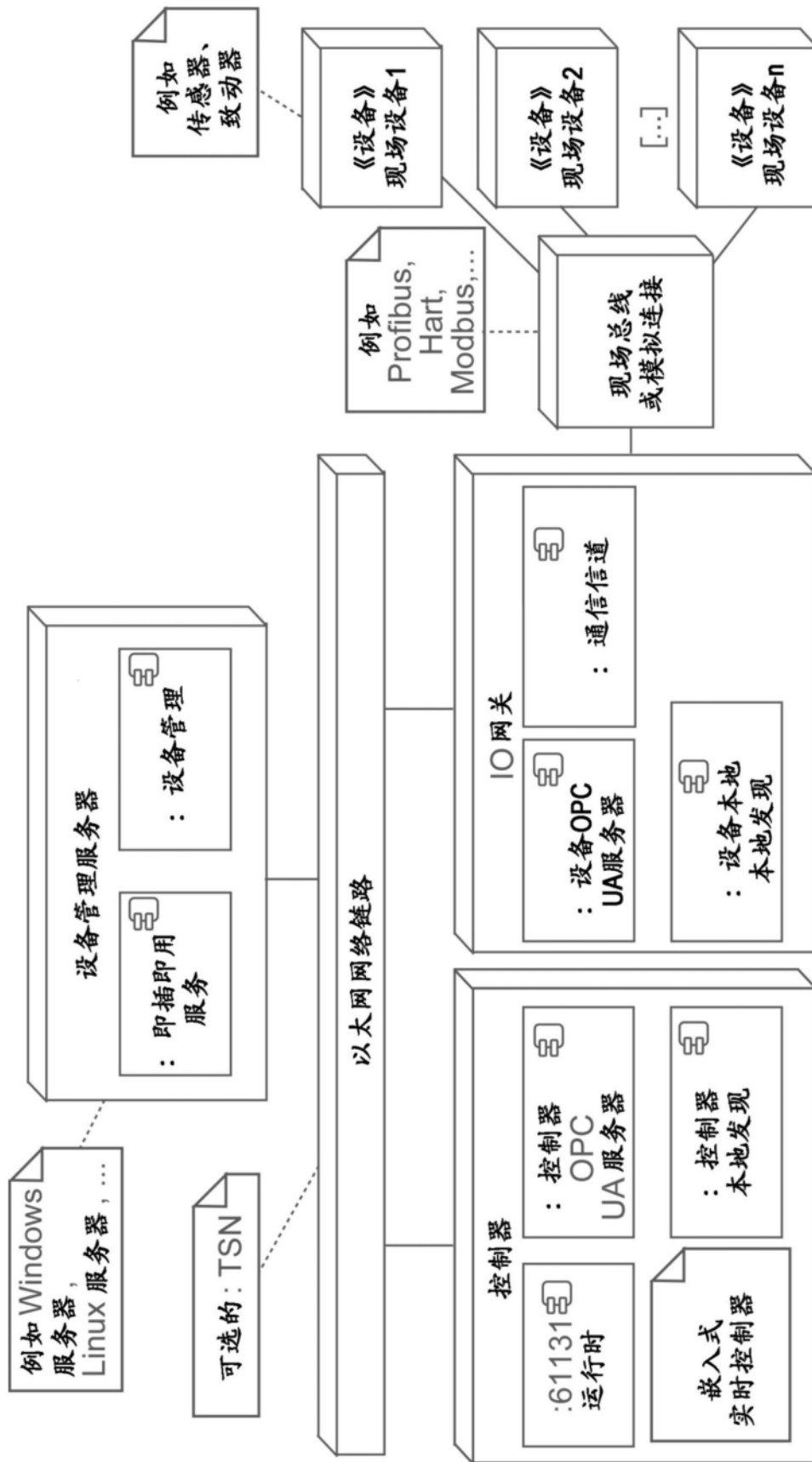


图4

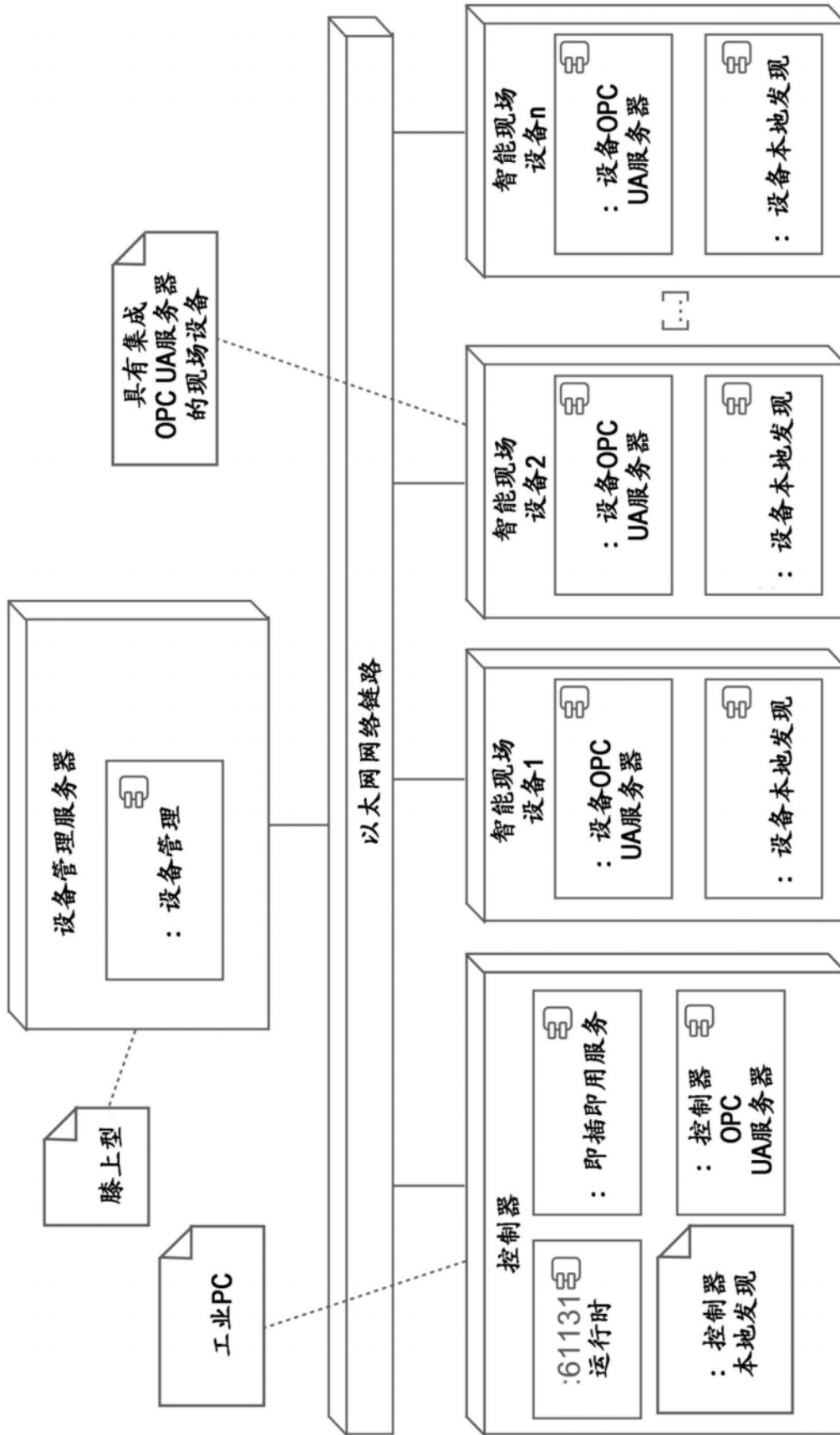


图5

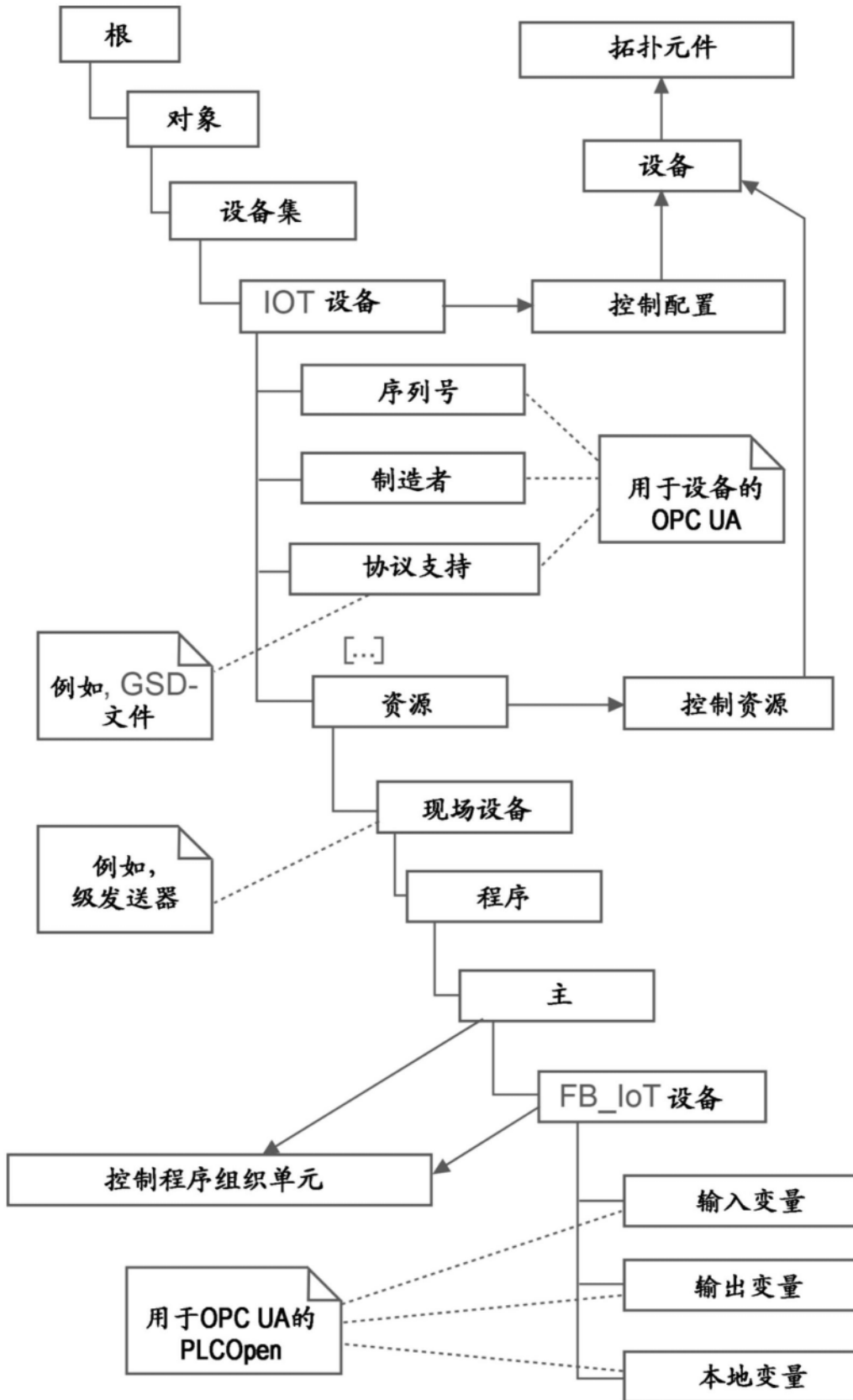


图6

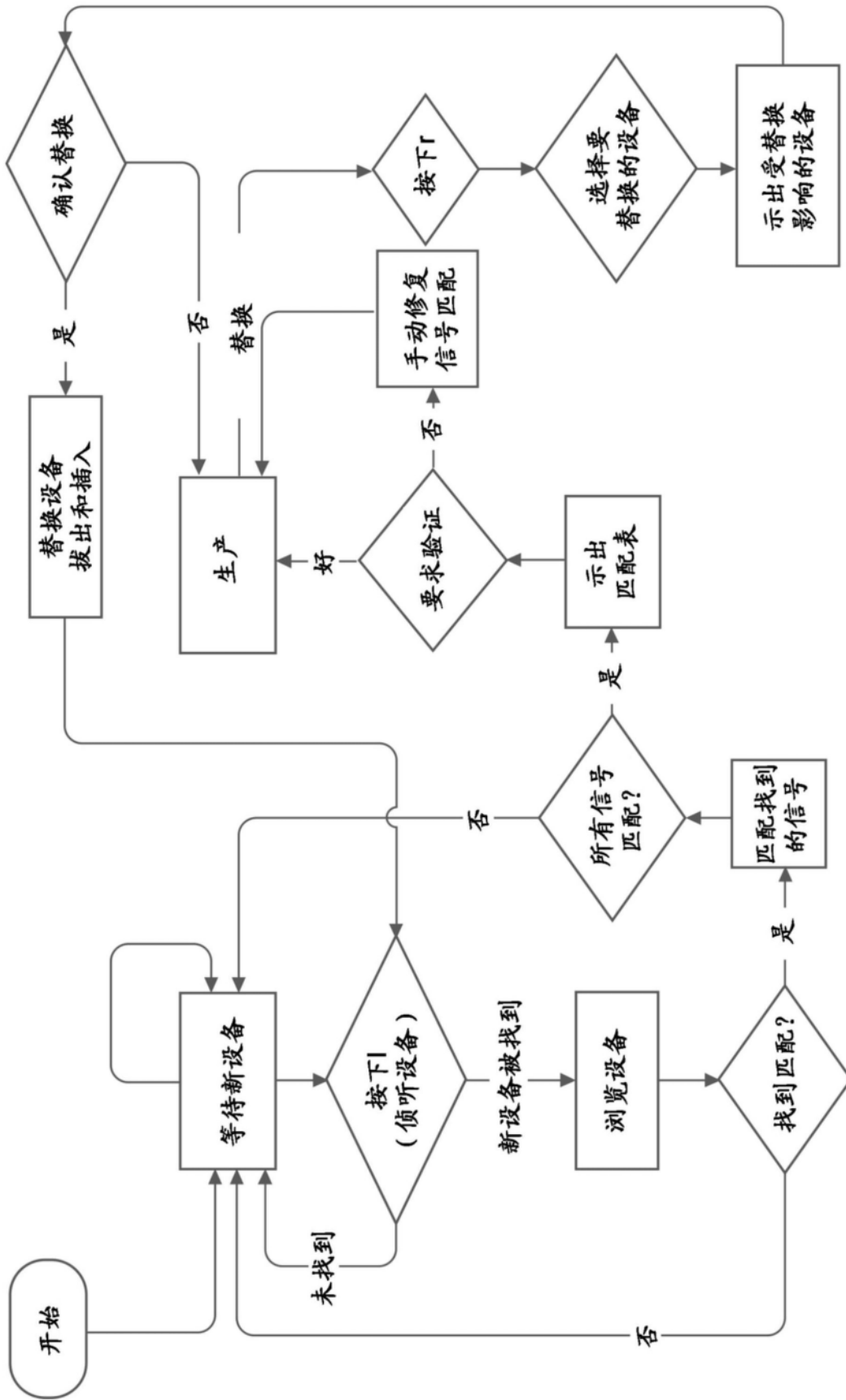


图7