



(12)发明专利申请

(10)申请公布号 CN 109564419 A

(43)申请公布日 2019.04.02

(21)申请号 201680088176.4

(51)Int.Cl.

(22)申请日 2016.08.10

G05B 19/418(2006.01)

(85)PCT国际申请进入国家阶段日
2019.01.31

(86)PCT国际申请的申请数据
PCT/US2016/046226 2016.08.10

(87)PCT国际申请的公布数据
W02018/031005 EN 2018.02.15

(71)申请人 西门子股份公司
地址 德国慕尼黑

(72)发明人 理查德·加里·麦克丹尼尔

(74)专利代理机构 北京康信知识产权代理有限
责任公司 11240
代理人 沈敬亭 陈方鸣

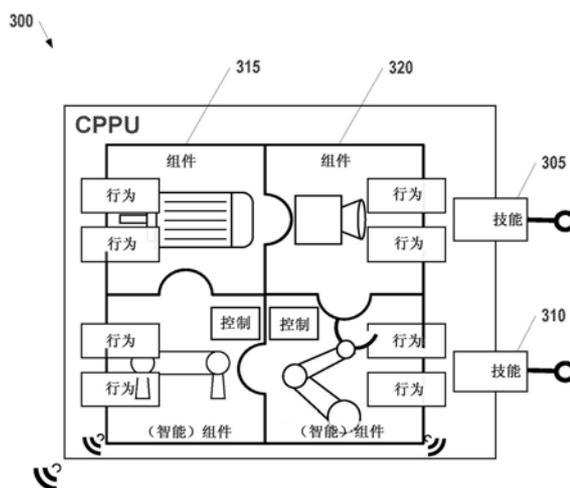
权利要求书2页 说明书9页 附图9页

(54)发明名称

用于工业应用的技能接口

(57)摘要

一种信息物理生产系统,包括多个信息物理单元,这些信息物理单元被配置为共同生产包括一个或多个工件的产品。每个信息物理单元包括一个或多个自动化系统设备、网络接口和处理器。网络接口被配置为接收一个或多个技能实例。每个技能实例提供由一个或多个自动化系统设备对工件进行转换的与机器无关的请求。处理器被配置为通过应用对自动化系统设备进行控制的行为来执行一个或多个技能实例中的每一个。



1. 一种信息物理生产系统,包括:
多个信息物理单元,被配置为共同生产包括一个或多个工件的产品,每个信息物理单元包括:
一个或多个自动化系统设备,
网络接口,被配置为接收一个或多个技能实例,每个技能实例提供由所述一个或多个自动化系统设备对工件进行转换的与机器无关的请求,以及
处理器,被配置为通过应用对所述自动化系统设备进行控制的行为来执行所述一个或多个技能实例中的每一个。
2. 根据权利要求1所述的信息物理生产系统,其中,每个信息物理单元的所述网络接口包括使用表述性状态转移 (REST) 协议来接收一个或多个技能实例的网络服务接口。
3. 根据权利要求1所述的信息物理生产系统,其中,每个信息物理单元还包括:
单元指令队列,被配置为在由所述处理器执行之前存储所述一个或多个技能实例。
4. 根据权利要求3所述的信息物理生产系统,其中,所述处理器还被配置为在将技能实例输入所述单元指令队列中之后,基于优选序列对所述单元指令队列重新排序。
5. 根据权利要求3所述的信息物理生产系统,其中,所述处理器还被配置成按顺序执行所述一个或多个技能实例,跳过不能立即执行的技能实例。
6. 根据权利要求3所述的信息物理生产系统,其中,每个信息物理单元还包括:
扫描仪设备,被配置为读取所述工件上的物理标签,所述物理标签指定对应于所述工件的统一资源标识符 (URI),
其中,所述处理器在执行期间使用URI将所述单元指令队列中的所述一个或多个技能实例与所述工件相关联。
7. 根据权利要求6所述的信息物理生产系统,其中,所述一个或多个技能实例包括与对应于所述工件的所述URI相匹配的关键技能值。
8. 根据权利要求3所述的信息物理生产系统,其中,所述处理器还被配置为在从用于执行的所述单元指令队列中移除技能实例时,向所述信息物理生产系统发送指示所述技能实例完成的消息。
9. 根据权利要求3所述的信息物理生产系统,其中,所述处理器还被配置为在从用于执行的所述单元指令队列中移除技能实例时,修改所述技能实例的技能状态参数。
10. 根据权利要求1所述的信息物理生产系统,其中,所述信息物理单元中的至少一个对应于传输系统,所述传输系统有助于工件在所述信息物理生产系统中所包括的其它信息物理单元之间传输。
11. 根据权利要求1所述的信息物理生产系统,其中,所述信息物理单元中的每个包括可编程逻辑控制器。
12. 一种使用技能接口控制信息物理单元的计算机实现的方法,所述方法包括:
接收技能实例,所述技能实例提供由所述信息物理单元中所包括的一个或多个自动化系统设备对工件进行转换的与机器无关的请求;
将所述技能实例输入到包括一个或多个附加技能实例的单元指令队列中;
通过应用对所述自动化系统设备进行控制的行为来选择性地执行所述单元指令队列中的每个技能实例。

13. 根据权利要求12所述的方法,还包括:

在将技能实例输入所述单元指令队列中之后,基于优选序列对所述单元指令队列重新排序。

14. 根据权利要求12所述的方法,其中还包括按顺序执行所述单元指令队列中的技能实例,跳过不能立即执行的技能实例。

15. 根据权利要求12所述的方法,还包括:

由扫描仪设备读取所述工件上的物理标签,所述物理标签指定对应于所述工件的统一资源标识符 (URI),

在执行期间使用URI将所述技能实例与所述工件相关联。

16. 根据权利要求12所述的方法,其中,使用与所述技能实例中包括的URI匹配的关键技能值,将所述URI与所述技能实例相关联。

17. 根据权利要求12所述的方法,还包括:

在从用于执行的所述单元指令队列中移除所述技能实例时,向信息物理生产系统发送指示所述技能实例完成的消息。

18. 根据权利要求12所述的方法,还包括:

在从用于执行的所述单元指令队列中移除技能实例时,修改所述技能实例的技能状态参数。

19. 一种信息物理生产系统,包括:

传输信息物理单元,包括:

物理传输系统,

传输网络接口,被配置为接收一个或多个传输技能实例,每个传输技能实例提供在所述信息物理生产系统中所包括的多个其他信息物理单元之间传输工件的与机器无关的请求,以及

处理器,被配置为将一个或多个传输行为应用于所述物理传输系统以实现所述传输技能实例。

20. 根据权利要求19所述的信息物理生产系统,还包括:

一个或多个加工信息物理单元,每个加工信息物理单元包括:

物理机器,用于物理地转换所述工件,

加工网络接口,被配置为接收一个或多个加工技能实例,每个加工技能实例提供由所述加工信息物理单元加工所述工件的与机器无关的请求,以及

处理器,被配置为将一个或多个加工行为应用于所述物理机器以实现所述加工技能实例。

用于工业应用的技能接口

技术领域

[0001] 本发明主要涉及用于工业应用的技能接口,以及与之相关的方法、系统和装置。所公开的技术可以应用于例如使用可编程控制器的各种自动化生产环境。

背景技术

[0002] 制造过程高度自动化,可以分成几个层次。例如,在最高级别,进行企业资源规划(ERP),其可以被称为业务层。在较低级别,进行硬件实现和控制,其可以被称为各种控制或单元层。中间层集成并连接业务层和控制层。该中间层包括制造执行系统(MES),其根据数据和功能之间的交互来定义MES流程,例如资源管理、资源分配、调度、数据收集和获取、质量保证管理、维护管理、性能分析、调度、文档控制、人工管理以及材料和生产跟踪。

[0003] 设计工业应用的一个挑战是机器对机器的交互高度依赖于MES如何实现工业应用的过程。结果是整个系统的稳固性受到限制。当需要定制时,必须手动执行。例如,工业环境中的操作员或其他工人可以阅读工作指令(work order),为特定的工艺步骤拾取输入材料,将这些材料送到机器上,并运行机器来完成该步骤。在某些情况下,诸如起重机之类的机器可能有助于物品的传输;但是原则上,系统的任何定制仍然依赖于人类与系统的交互。

发明内容

[0004] 本发明的实施例通过提供与工业应用的技能接口相关的方法、系统和装置来解决和克服一个或多个上述缺点和缺陷。这里描述的技术在称为“技能”的抽象的角度定义了自动化应用程序中多个交互设备之间通信意图的问题。技能的目的是将自动化过程中建立和执行过程的过程进行标准化和简化。它还用作系统级机器之间的通信介质,并确定要执行哪些操作来计算优化。技能与人们在MES中找到的过程相关,但可以在单元级别实现。

[0005] 根据本发明的一个方面,一种信息物理生产系统包括多个信息物理单元,信息物理单元被配置为共同生产包括一个或多个工件的产品。在一些实施例中,每个信息物理单元包括可编程逻辑控制器。每个信息物理单元包括一个或多个自动化系统设备、网络接口和处理器。网络接口被配置为接收技能实例,技能实例提供由自动化系统设备对工件进行转换的与机器无关的请求。在一些实施例中,网络接口包括使用表述性状态转移(REST)协议来接收技能实例的web服务接口。处理器被配置为通过应用控制自动化系统设备的行为来执行每个技能实例。在信息物理生产系统的一些实施例中,至少一个信息物理单元对应于传输系统,该传输系统有助于工件在系统中所包括的其它信息物理单元之间传输。

[0006] 根据前述信息物理生产系统的一些实施例,每个信息物理单元还包括单元指令队列,该单元指令队列被配置为在处理器执行之前存储技能实例。处理器还可以被配置为在将技能实例输入到单元指令队列中之后,基于优选序列对单元指令队列重新排序。此外(或可替换地),处理器还可以被配置成按顺序执行技能实例,跳过不能立即执行的技能实例。在前述信息物理生产系统的一些实施例中,处理器还被配置为在从执行的单元指令队列中移除技能实例时,向信息物理生产系统发送指示该技能实例完成的消息。处理器还可以被

配置为在从执行的单元指令队列中移除技能实例时,修改技能实例的技能状态参数。

[0007] 在一些实施例中,每个信息物理单元还包括被配置为读取工件上的物理标签的扫描仪设备。该物理标签指定与工件相对应的统一资源标识符 (URI)。处理器可以在执行期间使用URI将单元指令队列中的技能实例与工件相关联。例如,在一个实施例中,技能实例包括与对应于工件的URI相匹配的关键技能值。

[0008] 根据本发明的另一方面,一种用于使用技能接口来控制信息物理单元的计算机实现的方法包括接收技能实例,该技能实例提供由信息物理单元中包括的一个或多个自动化系统设备对工件进行转换的与机器无关的请求。技能实例被输入到包括一个或多个附加技能实例的单元指令队列中。一旦输入技能实例,队列可能被重新排序。例如,在一个实施例中,该方法包括在单元指令队列中输入技能实例之后,基于优选序列对单元指令队列重新排序。然后,通过应用控制自动化系统设备的行为来选择性地执行单元指令队列中的每个技能实例。在一些实施例中,按顺序执行单元指令队列中的技能实例,跳过不能立即执行的技能实例。在从执行的单元指令队列中移除技能实例时,可以向信息物理生产系统发送指示技能实例完成的消息,或者可以修改技能实例的技能状态参数。

[0009] 根据本发明的其他实施例,信息物理生产系统包括传输信息物理单元,该传输信息物理单元包括物理传输系统、传输网络接口和处理器。传输网络接口被配置为接收传输技能实例,该传输技能实例提供了在信息物理生产系统中所包括的其他信息物理单元之间对工件进行传输的与机器无关的请求。处理器被配置为将传输行为应用于物理传输系统以实现传输技能实例。在一些实施例中,前述信息物理生产系统还包括一个或多个加工信息物理单元。每个加工信息物理单元包括用于物理转换工件的物理机器、被配置为通过提供由加工信息物理单元加工工件的与机器无关的请求来接收一个或多个加工技能实例的加工网络接口以及被配置为将加工行为应用于物理机器以实现加工技能实例的处理器。

[0010] 本发明的其它特征和优点在参考附图继续阅读以下对说明性实施例的详细描述后将变得显而易见。

附图说明

[0011] 当结合附图阅读时,由下面详细说明可以最好地理解本发明的前述和其他方面。为了解释本发明,在附图中示出了当前优选的实施例,然而,可以理解,本发明不限于所公开的特定手段。附图中包括以下图:

[0012] 图1示出了根据一些实施例的在自主机器之间自动传输材料的示例性信息物理生产系统 (CPPS);

[0013] 图2提供了根据一些实施例的如图1所示的示例CPPS的信息物理单元 (CPPU) 和相关技能的高级概要图;

[0014] 图3提供了根据一些实施例的单个CPPU 300的组成的示例;

[0015] 图4示出了一个示例场景,其包括与其他传感器和推动器一起布置的一组三个输送机;

[0016] 图5示出了CPPU的本体,其可以在本发明的一些实施例中实现。CPPU包括行为、组件和技能;

[0017] 图6示出了根据一些实施例的添加到单元的指令队列中的技能实例的示例;

[0018] 图7示出了标识工作托盘的QR码标识符的示例；

[0019] 图8示出了技能接口的示例，其中数据流语言被用来通过指定技能和行为之间的内部连接来实现编程行为；以及

[0020] 图9提供了另一个数据流示例，其示出了比图8所示更复杂的技能行为连接。

具体实施方式

[0021] 以下公开根据若干实施例描述了本发明，这些实施例针对与有助于自主过程中的通信的技能接口相关联的方法、系统和装置。这里使用的术语“技能”指的是工件或材料需要如何被转换以最终转化为最终产品的与机器无关的描述。支持技能接口的设备在此被称为信息物理生产单元 (CPPU)，多个CPPU的组合活动由信息物理生产系统 (CPPS) 控制。

[0022] 这里描述的技能接口可用于控制单元组以协作且自动地创建产品。工作产品可以基于例如优化的成本被传送到不同的机器，或者可以利用相同的单元套件来创建多个产品，甚至可以同时进行。技能接口提供了内省和选择，因此自动化系统可以动态选择机器，而不依赖于固定的生产路径。给定的CPPU可以支持可应用于不同产品的几种不同技能，并且不同的CPPU可以支持共同的技能，这样系统可以选择哪个CPPU会更有效地使用。

[0023] 技能接口的实现涉及技能类的定义。例如，这可以包括技能的名称以及描述工件如何被转换的一组参数。CPPS可以创建技能实例，该实例描述用于特定工件的参数值。该实例可以是通用的，并且适用于任何工件，或者可以使用标识要影响哪个工件的关键字 (key)。实例存储在队列中，单元从该队列中检索实例并解包技能的参数。单元还可以在队列中预测，以选择下一个要执行的技能，因为并非所有技能都可以立即应用。然后，单元通过将其参数应用于控制各种组件、物理和功能设备的行为来执行这项技能，这些组件、物理和功能设备是实际执行工作的。多个技能可以同时执行，包括同一类别的技能，取决于该单元的物理限制。当技能完成后，可以将它们从队列中删除，并在完成时向MES发出信号。

[0024] 技能是用于在半自主生产单元套件之间传递产品信息的媒介。有许多理由使用大量的半智能设备来制造产品，而不是创建一个固定的、手工加工的、只有一个产品的工厂。例如，如果希望创建高度定制的产品或创建各种小批量的产品，则使用可定制配置的通用机器会更具成本效益，因为移除了为单个产品建造整个工厂的固定成本。

[0025] 图1示出了根据一些实施例的在自主机器之间自动传输材料的示例性信息物理生产系统 (CPPS) 100。在图1中，传输系统在装载/卸载站105、钻孔机110和铣床115之间运送材料和工作产品，这些设备分别执行将工件加工成产品的过程。

[0026] 生产系统被分成多个单元，每个单元可以执行整个过程的某些方面。例如，铣床115可用于形成产品的整体形状。钻孔机110可用于在产品中钻孔，这些钻孔对于铣床115来说太精细而无法制造。装载/卸载站105保持货物和材料，直到它们需要被部署在其他机器上。传输系统将工作产品移入和移出进行加工的机器。CPPS 100的各个部分被称为信息物理生产单元 (CPPU)。CPPU是自主和智能的，因此他们提供的大部分工作可以在没有监督或特殊指导的情况下进行。

[0027] 在图1的示例中，注意传输系统的元件 (例如将工件从主输送机运送到钻孔机的装置) 被认为是钻孔机110的CPPU的一部分。它可能同样容易被认为是传输单元的一部分。机器之间的分工可能是任意的，但通常由系统集成商突发奇想以外的几个因素决定。例如，钻

孔机110的部件更换系统可以物理硬连线或以其他方式连接到钻孔机。如果要重新配置该过程,将钻孔部件装载器与钻子分开可能没有意义,因为装载器可能是定制的且不会用于任何其他目的。此外,钻子和装载器的结合可以使钻孔机110作为一个整体更加自主。如果没有装载器,钻子将需要一些其他设备或人员将材料放在里面进行钻孔。

[0028] 随着CPPU确定,该过程的目标随后可以被传达给执行这些目标的设备。CPPU的技能随后被用来传达对工件采取什么行动。

[0029] 图2提供了根据一些实施例的如图1所示的示例CPPS的CPPU和相关技能的高级概要图。每项技能都反映了应用于工作产品的操作以及每个单元的关键用途。因此,装载/卸载站105实现装载技能205A和供应技能205B。钻孔机110和铣床115分别实现钻孔技能210和铣削技能215。最后,输送机系统具有传输技能220。为了运行该过程,工作产品通过传输技能220被运送到各个单元。使用铣削技能215成型产品;通过钻孔技能210对产品钻孔,等等。规划系统可用于根据需要向正确的单位发送正确的技能信息。

[0030] 图3提供了根据一些实施例的单个CPPU 300的组成的示例。CPPU是产品创建和修改的地方。它被认为采用了信息技术,因为它由作为CPPS一部分应用于它的技能控制,并且通常是完全自动化的;但是,在一些实施例中,也可以使用人工辅助。CPPU旨在自主地工作,使得它不需要来自其他CPPU的动作就能在传输系统之外起作用,而传输系统可能会将工作产品运进单元和从单元运出。CPPU应该也是智能的,因为它有能力从应用技能中解释所需的产品信息,并且能够执行自己的行为来实现这一结果。

[0031] 图3中示出的CPPU包括两个组件315、320,其包括执行工作的物理资产以及辅助设备,辅助设备供给材料、更换工具、喷洒化学药品、感测位置以及单元需要做的任何其他事情。这些物品被称为组件。图3的示例示出了包括在组件315、320中的机器人、照相机、马达和输送机传送带;然而,应该理解,任何物理资产通常都可以包括在CPPU的一个组件中。

[0032] 技能305、310充当CPPU 300的接口。在图3中,技能305、310用接口符号来描述,以指示从外部进行通信,从而将技能应用于CPPU。通信介质可能是一种允许任意和可变长度消息的网络协议。例如,向web服务器发送数据和从web服务器发送数据的消息将满足这种协议。

[0033] 在告诉机器需要做什么的技能305、310和执行工作的组件315、320之间是包括描述如何实际执行工作的所有内在和编程功能的行为。行为被定义为组件所执行的基本活动。该活动可以是物理的或计算的。例如,作为一种物理行为,机器人手臂可以在空间移动它的末端执行器。如果还考虑用于确定机器人移动到的终点的函数,则也可计算该行为,因为需要计算该点。虽然可以区分纯物理行为和抽象计算行为,但原则上,大多数应用行为将是两者的结合。因此,行为可以被定义为生产组件做什么,而不管这可能会如何影响工作产品的状态。例如,输送机可以被打开和关闭。它的行为是在打开时运行传送带。如果传送带上碰巧有产品,那么产品会移动,但是输送机本身的行为与这个问题无关。无论产品是否在那里,输送机都能运行。

[0034] 当然,组件的组合活动确实会影响工作产品。以输送机为例,输送机的行为是运行,输送机顶部的东西会移动。图4示出了一个示例场景,其包括与其他传感器和推动器一起布置的一组三个输送机。这个应用的目的可能是对输入的产品进行分类。如果产品被引入IN输送机右侧的灰色盒子位置,传感器可以确定盒子是应该向下传送到A输送机还是B输

送机上(以及到其他流程上)。推动器的行为是伸出和缩回,如果在适当的时间伸出,它会将盒子推到输送机B上。这些组件的组合行为改变了工作产品的状态;盒子从未分类状态变为已分类状态。因此,这个应用的“技能”是对盒子进行分类。这种设备的技能可以表述为“SORT(box_id,dest)”,或者换句话说,给定一个特定的盒子,把它放在一个特定的目的输送机上。这种技能的一个实例可以是“SORT(box_id=1007,dest=B)”,它会将1007号盒子放在传送带B上。

[0035] 其他类型的技能会更直接地影响工作产品。例如,可以通过切削、铣削或钻孔来改变工件。多个工件可以组装成一个组合部件,并且可以胶合或紧固在一起。一般来说,技能通过声明如何修改工作产品来决定生产系统中生产单元的用途。因此,技能实际上是一种向外界展示的命令,它阐明了该单元应该做什么。

[0036] 图5是CPPU的本体,其可以在本发明的一些实施例中实现。CPPU包括行为、组件和技能。这些行为被定义为描述组件如何工作以及它们如何一起工作的实体。因此,应用程序的逻辑也与组件的物理行为一起被视为行为。组件还可以包含行为,因为它们具有物理行为,但是它们也可以包含控制其使用的更复杂的功能。例如,机器人可能有一个运动规划功能,可以用更高级别的命令来控制它的行为。由于工作产品是一个物理实体,它也被视为一种组件。请注意,工件本身甚至可以包含计算元素,如检测其内容或确定其自身路由目的的托盘。尽管如此,即使是普通的工件也会有物理行为和状态。

[0037] 组件的行为导致工作产品被修改。因此,尽管即使没有工件存在,行为也可能会发生,但组件的行为在工件上的应用仍然会让机器做任何事情。CPPU提供技能作为描述它能做什么和引发实际导致这些事情发生的行为的技术。技能本身描述了在给定的产品上希望获得什么结果,尽管导致结果发生的特定事件链是由行为引起的。

[0038] CPPU为外部设备提供技能。这些技能启动行为中包含的动作,并描述将应用于产品的结果。因为技能可以与行为区分开来,所以人们可以为CPPU提供技能接口,而不必暴露实际执行任务的行为。这很方便,因为它允许使用通用接口协议指定技能,而单元的行为通常需要用特定领域的语言来实现。因此,实现一项技能是关于获得最匹配所需工件转换的技能的抽象表示,然后将该抽象的编程版本附加到执行任务的实际行为函数上。

[0039] 可能令人困惑的是,技能一词没有暗指它在应用中的使用方式。技能可能指任务的抽象定义—这类似于它的类或本体。技能也可以指应用于特定单元的抽象实例。例如,“用红色块装满托盘”是涉及托盘和块的特定任务的技能实例。此外,技能可能是指技能在单元上的执行。技能是用行为来进行的;然而,由于在单元上设置的技能实例与该单元上为完成该技能而采取行动的相应行为之间存在直接的因果关系,因此可以认为该技能正在被执行。在本公开中,当从上下文中不明显时,将阐明技能的这些不同含义。

[0040] CPPU可以在可编程逻辑控制器或与CPPU的组件紧密耦合的另一个单元级计算设备上实现。原则上,人们可以使用任何类型的计算机控制媒体将技能实例从CPPS传送到CPPU,但是网络协议可能是最直接的。网络协议例如web服务的好处包括易于扩展的能力,并且可以与通用通信设备和编程库一起使用。网络协议也可以具有任意的消息长度,具有众所周知的数据编码和解码方法。在这里描述的示例中,假设使用表述性状态转移(REST)协议和使用预定义的统一资源标识符(URI)标签经由web服务接口来进行通信。然而,应当理解,在本发明的其他实施例中可以采用提供类似功能的替代技术。

[0041] 技能可以由系统集成商来实现,因此,不能仅仅基于过程的类型来预测。任何给定过程的细节都将是独一无二的,因为制造商会运用他们自己的知识来为他们的产品增加独特的价值。出于这个原因,可以使用简单的技能协议,使集成者能够将他们自己对过程的想法直接反映到单元的代码结构中。向技能数据集添加更多的信息和语义当然是可能的,但是应该将其视为核心规范的附加内容,而不是开发技能的基本需求。

[0042] 基本技能定义是带有一组参数的名称。这个名称可以像文本字符串一样简单,表示要执行行为实现中的哪个技能。参数集更有趣。通常,参数名称可以是字符串值,如技能名称。参数可以是可选的,对于不同的用例,给定的技能可能会因不同的参数集而过载。参数值可以是常量值或常量表达式。同样,参数值可以保持简单,例如单例浮点数、布尔值或整数。使用简单参数可使技能实例紧凑且高效地与单元通信并存储在技能队列中(如下所述)。

[0043] 技能实现包括将技能实例应用于单元的方法以及用于发现可用技能的反射方法。这里使用的术语“反射”指的是编程语言在运行时检查和动态调用类、方法、属性和其他程序相关信息的能力,而不需要关于被检查项目的先验的特定知识。反射应该至少包括所有技能及其参数的列表,还可以包括参数的类型信息(其包括值限制)、参数是否可选、工程单位和其他约束标准。技能发现可能会在首次激活设备时发生,因此技能列表不应经常更改。大多数单元在运行过程中会提供相同的技能组,除非设备重新编程或系统本身重新配置,否则不会改变。

[0044] 因为CPPU是自主的,所以通常不会像函数调用一样从外部调用技能。CPPU将能够检测自己的状态,并根据自己的能力确定工件的状态,不会被外界过多地引导。当工件和组件通过单元自身的智能准备好时,它就会执行其功能。一个简单的解决方案是将每个技能应用视为一种工作指令,并将其输入单元的指令队列中。

[0045] 图6提供了根据一些实施例的被添加到单元的指令队列的技能实例的示例。当技能实例被添加到单元中时,它会进入单元的指令队列,如图中所示。一旦进入队列,该单元就可以查看技能实例的内容以及它是哪种技能。可以对技能的实例进行排序,以便按特定顺序跟踪技能,或者单元可以跳过不能立即为队列中的后者执行的实例。为实现通用技能,单元可以访问队列头部的实例的参数。单元还可以扫描队列以查找其可以使用的实例。为此,单元将使用索引变量迭代队列。出于这个原因,这种技能被称为“索引技能”,因为单位根据其在队列中的位置挑选技能实例。队列的头部将是索引零。当单元完成指令后,它可以从队列中删除技能实例。技能移除可以作为一条“完成”消息传递给CPPS。CPPS还可以在技能中使用回调、系统消息或可变参数来监视以确定技能状态。

[0046] 在一些实施例中,CPPS被配置为使得其不将技能实例应用于不能实际执行默示契约的单元。从CPPS的角度来看,将工件发送到不能与之一起工作的单元的计划不好,并且会被计划过程本身所阻止。在其他实施例中,当向一个单元提供了一个糟糕的技能实例时,可以提供一种错误机制,并且CPPS可以被配置为将工作材料重新路由到不同的单元。

[0047] 当一个单元中断、关闭或离线时,也会出现类似的困境。在这种情况下,所有待定技能实例都将从单元中清除,单元的通信线路将进入错误返回状态。CPPS可能在某个时候从其数据库中删除单元的条目,并在新拓扑下重新规划。显示故障可能不是单元本身的技能,但可能是需要传达的基本单元状态。

[0048] 操作员、工程师或其他集成者可以用多种不同的方式来设置技能的实现,但是一个常见的情况是工作产品(通常是工作产品的载体)由某种物理标签来标识。该标签可以是条形码、QR码、RFID标签、印刷号码等,其将由CPPU中包括的扫描仪或阅读器设备读取。图7中示出了识别工作托盘的QR码标识符的示例。标识符的值可以是例如数字或字母数字字符串。在一些实施例中,“关键技能”用于指定由这种标签识别的工作产品的参数。关键技能提供了一个特殊参数,即关键字,其值将与单元从工作产品标签中获得的标识符相匹配。使用此功能,单元可以快速挑选出与其手头的工件相对应的技能实例。在使用这种技能的过程中,CPPS会在该单元准备好后立即添加所有适用于单元的技能实例,并让单元决定在给定时间应用哪个技能实例。在一些实施例中,可以使用通配符或本领域通常已知的其他搜索方案来通过它们的关键字选择技能实例。

[0049] 如上所述,单元的内部编程由行为形成。因此,实现一项技能的意图就是执行操作完成工作的组件的行为。可以用不同的语言对一个单元进行编程;然而,一般来说,本领域已知的任何编程语言都可以用来定义技能类、从队列中挑选一个技能实例以及读取将在程序的其余部分中使用的实例参数。在CPPU中,可以使用混合编程语言。例如,在一些实施例中,可以使用相同的编程语言或函数来定义技能和行为。在其他实施例中,用于定义技能和行为的编程语言或函数可以是不同的。

[0050] 图8示出了技能接口的示例,其中数据流语言被用来通过指定技能和行为之间的内部连接来实现编程行为。在数据流的说法中,输入值被应用于图中的一个节点。节点执行计算,结果显示在输出值上。然后,输出可以继续到进行更多计算的其他节点,以此类推。在此示例中,技能定义节点使用两个输入值,并提供两个输出值。第一个输入值是一个标记,当它被激活时,将导致节点搜索队列中的技能。节点定义技能名称的配置和技能使用的参数名称列表。在这种情况下,技能名称是“倾卸盘”,它有一个名为“色彩”的技能参数(未示出)。第二个输入值是一个整数,用于标识应该选择队列中的哪个技能实例。对于关键技能类型,第二个输入值是与技能实例匹配的关键字的值。如果节点找到一项技能,它会将第一个输出值设置为真,表示节点处于活动状态。它还将技能实例参数的值按与参数列表相同的顺序放入第二个输出值中。该示例显示了一个用于挑选第一个参数值(数字“1”)的数组索引节点,它显示在文本框中。当程序将技能节点输入标记设置为假时,可以将节点配置为自动删除最后显示给其输出的技能。

[0051] 图9提供了另一个数据流示例,其示出了比图8所示更复杂的技能行为连接。此示例包含用于从技能实例获取信息、确定单元是否具有执行技能所必需的材料代码,最后,它指导单元的其他行为来执行动作。用不同语言实现技能的行为可能会产生类似的活动。图9所示的技能实现足以代表技能来执行行为。添加进一步的反馈也是合适的。当从队列中删除技能实例时,此示例指示完成。在一些实施例中,一个技能实例被标记为正在使用,并且一个值表示该行为在完成技能方面接近完成的程度。在这些实现中,可以为完成百分比提供另一个输入端口,尽管它仍然是单元中将为该端口生成值的行为的编程实现。

[0052] 尽管这里描述的技术利用了许多常见的技术,但是这些技术的实现方式不同于传统的方法。例如,在传统系统中找不到将CPPS与CPPU分离并将通信过程操作的过程定义为对工作产品的修改的整个体系结构。在通常的方法中,机器操作是根据机器的行为来描述的。不管工件如何受到影响,输送机都会被打开和关闭。全新的焦点将系统的设计从一系列

硬编码指令转变为一组知晓其角色并将其技能应用于手头任务的智能行为者。

[0053] 在技术层面,内置于控制器中的标准接口机制通常会像远程过程调用一样工作。这就是将对机器行为的访问导出到外部的的方式。可以暴露用于打开和关闭机器的开关,并且一些外部控制器可以执行该操作。这种效果旨在像编程语言中的函数调用或者将值写入共享内存一样直接。技能接口不是直接的。技能本身就像放在队列中的指令。机器自身的内部操作决定了指令在哪里以及如何应用于工件。相对于索引技能抽象,关键技能用于常见的情况,在这种情况下,工件可以通过标记来识别,这样相关的技能就容易定位。

[0054] 技能接口对自动化很有用,但也可以应用于优化和成本计算之外。技能可以用来向模拟系统传达意图,以确定该过程是否可能,以及它将如何影响机器和相关系统。通过这种方式,技能抽象成为从编程自动化设备到调度和规划的整个自动化过程不可或缺的一部分。

[0055] 这里描述的各种设备,包括但不限于控制层设备和相关计算基础设施,可以包括至少一个计算机可读介质或存储器,用于保存根据本发明实施例编程的指令,并用于包含此处描述的数据结构、表、记录或其他数据。这里使用的术语“计算机可读介质”是指参与向一个或多个处理器提供以执行指令的任何介质。计算机可读介质可以采取多种形式,包括但不限于非暂时性、非易失性介质、易失性介质和传输介质。非易失性介质的非限制性示例包括光盘、固态驱动器、磁盘和磁光盘。易失性介质的非限制性示例包括动态存储器。传输介质的非限制性示例包括同轴电缆、铜线和光纤,包括构成系统总线的导线。传输介质也可以采用声波或光波的形式,例如在无线电波和红外数据通信期间产生的声波或光波。

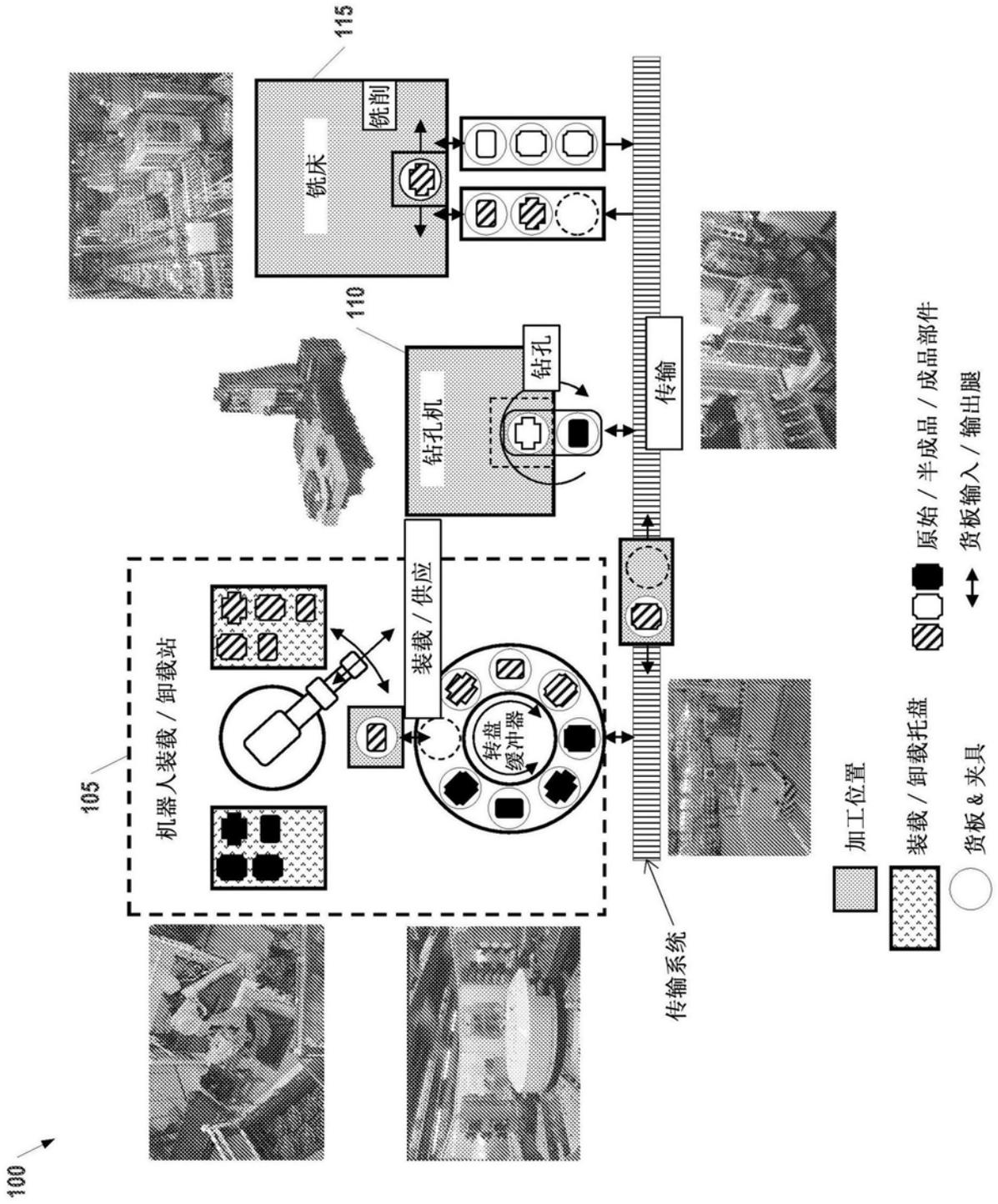
[0056] 前述控制层设备和相关计算基础设施可以包括用于实现此处描述的技术的一个或多个处理器。此处描述的由控制设备使用的处理器可以包括一个或多个中央处理单元(CPU)、图形处理单元(GPU)或本领域已知的任何其他处理器。更一般地,这里使用的处理器是用于执行存储在计算机可读介质上的机器可读指令的设备,用于执行任务并且可以包括硬件和固件中的任何一个或组合。处理器还可以包括存储器,其存储可执行以执行任务的机器可读指令。处理器通过操纵、分析、修改、转换或传输供可执行程序或信息设备使用的信息,和/或通过将该信息路由到输出设备,来实现对信息的操作。例如,处理器可以使用或包括计算机、控制器或微处理器的能力,并使用可执行指令进行调节,以执行通用计算机不执行的特殊目的功能。处理器可以与任何其它能够在它们之间进行交互和/或通信的处理器耦合(电耦合和/或包括可执行组件)。用户界面处理器或生成器是已知的元件,包括用于生成显示图像或其部分的电子电路或软件或两者的组合。用户界面包括一个或多个显示图像,使得用户能够与处理器或其他设备交互。

[0057] 在一些实施例中,部分的CPPU(包括技能接口)使用一个或多个可执行应用程序来实现。这里使用的可执行应用程序包括代码或机器可读指令,用于调节处理器以实现预定功能,例如响应于用户命令或输入的操作系统、上下文数据采集系统或其他信息处理系统的功能。可执行程序是用于执行一个或多个特定处理的代码或机器可读指令的片段、子程序、或者代码的其它不同片段或者可执行应用程序的一部分。这些过程可以包括接收输入数据和/或参数、完成对所接收的输入数据的操作和/或响应接收到的输入参数实现功能,以及提供最终输出数据和/或参数。

[0058] 这里的功能和处理步骤可以自动地、全部地或部分地响应于用户命令来执行。响

应于一个或多个可执行指令或设备操作,自动执行的活动(包括步骤)被执行,而无需用户直接启动该活动。

[0059] 附图的系统 and 过程不是排他性的。根据本发明的原理,可以派生出其他系统、过程和菜单来实现相同的目的。尽管已经参照特定实施例描述了本发明,但是应当理解,这里示出和描述的实施例和变型仅用于说明目的。在不脱离本发明范围的情况下,本领域技术人员可以对当前设计进行修改。如本文所述,可以使用硬件组件、软件组件和/或其组合来实现各种系统、子系统、代理、管理器和过程。本文中的任何权利要求都不应根据35U.S.C112,第六段的规定来解释。除非使用短语“用于...的装置”明确叙述该要素。



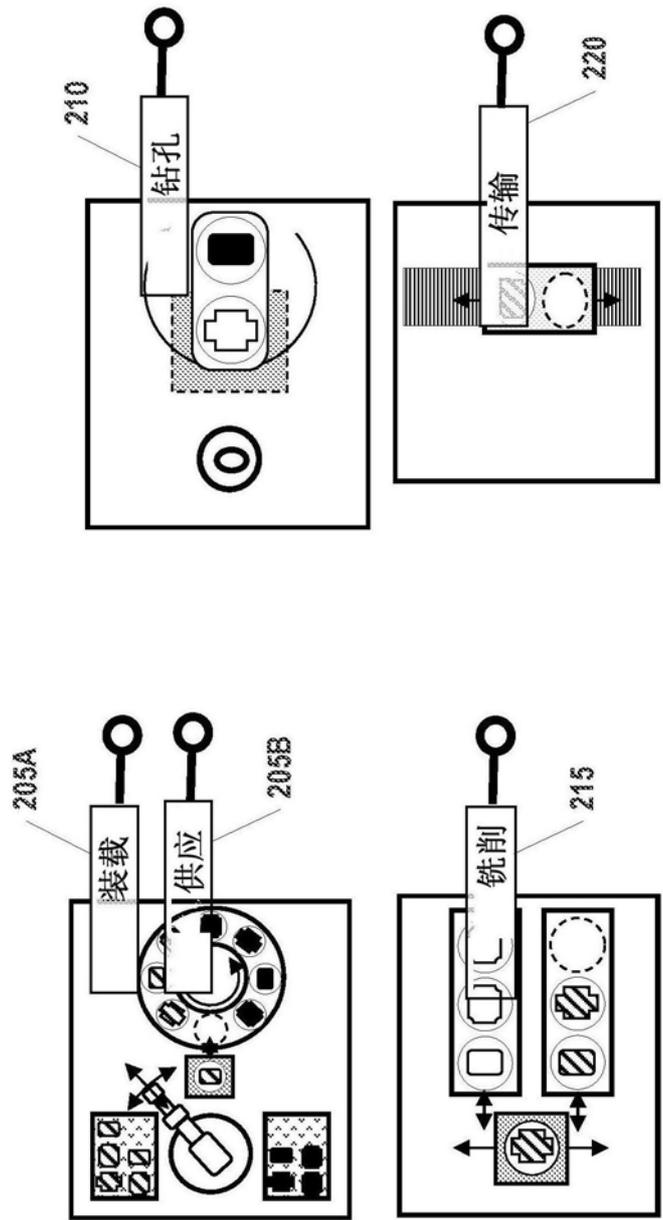


图2

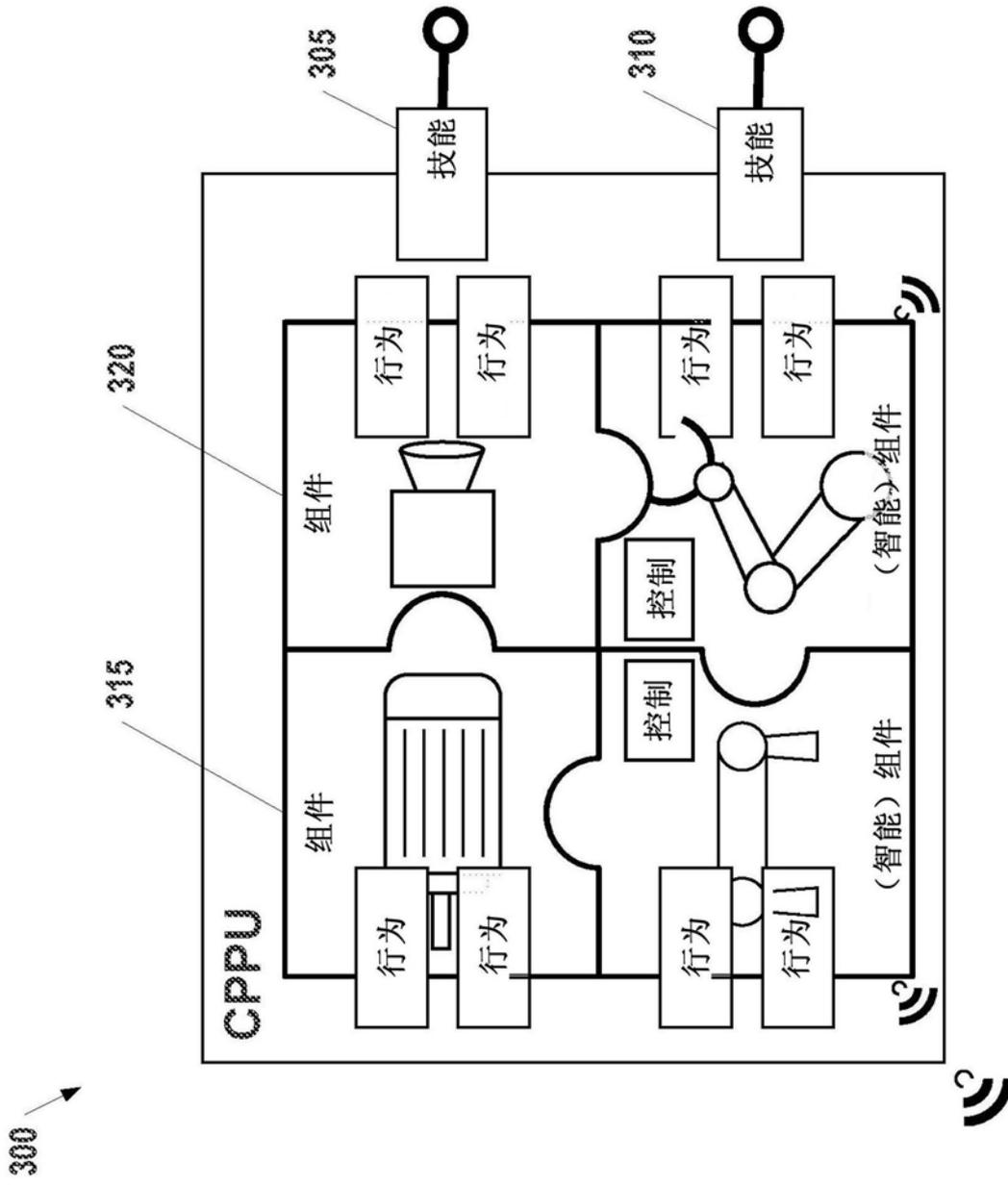


图3

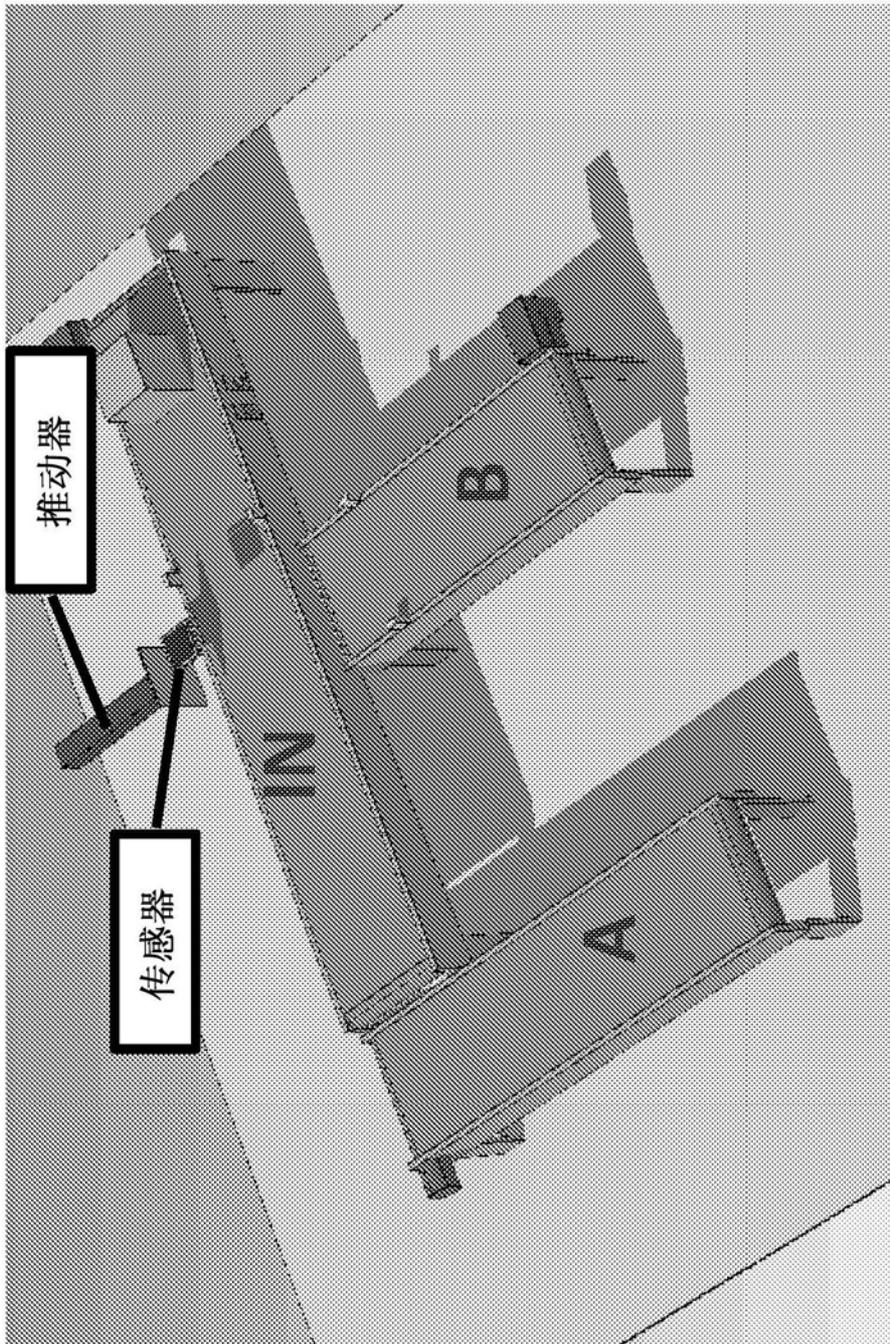


图4

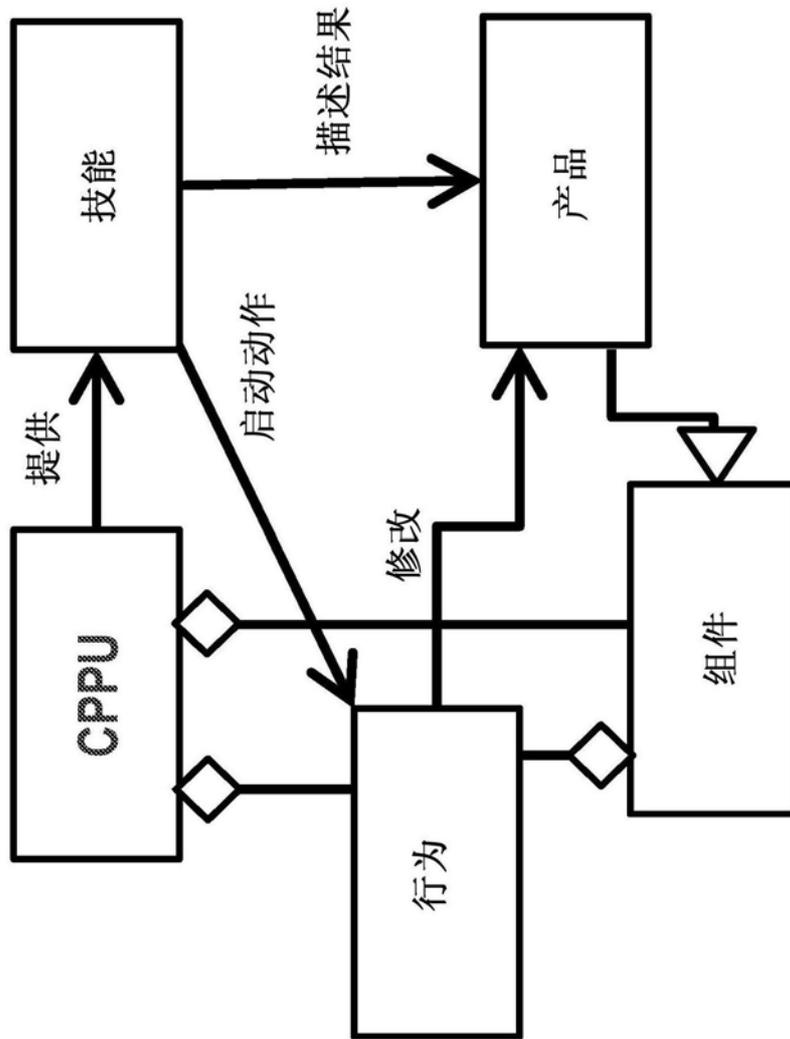


图5

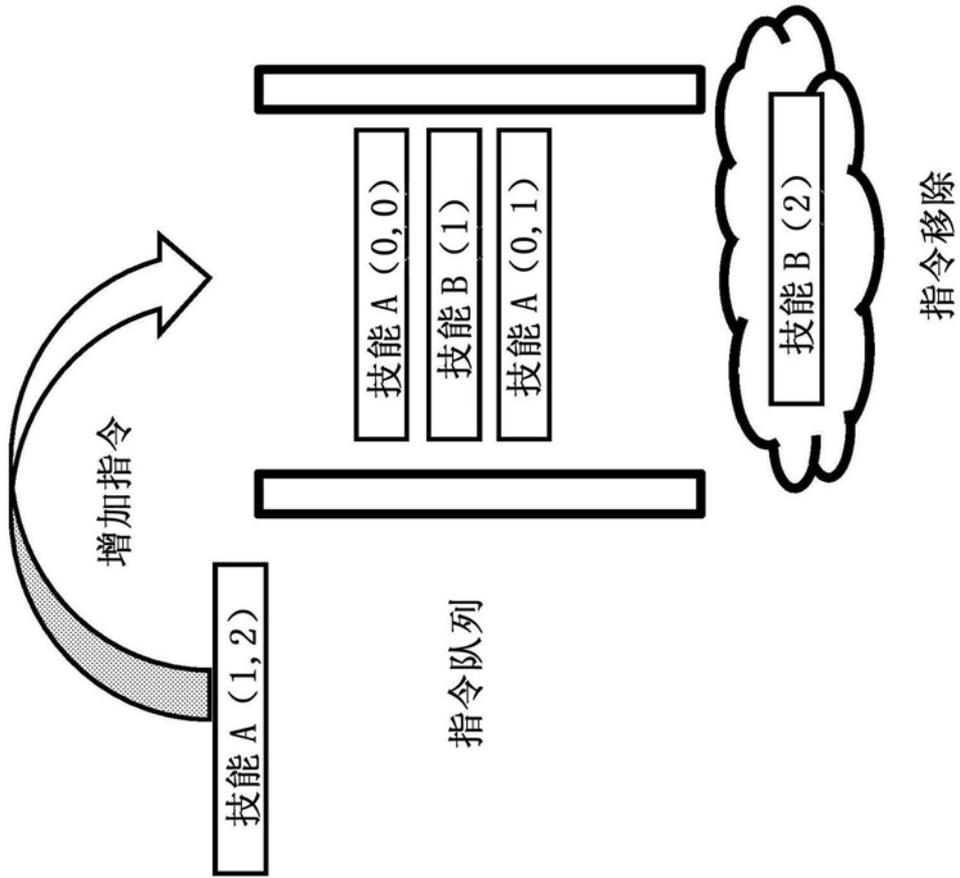


图6

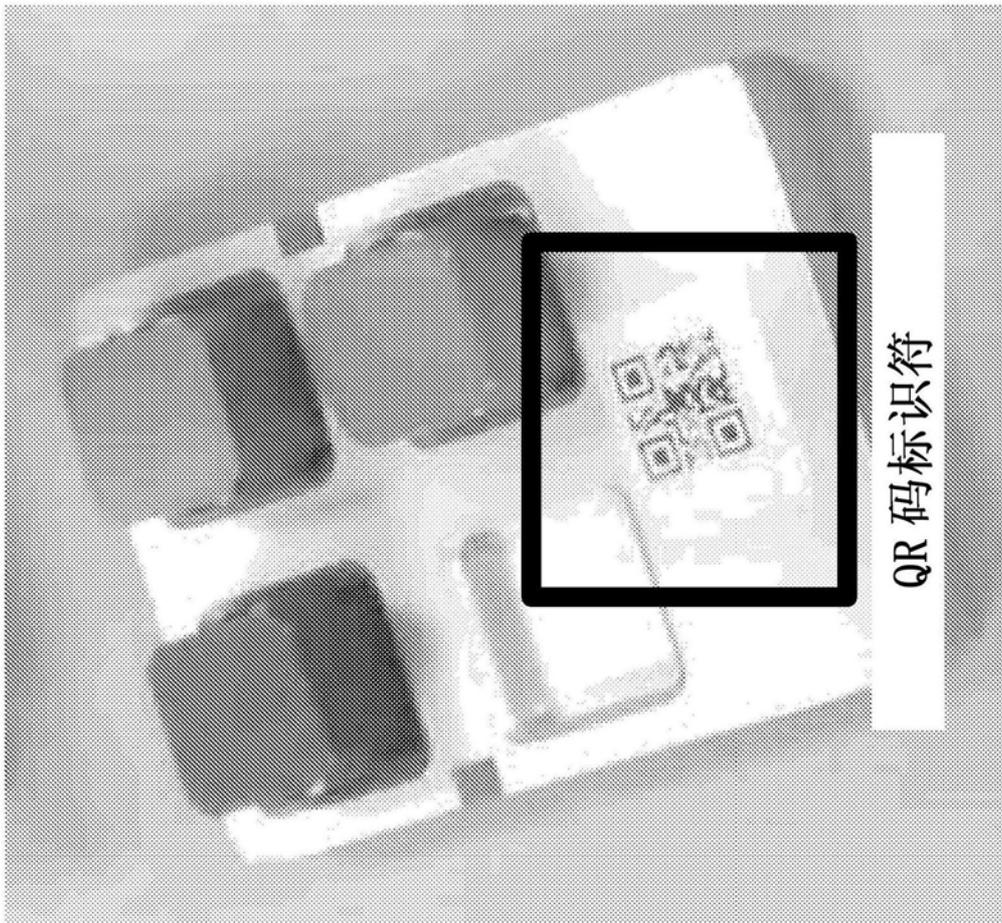


图7

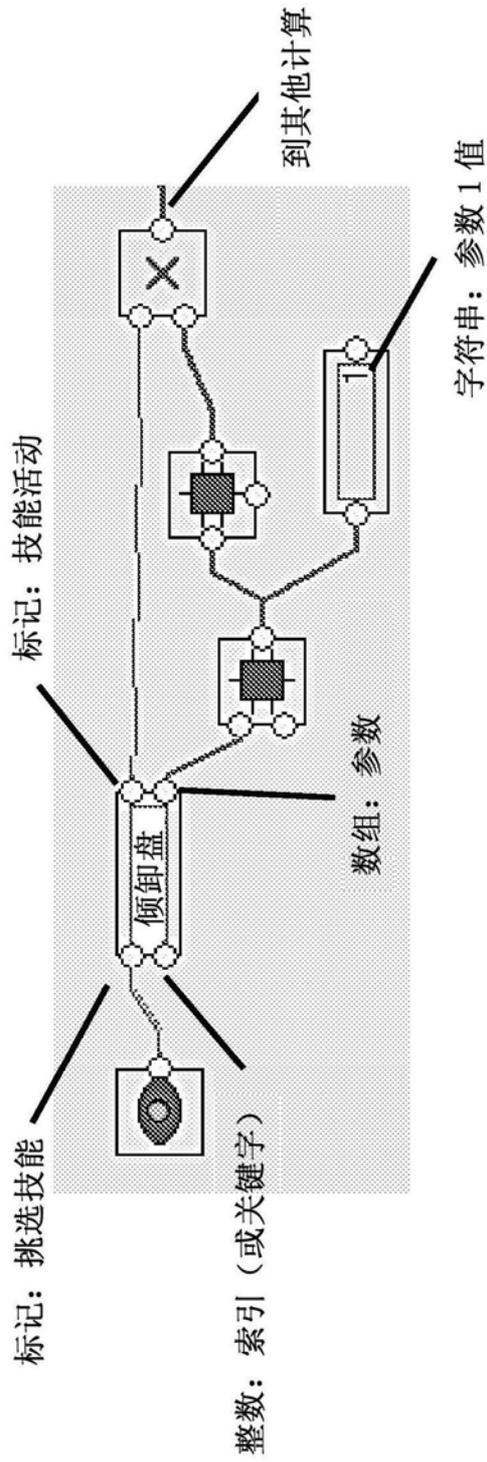


图8

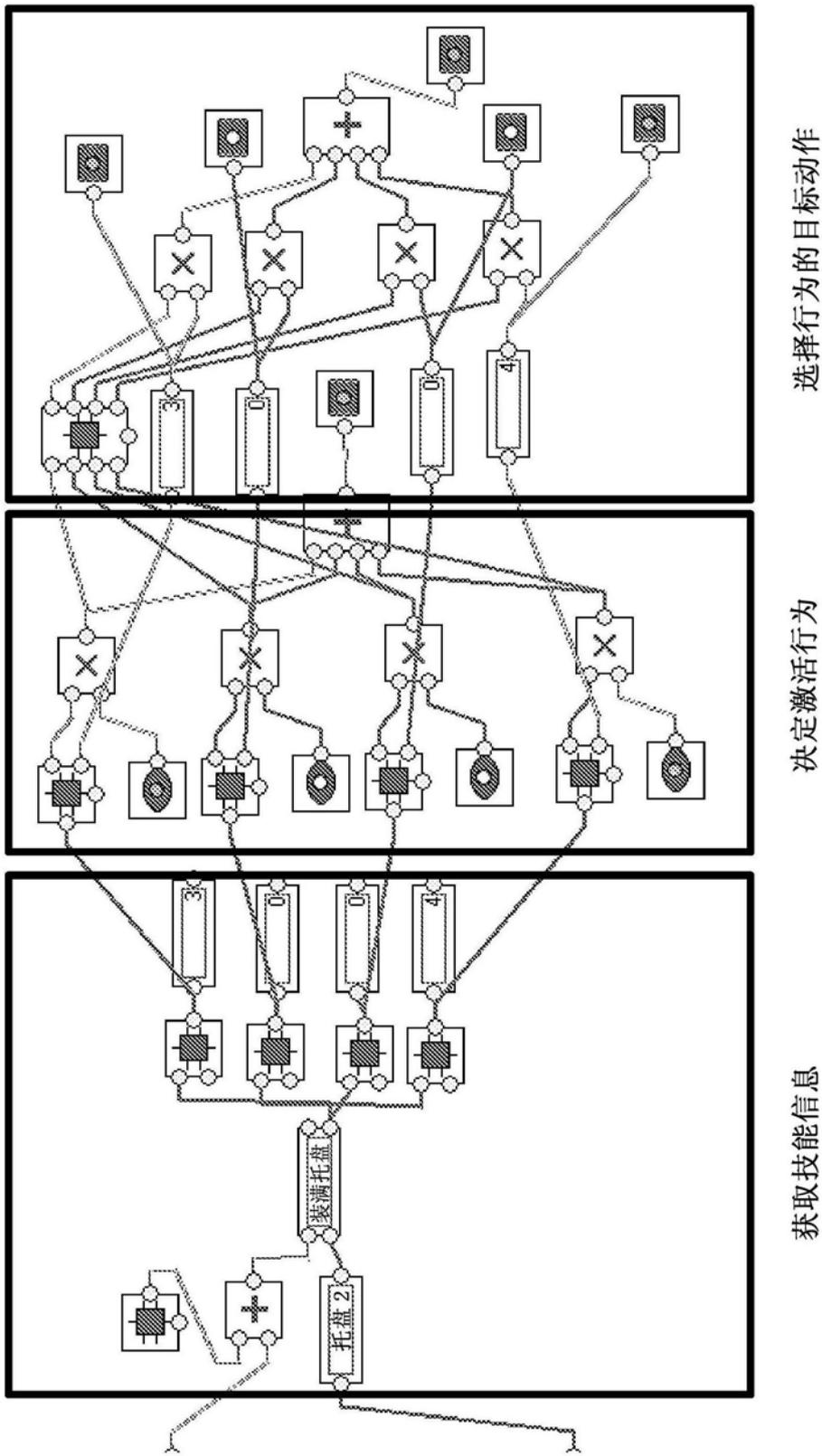


图9