



(12) 发明专利申请

(10) 申请公布号 CN 118633077 A

(43) 申请公布日 2024. 09. 10

(21) 申请号 202280083166.7

(22) 申请日 2022.10.14

(30) 优先权数据

21202933.4 2021.10.15 EP

(85) PCT国际申请进入国家阶段日

2024.06.14

(86) PCT国际申请的申请数据

PCT/EP2022/078690 2022.10.14

(87) PCT国际申请的公布数据

W02023/062209 EN 2023.04.20

(71) 申请人 西门子股份公司

地址 德国慕尼黑

(72) 发明人 M·呢 E·安东尼 V·巴拉

B·日 Y·卡马斯 A·K·日

H·赖

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

专利代理师 浩路 吕传奇

(51) Int.Cl.

G06F 8/72 (2006.01)

G06F 8/51 (2006.01)

G06F 8/36 (2006.01)

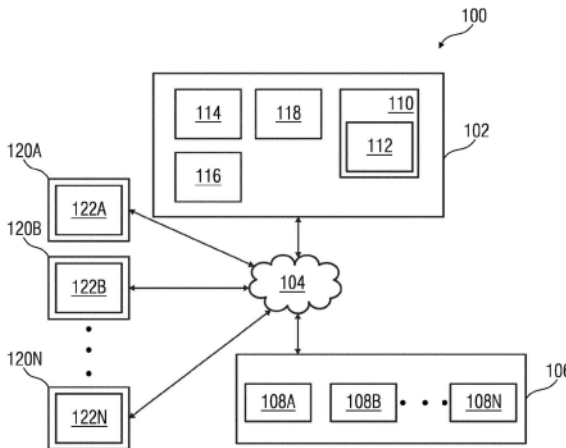
权利要求书3页 说明书15页 附图7页

(54) 发明名称

用于针对技术设施生成和优化工程程序的方法和系统

(57) 摘要

本发明提供了一种用于从另一工程程序来生成和优化具有给定规范的工程程序的方法和系统。该方法包括由处理单元(202)接收从具有第二规范的第二工程程序生成具有第一规范的第一工程程序请求。该方法进一步包括由处理单元(202)生成第一本体模式,该第一本体模式包括与第二工程程序中的多个代码语句之间的多个数据流和控制流相互关系相关联的信息。该方法进一步包括由处理单元(202)基于在第一本体模式的第一序列表示上应用机器学习算法来生成第二本体模式。该方法包括由处理单元(202)基于对第二本体模式的分析从第二工程程序生成第一工程程序。



1. 一种针对技术设施(106)生成工程程序的方法,所述方法包括:

由处理单元(202)接收从具有第二规范的第二工程程序生成具有第一规范的第一工程程序请求;

由处理单元(202)分析第二工程程序;

由处理单元(202)生成第一本体模式,第一本体模式包括与第二工程程序中的多个代码语句之间的多个数据流和控制流相互关系相关联的信息,其中第一本体模式是通过对第二工程程序的分析来生成的;

由处理单元(202)基于对第一本体模式的分析来生成第一本体模式的第一序列表示;

由处理单元(202)通过在第一序列表示上应用机器学习算法将第一序列表示转换成第二序列表示;

由处理单元(202)通过比较第一序列表示和第二序列表示来确定第一序列表示与第二序列表示之间的多个差异;

由处理单元(202)基于第一序列表示与第二序列表示之间的所确定的多个差异从第一本体模式生成第二本体模式;

其中第二本体模式包括与具有第一规范的第一工程程序的多个数据流和控制流相互关系相关联的信息;以及

由处理单元(202)基于对第二本体模式的分析从第二工程程序生成第一工程程序。

2. 根据权利要求1所述的方法,其中所述机器学习算法被训练成将第一本体模式的第一序列表示转换成第二本体模式的第二序列表示,并且所述机器学习算法是基于对多个工程程序的分析来训练的。

3. 根据权利要求2所述的方法,其中基于第一序列表示与第二序列表示之间的所确定的多个差异从第一本体模式生成第二本体模式包括:

由处理单元(202)基于对第一本体模式的分析将第一序列表示与第二序列表示之间的所确定的多个差异中的每一个映射到第二工程程序的多个代码语句中的一个或多个代码语句;以及

由处理单元(202)通过修改第二工程程序的所确定的一个或多个代码语句从第二工程程序生成第一工程程序。

4. 根据权利要求1的方法,其中第一工程程序是第二工程程序的优化版本。

5. 一种优化工程程序的方法,所述方法包括:

由处理单元(202)接收优化工程程序请求;

由处理单元(202)分析工程程序;

由处理单元(202)生成本体模式,所述本体模式包括与工程程序中的多个代码语句之间的多个数据流和控制流相互关系相关联的信息,其中本体模式是通过对工程程序的分析来生成的;

由处理单元(202)基于对本体模式的分析来生成本体模式的序列表示;

由处理单元(202)通过在序列表示上应用机器学习算法将序列表示转换成优化的序列表示;

由处理单元(202)通过将序列表示与优化的序列表示进行比较来确定序列表示与优化的序列表示之间的多个差异;

由处理单元(202)基于序列表示与优化的序列表示之间的所确定的多个差异从本体模式生成优化的本体模式,其中优化的本体模式包括与优化的工程程序的多个数据流和控制流相互关系相关联的信息;以及

由处理单元(202)基于对优化的本体模式的分析从工程程序生成优化的工程程序。

6.根据权利要求5所述的方法,其中所述机器学习算法被训练以将序列表示转换为优化的本体模式的优化的序列表示,并且所述机器学习算法是基于对多个未优化的工程程序和多个优化的工程程序的分析来训练的。

7.根据权利要求5至6中任一项所述的方法,其中基于序列表示与优化的序列表示之间的所确定的多个差异从本体模式生成优化的本体模式包括:

由处理单元(202)基于对本体模式的分析将所确定的多个差异中的每一个映射到工程程序的一个或多个代码语句;以及

由处理单元(202)通过修改工程程序的所确定的一个或多个代码语句来从工程程序生成优化的工程程序。

8.根据权利要求7所述的方法,进一步包括:

由处理单元(202)接收修改工程程序的多个代码语句中的代码语句的请求;

由处理单元(202)基于接收到的请求来修改工程程序的所述代码语句;以及

由处理单元(202)基于工程程序的多个代码语句中的所述代码语句的修改来训练所述机器学习算法。

9.根据权利要求5至8中任一项所述的方法,进一步包括:

由处理单元(202)生成用于优化的工程程序的仿真实例;以及

由处理单元(202)通过执行优化的工程程序从而在仿真环境中仿真由技术设施(106)的一个或多个工程对象(108A-N)对所生成的优化的工程程序的执行。

10.根据权利要求9所述的方法,进一步包括:

由处理单元(202)基于仿真执行的结果来确定优化的工程程序是否有效;

由处理单元(202)基于所生成的工程程序有效的确定将优化的工程程序实时部署到一个或多个工程对象(108A-N)上;以及

由处理单元在显示设备上显示所生成的工程程序。

11.一种用于针对技术设施(106)生成和优化工程程序的工程系统,其中所述工程系统包括:

一个或多个处理单元(202);以及

耦合到所述一个或多个处理单元的存储器(204),其中所述存储器包括以可由所述一个或多个处理单元执行的机器可读指令的形式存储的自动化模块,其中所述自动化模块(112)能够执行根据权利要求1-10中任一项所述的方法。

12.一种工业环境(100),包括:

根据权利要求11所述的工程系统(102);

包括一个或多个物理组件的技术设施(106);以及

经由网络(104)通信地耦合到所述工程系统(102)的一个或多个客户端设备(120A-N),其中所述工程系统(102)被配置成执行根据权利要求1至10中任一项所述的方法。

13.一种计算机程序产品,其中存储有机器可读指令,所述机器可读指令当由一个或多

个处理单元(202)执行时使所述处理单元执行根据权利要求1-10中任一项所述的方法。

用于针对技术设施生成和优化工程程序的方法和系统

技术领域

[0001] 本发明涉及计算机辅助程序的工程领域,并且更特别地涉及一种用于针对技术设施生成具有给定规范的工程程序的方法和系统。

背景技术

[0002] 诸如工厂之类的技术设施包括由一个或多个工程对象控制的一个或多个设备。一个或多个工程对象的示例可以包括例如可编程逻辑控制器、一个或多个现场设备、或任何其他自动化设备。一个或多个设备的示例包括但不限于控制阀、马达、泵和致动器。一个或多个工程对象通过执行可以被存储在存储器中的工程程序来控制一个或多个设备。工程程序可以包括多个机器可读指令来控制一个或多个设备。

[0004] 通常,工程程序由代码开发者通过考虑与一个或多个设备、一个或多个工程对象、以及技术设施相关联的多个参数来手动编码。多个参数的示例包括但不限于一个或多个工程对象的处理速度和存储能力。因此,工程程序被编码成使得工程程序的存储能力和处理能力需求是最小的。

[0005] 通常,代码开发者在被配置成处置图形编程块的工程系统中生成工程程序。为了生成技术设施的工程程序,代码开发者可能必须基于多个参数值为一个或多个工程对象中的每一个手动编码若干个图形程序。在一个示例中,代码开发者通过将一或多个期望图形程序块从元素库拖放到工程系统中来生成工程程序。在另一个示例中,代码开发者使用诸如计算机键盘之类的输入设备来输入基于文本的代码。之后,工程程序被编译并下载到一个或多个工程对象(诸如用于工业控制应用的可编程逻辑控制器(PLC))上。

[0006] 传统的工程系统采用函数和其他资源来向代码开发者设计和实现与技术设施相关的工程程序提供辅助。然而,这些传统的工程系统缺乏针对技术设施自动地生成具有给定规范的工程程序的自动化工程方法。工程程序的规范表示关于一个或多个约束的信息,这些约束将被施加在工程程序上以使工程程序与一个或多个工程对象兼容。例如,如果工程程序具有可能不适合工程对象的存储器空间的巨大代码长度,则工程程序可能不与工程对象兼容。在这种情况下,传统工程系统可能不会通知代码开发者所生成的工程程序可能与工程对象不兼容。

[0007] 在另一个示例中,传统的工程系统未能通过提供指导或建议来优化由代码开发者编写的代码的质量,从而提供自动化编程辅助。

[0008] 在又一个示例中,如果代码开发者必须校正错误代码,则代码开发者可能必须逐行地分析错误代码,这可能消耗大量时间。传统的工程系统未能自动确定错误代码中的一个或多个错误。因此,不存在使生成工程程序时所涉及的工程自动化的机制。

[0009] 目前,在技术设施中使用的工程程序包括大量编程块,每个编程块在代码长度方面是巨大的。代码开发者可能发现不可能手动地检查如此庞大的代码块以便预测错误并优化代码。因此,如果没有向代码开发者提供自动化辅助来优化代码、检测代码中的错误以及优化工程程序,则会浪费大量的劳动和时间。此外,代码开发者可能发现不可能及时完成代

码开发。

[0010] 此外,工程程序可以具有大量编程元素,诸如可以在工程程序的第一编程块中定义的变量、类和函数。编程元素可以在多个编程块中的其他编程块中被引用。因此,导致多个编程元素的定义中的修改的对第一编程块的任何修改都可能引起该多个编程块中的其他编程块中的多个错误。在现有系统中,在优化工程程序之前,代码开发者可能必须研究该多个编程元素的无数这种定义和引用。对于代码开发者,不可能手动地检查大量编程元素以及标识工程程序中的编程元素之间的相互关系。

[0011] 鉴于以上内容,需要一种用于针对技术设施生成具有给定规范的工程程序的高效方法和系统。

发明内容

[0012] 因此,本发明的目的是提供一种用于针对技术设施生成具有给定规范的工程程序的方法和系统。

[0013] 本发明的目的通过一种针对技术设施生成工程程序的方法来实现。该方法包括由处理单元接收从具有第二规范的第二工程程序生成具有第一规范的第一工程程序的请求。该请求包括关于第一工程程序的第一规范的信息。第一和第二规范分别包括与关联于第一和第二工程程序的编码语言、编码约定、软件配置、处理速度限制、存储器限制和关键过程指标相关联的信息。请求针对技术设施生成第一工程程序。第二工程程序被存储在技术设施的存储器存储设备中。

[0014] 技术设施是工业制造厂、工业加工厂或工业发电厂中的至少一个。技术设施包括一个或多个工程对象。一个或多个工程对象是在技术设施中一起运作以实现技术设施的一个或多个目标的设备。一个或多个工程对象的示例包括服务器、机器人、开关、自动化设备、可编程逻辑控制器(PLC)、人机接口(HMI)、马达、阀、泵、致动器、传感器、以及(一个或多个)其他工业装备。一个或多个工程对象包括处理单元和存储器。存储器被配置成存储一个或多个工程程序。在一个示例中,存储器被配置成存储第二工程程序。一个或多个工程对象的处理单元被配置成执行一个或多个工程程序以实现与一个或多个工程程序相关联的一个或多个关键性能指标。一个或多个关键性能指标的示例包括当一个或多个工程程序由一个或多个工程对象的处理单元执行时一个或多个工程程序的处理速度、存储器要求和处理效率。技术设施中的一个或多个工程对象可以经由一个或多个物理连接彼此连接。一个或多个物理连接可以包括物理链接(诸如导线或线缆)。在替代实施例中,连接也可以是虚拟链接。此外,一个或多个工程对象的功能可以基于多个参数值来定义。多个参数值包括马达配置参数、网络和通信参数、阀控制、传感器的温度或压强值、速度、扭矩等。

[0015] 在优选实施例中,该方法包括由处理单元在具有第二规范的第二工程程序上应用自然语言处理算法。自然语言处理算法的示例包括但不限于贝叶斯网络算法和条件随机场算法。自然语言处理算法被应用于第二工程程序上,以从第二工程程序中提取多个代码语句。多个代码语句可以在第二工程程序中编码为以自然语言、高级编程语言、汇编语言或编译代码编写的机器可读指令中的至少一个。多个代码语句包括与第二工程程序相关联的多个数据项和多个元数据项。多个数据项包括第二工程程序的多个变量、多个函数和多个代码语句。多个元数据项包括创建日期、作者姓名、其中部署第二工程程序的工业环境的域。

第二工程程序的多个数据项和多个元数据项中的每一个可以具有与第二工程程序的其他数据项和元数据项的数据流和控制流相互关系。

[0016] 在优选实施例中,该方法进一步包括由处理单元基于自然语言处理算法的应用来确定与第二工程程序相关联的多个数据项和多个元数据项。此外,该方法包括由处理单元生成第一本体模式(schema),第一本体模式包括与第二工程程序中的多个数据项和多个元数据项之间的多个数据流和控制流相互关系相关联的信息。第一本体模式是通过对第二工程程序的分析来生成的。第一本体模式进一步包括关于在对应于第二工程程序中的每个编程块的变量集合与第二工程程序中的多个代码语句之间的关系的的信息。该变量集合包括在第二工程程序中定义和使用的多个数据和指针变量。第一本体模式是编程域的抽象,该抽象捕获多个编程概念(诸如类层次结构、命令行、数据变量、以及编程块集合的数据结构)的语义、以及在多个编程概念和多个程序属性(如第二工程程序的类层次结构、声明的成员变量、变量的数据类型、所使用的存储器段、所使用的系统功能和资源、到其他工业域对象的链接等)之间的一个或多个关系。在一个示例中,第一本体模式被实现为知识图。在这种情况下,知识图是通过在第二工程程序上应用解析器并由此捕获在多个数据项和多个元数据项之间的多个数据流和多个控制流相互关系来获得的。

[0017] 在优选实施例中,该方法进一步包括由处理单元基于对第一本体模式的分析来生成第一本体模式的第一序列表示。为了生成第一序列表示,第一本体模式的知识图的多个节点由处理单元串行化。在一个示例中,第一序列表示是包括由第一本体模式表示的信息的自然语言句子。在另一个示例中,第一序列表示是第一本体模式的基于对象简谱(JSON)的表示。

[0018] 在优选实施例中,该方法进一步包括由处理单元训练机器学习模型,以将第一本体模式的第一序列表示转换成与第一工程程序相关联的第二本体模式的第二序列表示。第二本体模式包括与具有第一规范的第一工程程序的多个数据流和控制流相互关系相关联的信息。机器学习模型是基于对多个工程程序的分析来训练的。在一个示例中,多个工程程序可以包括工程程序的未优化版本和优化版本。在这种情况下,机器学习模型是在工程程序的未优化版本和优化版本上训练的,使得机器学习模型被训练成将工程程序的未优化版本的序列表示转换成工程程序的优化版本的序列表示。在另一个示例中,多个工程程序包括至少两个工程程序,该至少两个工程程序在功能上相同,但是在编码参数(诸如编码语言、编码长度或编码惯例)上不同。在这种情况下,机器学习模型被训练成将具有第一编码参数的工程程序的序列表示转换成具有第二编码参数的工程程序的序列表示。

[0019] 在优选实施例中,该方法进一步包括由处理单元通过在第一序列表示上应用机器学习算法将第一序列表示转换成第二序列表示。该方法进一步包括由处理单元通过比较第一序列表示和第二序列表示来确定第一序列表示与第二序列表示之间的多个差异。该方法进一步包括由处理单元基于第一序列表示与第二序列表示之间的所确定的多个差异从第一本体模式生成第二本体模式。

[0020] 该方法进一步包括由处理单元基于对第一本体模式的分析将第一序列表示与第二序列表示之间的所确定的多个差异中的每一个映射到第二工程程序的多个代码语句中的一个或多个代码语句。该方法进一步包括由处理单元通过修改第二工程程序的所确定的一个或多个代码语句从第二工程程序生成第一工程程序。该修改包括任何改变,诸如一个

或多个代码语句中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。由于第二工程程序是基于对第一本体模式的分析来修改的，因此基于对应于第二工程程序的变量集合之间的关系来修改一个或多个代码语句。

[0021] 有利地，针对技术设施生成第一工程程序所需的代码开发时间和劳动显著地减少。在一个示例中，第一工程程序是第二工程程序的优化版本。有利地，针对技术设施优化第二工程程序所需的时间和劳动显著地减少。

[0022] 本发明的目的通过一种针对技术设施优化工程程序的方法来实现。该方法包括由处理单元接收优化工程程序的请求。该请求包括关于要优化的工程程序的一个或多个参数的信息。一个或多个参数包括编码语言、编码惯例、软件配置、代码大小、存储器要求、处理能力要求、以及与工程程序相关联的关键性能指标。在一个示例中，接收到的请求是优化工程程序的代码长度。请求针对技术设施优化工程程序。工程程序被存储在技术设施的存储器存储设备中。工程程序包括多个代码语句，这些代码语句被编码为以自然语言、高级编程语言、汇编语言或编译代码编写的机器可读指令中的至少一个。

[0023] 在优选实施例中，该方法包括由处理单元在未优化的工程程序上应用自然语言处理算法。自然语言处理算法的示例包括但不限于贝叶斯网络算法和条件随机场算法。

[0024] 在优选实施例中，该方法进一步包括由处理单元基于自然语言处理算法的应用来确定与工程程序相关联的多个数据项和多个元数据项。多个数据项包括工程程序的多个变量、多个函数和多个代码语句。多个元数据项包括创建日期、作者姓名、其中部署第二工程程序的工业环境的域。第二工程程序的多个数据项和多个元数据项中的每一个可以具有与工程程序的其他数据项和元数据项的数据流和控制流相互关系。

[0025] 在优选实施例中，该方法包括由处理单元生成本体模式，该本体模式包括与工程程序中的多个代码语句之间的多个数据流和控制流相互关系相关联的信息。本体模式是通过工程程序的分析来生成的。该方法进一步包括由处理单元基于对本体模式的分析来生成本体模式的序列表示。

[0026] 该方法进一步包括由处理单元通过在序列表示上应用机器学习算法将序列表示转换成优化的序列表示。机器学习算法被训练成将序列表示转换成优化的本体模式的优化的序列表示。机器学习算法是基于对多个未优化的工程程序和多个优化的工程程序的分析来训练的。优化的本体模式包括与多个数据流和控制流相互关系相关联的信息，该多个数据流和控制流相互关系与多个数据项和元数据项相关联，该多个数据项和元数据项与工程程序的优化版本相关联。

[0027] 该方法进一步包括由处理单元通过将序列表示与优化的序列表示进行比较来确定序列表示与优化的序列表示之间的多个差异。该方法进一步包括由处理单元基于序列表示与优化的序列表示之间的所确定的多个差异从本体模式生成优化的本体模式。

[0028] 该方法进一步包括由处理单元基于对本体模式的分析将所确定的多个差异中的每一个映射到工程程序的一个或多个代码语句。该方法进一步包括由处理单元通过修改工程程序的所确定的一个或多个代码语句从工程程序生成优化的工程程序。该修改包括任何改变，诸如一个或多个代码语句中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。由于工程程序是基于对本体模式的分析来修改的，因此基于对应于工程程序的变量集合之间的关系来修改一个或多个代码语句。换句话说，基于本体模式来

优化工程程序,以针对技术设施生成优化的工程程序。有利地,优化工程程序以便与技术设施兼容所需的代码开发时间和劳动显著地减少。

[0029] 在优选实施例中,该方法进一步包括由处理单元接收修改工程程序的多个代码语句中的代码语句的请求。该请求是基于文本的用户指令、基于语音的用户指令、以及基于手势的用户指令中的至少一个。在一个示例中,诸如代码开发者之类的用户可以通过使用诸如键盘之类的数据输入设备将一个或多个编程行录入到工程程序的代码语句中来修改代码语句。该方法进一步包括由处理单元基于接收到的请求来修改工程程序的代码语句。该方法进一步包括由处理单元基于工程程序的多个代码语句中的该代码语句的修改来训练机器学习算法。

[0030] 在优选实施例中,该方法进一步包括由处理单元生成用于优化的工程程序的仿真实例。在一个示例中,仿真实例是技术设施中的一个或多个工程对象的数字孪生体。该方法进一步包括由处理单元通过在所生成的仿真实例上执行工程程序从而在仿真环境中仿真由技术设施的一个或多个工程对象对所生成的工程程序的执行。该方法进一步包括由处理单元基于所生成的第二编程块集合的仿真执行的结果来确定所生成的工程程序是有效的。此外,该方法包括由处理单元基于所生成的工程程序有效的确定将所生成的工程程序实时部署到一个或多个工程对象上。有利地,所生成的工程程序仅在确定所生成的工程程序有效之后被部署到一个或多个工程对象上。此外,该方法包括由处理单元在显示设备上显示所生成的工程程序。有利地,使得用户能够将所生成的工程程序记入文档。

[0031] 本发明的目的还通过一种用于针对技术设施生成和优化工程程序的工程系统来实现。该工程系统包括一个或多个处理单元以及耦合到处理单元的存储器。存储器包括以可由处理单元执行的机器可读指令的形式存储的自动化模块。自动化模块被配置用于执行如上所述的方法。

[0032] 本发明的目的还通过一种工业环境来实现。该工业环境包括工程系统、包括一个或多个物理组件的技术设施、以及通信地耦合到工程系统和技术设施的一个或多个客户端设备。该工程系统被配置成执行上述方法步骤。

[0033] 本发明的目的还通过一种其中存储有机器可读指令的计算机程序产品来实现,该机器可读指令在被一个或多个处理单元执行时使得一个或多个处理单元执行如上所述的方法步骤。

[0034] 现在将参照本发明的附图来论述本发明的上述特征和其他特征。所图示的实施例旨在说明而非限制本发明。

附图说明

[0035] 在下文中,参考附图中所示的所图示实施例进一步描述了本发明,在附图中:

图1是根据本发明实施例的能够针对技术设施生成和优化工程程序的工业环境的框图;

图2是其中可以实现本发明的实施例的工程系统(诸如图1所示的工程系统)的框图;

图3是其中可以实现本发明的实施例的自动化模块(诸如图2所示的自动化模块)的框图;

图4A-图4B是图示了根据本发明实施例的在工程系统中生成工程程序的示例性方法的过程流程图;以及

图5A-图5C是图示了根据本发明实施例的生成对象行为模型的示例性方法的过程流程图。

具体实施方式

[0036] 参考附图描述了各种实施例,其中相似的附图标记用于指代附图,其中相似的附图标记始终用于指代相似的元件。在以下描述中,出于解释的目的,阐述了许多具体细节,以便提供对一个或多个实施例的透彻理解。可以显然的是,这种实施例可以在没有这些具体细节的情况下实践。

[0037] 图1是根据本发明实施例的能够生成工程程序的工业环境100的框图。在图1中,工业环境100包括工程系统102、技术设施106和一个或多个客户端设备120A-N。如本文所使用的,“工业环境”指代包括可配置的物理和逻辑资源(例如网络、服务器、存储、应用、服务等)、以及分布在诸如云计算平台之类的平台上的数据的处理环境。工业环境100提供对可配置的物理和逻辑资源的共享池的按需网络访问。工程系统102经由网络104(诸如局域网(LAN)、广域网(WAN)、Wi-Fi、互联网、任何短范围或广范围通信)通信地连接到技术设施106。工程系统102还经由网络104连接到一个或多个客户端设备120A-N。

[0038] 工程系统102经由网络104连接到技术设施106中的一个或多个工程对象108A-N。一个或多个工程对象108A-N可以包括服务器、机器人、开关、自动化设备、可编程逻辑控制器(PLC)、人机接口(HMI)、马达、阀、泵、致动器、传感器、以及(一个或多个)其他工业装备。一个或多个工程对象108A-N可以经由物理连接彼此连接或连接到若干个其他组件(图1中未示出)。物理连接可以通过一个或多个工程对象108A-N之间的布线。替代地,一个或多个工程对象108A-N也可以经由非物理连接(诸如物联网(IOT))和5G网络连接。尽管图1图示了连接到一个技术设施106的工程系统102,但是本领域技术人员可以想象工程系统102可以经由网络104连接到位于不同地理位置处的若干个技术设施106。

[0039] 一个或多个客户端设备120A-N可以是台式计算机、膝上型计算机、平板电脑、智能电话等。一个或多个客户端设备120A-N中的每一个被提供有工程工具122A-N,用于分别生成和/或编辑工程程序。一个或多个客户端设备120A-N可以访问工程系统102以自动生成工程程序。一个或多个客户端设备120A-N可以访问云应用(诸如经由网络浏览器提供一个或多个工程对象108A-N的性能可视化)。遍及本说明书,术语“客户端设备”和“用户设备”可互换地使用。

[0040] 工程系统102可以是部署在控制站处的独立服务器,或者可以是云计算平台上的远程服务器。在优选实施例中,工程系统102可以是基于云的工程系统。工程系统102能够递送用于管理包括一个或多个工程对象108A-N的技术设施106的应用(诸如云应用)。工程系统102可以包括平台110(诸如云计算平台)、自动化模块112、包括硬件资源和操作系统(OS)的服务器114、网络接口116、以及数据库118。网络接口116实现在工程系统102、技术设施106和一个或多个客户端设备120A-N之间的通信。接口(诸如云接口)(图1中未示出)可以允许一个或多个客户端设备120A-N处的工程师访问存储在工程系统102处的工程项目文件,并且作为同一实例对工程项目文件执行一个或多个动作。服务器114可以包括其上安装了

OS的一个或多个服务器。服务器114可以包括一个或多个处理单元、一个或多个存储设备(诸如用于存储数据和机器可读指令(例如应用和应用编程接口(API))的存储器单元)、以及对于提供计算(诸如云计算)功能所需的其他外围设备。平台110使用服务器114的硬件资源和OS来实现诸如数据接收、数据处理、数据呈现、数据通信等功能,并且使用部署在其中的应用编程接口来递送上述服务。平台110可以包括构建在硬件和OS之上的专用硬件和软件的组合。在示例性实施例中,平台110可以对应于包括程序编辑器和编译器的集成开发环境(IDE),所述程序编辑器和编译器允许一个或多个客户端设备120A-N的用户生成工程程序。平台110可以进一步包括被配置用于生成工程程序的自动化模块112。图3中解释了自动化模块112的细节。

[0041] 数据库118存储与技术设施106和(一个或多个)客户端设备120A-N相关的信息。数据库118例如是结构化查询语言(SQL)数据存储库,或者是不仅仅SQL(NoSQL)数据存储库。在示例性实施例中,数据库118可以被配置为在工业环境100中实现的基于云的数据库,其中计算资源作为服务在平台110上递送。根据本发明的另一个实施例,数据库118是由自动化模块112可直接访问的文件系统上的位置。数据库118被配置成存储工程项目文件、工程程序、对象行为模型、与一个或多个工程对象108A-N相关联的参数值、测试结果、仿真结果、状态消息、一个或多个仿真实例、图形程序、程序逻辑、程序逻辑模式、工程对象108A-N和工程对象属性、一个或多个工程对象块、工程对象之间的关系信息、要求、程序更新消息等。

[0042] 图2是其中可以实现本发明的实施例的工程系统102(诸如图1所示的工程系统102)的框图。在图2中,工程系统102包括处理单元202、可存取存储器204、存储单元206、通信接口208、输入-输出单元210、网络接口212和总线214。

[0043] 如本文使用的处理单元202意味着任何类型的计算电路,诸如但不限于微处理器单元、微控制器、复杂指令集计算微处理器单元、精简指令集计算微处理器单元、超长指令字微处理器单元、显式并行指令计算微处理器单元、图形处理单元、数字信号处理单元、或任何其他类型的处理电路。处理单元202还可以包括嵌入式控制器,诸如通用或可编程逻辑器件或阵列、专用集成电路、单芯片计算机等。

[0044] 可存取存储器204可以是非暂时性易失性存储器和非易失性存储器。可存取存储器204可以被耦合用于与处理单元202通信,诸如作为计算机可读存储介质。处理单元202可以执行存储在可存取存储器204中的机器可读指令和/或源代码。各种机器可读指令可以存储在可存取存储器204中并从可存取存储器204访问。可存取存储器204可以包括用于存储数据和机器可读指令的任何合适的元件,诸如只读存储器、随机存取存储器、可擦除可编程只读存储器、电可擦除可编程只读存储器、硬盘驱动器、用于处置致密盘、数字视频光盘、磁盘、盒式磁带、存储卡等的可移除介质驱动器。在本实施例中,可存取存储器204包括集成开发环境(IDE) 216。IDE 216包括以机器可读指令的形式存储在上述存储介质中的任一个上的自动化模块112,并且可以与处理单元202通信并由处理单元202执行。

[0045] 当由处理单元202执行时,自动化模块112使得处理单元202在工程系统102中生成工程程序。在一实施例中,自动化模块112使得处理单元202接收从具有第二规范的第二工程程序生成具有第一规范的第一工程程序的请求。该请求包括关于第一工程程序的第一规范的信息。第一和第二规范分别包括与关联于第一和第二工程程序的编码语言、编码约定、软件配置、处理速度限制、存储器限制和关键过程指标相关联的信息。请求针对技术设施

106生成第一工程程序。第二工程程序被存储在技术设施106的存储器存储设备中。

[0046] 在一个示例中,可存取存储器204被配置成存储第二工程程序。自动化模块112进一步使得处理单元202在具有第二规范的第二工程程序上应用自然语言处理算法。自然语言处理算法的示例包括但不限于贝叶斯网络算法和条件随机场算法。将自然语言处理算法被应用于第二工程程序上,以从第二工程程序中提取多个代码语句。多个代码语句可以在第二工程程序中编码为以自然语言、高级编程语言、汇编语言或编译代码编写的机器可读指令中的至少一个。多个代码语句包括与第二工程程序相关联的多个数据项和多个元数据项。多个数据项包括第二工程程序的多个变量、多个函数和多个代码语句。多个元数据项包括创建日期、作者姓名、其中部署第二工程程序的工业环境100的域。第二工程程序的多个数据项和多个元数据项中的每一个可以具有与第二工程程序的其他数据项和元数据项的数据流和控制流相互关系。

[0047] 自动化模块112进一步使得处理单元202基于自然语言处理算法的应用来确定与第二工程程序相关联的多个数据项和多个元数据项。自动化模块112进一步使得处理单元202生成第一本体模式,该第一本体模式包括与第二工程程序中的多个数据项和多个元数据项之间的多个数据流和控制流相互关系相关联的信息。第一本体模式是通过第二工程程序的分析来生成的。第一本体模式进一步包括关于在对应于第二工程程序中的每个编程块的变量集合与第二工程程序中的多个代码语句之间的关系的信息。该变量集合包括在第二工程程序中定义和使用的多个数据和指针变量。第一本体模式是编程域的抽象,该抽象捕获多个编程概念(诸如类层次结构、命令行、数据变量、以及编程块集合的数据结构)的语义、以及在多个编程概念和多个程序属性(如第二工程程序的类层次结构、声明的成员变量、变量的数据类型、所使用的存储器段、所使用的系统功能和资源、到其他工业域对象的链接等)之间的一个或多个关系。在一个示例中,第一本体模式被实现为知识图。在这种情况下,知识图是通过在第二工程程序上应用解析器并由此捕获在多个数据项和多个元数据项之间的多个数据流和多个控制流相互关系来获得的。

[0048] 自动化模块112进一步使得处理单元202基于对第一本体模式的分析来生成第一本体模式的第一序列表示。为了生成第一序列表示,第一本体模式的知识图的多个节点由处理单元202串行化。在一个示例中,第一序列表示是包括由第一本体模式表示的信息的自然语言句子。在另一个示例中,第一序列表示是第一本体模式的基于Javascript对象简谱(JSON)的表示。

[0049] 自动化模块112进一步使得处理单元202训练机器学习模型,以将第一本体模式的第一序列表示转换成与第一工程程序相关联的第二本体模式的第二序列表示。第二本体模式包括与具有第一规范的第一工程程序的多个数据流和控制流相互关系相关联的信息。机器学习模型是基于对多个工程程序的分析来训练的。在一个示例中,多个工程程序可以包括工程程序的未优化版本和优化版本。在这种情况下,机器学习模型是在工程程序的未优化版本和优化版本上训练的,使得机器学习模型被训练成将工程程序的未优化版本的序列表示转换成工程程序和优化版本的序列表示。在另一个示例中,多个工程程序包括至少两个工程程序,该至少两个工程程序在功能上相同,但是在编码参数(诸如编码语言、编码长度或编码惯例)上不同。在这种情况下,机器学习模型被训练成将具有第一编码参数的工程程序的序列表示转换成具有第二编码参数的工程程序的序列表示。

[0050] 自动化模块112进一步使得处理单元202通过在第一序列表示上应用机器学习算法来将第一序列表示转换成第二序列表示。自动化模块112进一步使得处理单元202通过比较第一序列表示和第二序列表示来确定第一序列表示与第二序列表示之间的多个差异。自动化模块112进一步使得处理单元202基于第一序列表示与第二序列表示之间的所确定的多个差异从第一本体模式生成第二本体模式。

[0051] 自动化模块112进一步使得处理单元202基于对第一本体模式的分析将第一序列表示与第二序列表示之间的所确定的多个差异中的每一个映射到第二工程程序的多个代码语句中的一个或多个代码语句。

[0052] 自动化模块112进一步使得处理单元202通过修改第二工程程序的所确定的一个或多个代码语句从第二工程程序生成第一工程程序。该修改包括任何改变,诸如一个或多个代码语句中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。由于第二工程程序是基于对第一本体模式的分析来修改的,因此基于对应于第二工程程序的变量集合之间的关系来修改一个或多个代码语句。

[0053] 有利地,针对技术设施106生成第一工程程序所需的代码开发时间和劳动显著地减少。在一个示例中,第一工程程序是第二工程程序的优化版本。有利地,针对技术设施106优化第二工程程序所需的时间和劳动显著地减少。

[0054] 在一个示例中,自动化模块112进一步使得处理单元202接收优化工程程序的请求。该请求包括关于要优化的工程程序的一个或多个参数的信息。一个或多个参数包括编码语言、编码惯例、软件配置、代码大小、存储器要求、处理能力要求、以及与工程程序相关联的关键性能指标。在一个示例中,接收到的请求是优化工程程序的代码长度。请求针对技术设施106优化工程程序。工程程序被存储在技术设施106的存储器存储设备中。工程程序包括多个代码语句,这些代码语句被编码为以自然语言、高级编程语言、汇编语言或编译代码编写的机器可读指令中的至少一个。

[0055] 自动化模块112进一步使得处理单元202在未优化的工程程序上应用自然语言处理算法。自然语言处理算法的示例包括但不限于贝叶斯网络算法和条件随机场算法。

[0056] 自动化模块112进一步使得处理单元202基于自然语言处理算法的应用来确定与工程程序相关联的多个数据项和多个元数据项。多个数据项包括工程程序的多个变量、多个函数和多个代码语句。多个元数据项包括创建日期、作者姓名、其中部署第二工程程序的工业环境100的域。第二工程程序的多个数据项和多个元数据项中的每一个可以具有与工程程序的其他数据项和元数据项的数据流和控制流相互关系。

[0057] 自动化模块112进一步使得处理单元202生成本体模式,该本体模式包括与工程程序中的多个代码语句之间的多个数据流和控制流相互关系相关联的信息。本体模式是通过对工程程序的分析来生成的。自动化模块112进一步使得处理单元202基于对本体模式的分析来生成本体模式的序列表示。

[0058] 自动化模块112进一步使得处理单元202通过在序列表示上应用机器学习算法来将序列表示转换成优化的序列表示。机器学习算法被训练成将序列表示转换成优化的本体模式的优化的序列表示。机器学习算法是基于对多个未优化的工程程序和多个优化的工程程序的分析来训练的。优化的本体模式包括与多个数据流和控制流相互关系相关联的信息,该多个数据流和控制流相互关系与多个数据项和元数据项相关联,该多个数据项和元

数据项与工程程序的优化版本相关联。

[0059] 自动化模块112进一步使得处理单元202通过将序列表示与优化的序列表示进行比较来生成序列表示与优化的序列表示之间的多个差异。自动化模块112进一步使得处理单元202基于序列表示与优化的序列表示之间的所确定的多个差异从本体模式生成优化的本体模式。

[0060] 自动化模块112进一步使得处理单元202基于对本体模式的分析将所确定的多个差异中的每一个映射到工程程序的一个或多个代码语句。

[0061] 自动化模块112进一步使得处理单元202通过修改工程程序的所确定的一个或多个代码语句来从工程程序生成优化的工程程序。该修改包括任何改变,诸如一个或多个代码语句中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。由于工程程序是基于对本体模式的分析来修改的,因此基于对应于工程程序的变量集合之间的关系来修改一个或多个代码语句。换句话说,基于本体模式来优化工程程序,以针对技术设施106生成优化的工程程序。有利地,优化工程程序以便与技术设施106兼容所需的代码开发时间和劳动显著地减少。

[0062] 自动化模块112进一步使得处理单元202接收修改工程程序的多个代码语句中的代码语句的请求。该请求是基于文本的用户指令、基于语音的用户指令、以及基于手势的用户指令中的至少一个。在一个示例中,用户(诸如代码开发者)可以通过使用数据输入设备(诸如键盘)将一个或多个编程行录入到工程程序的代码语句中来修改代码语句。自动化模块112进一步使得处理单元202基于接收到的请求来修改工程程序的代码语句。自动化模块112进一步使得处理单元202基于工程程序的多个代码语句中的该代码语句的修改来训练机器学习算法。

[0063] 自动化模块112进一步使得处理单元202生成用于优化的工程程序的仿真实例。在一个示例中,仿真实例是技术设施106中的一个或多个工程对象108A-N的数字孪生体。

[0064] 自动化模块112进一步使得处理单元202通过在所生成的仿真实例上执行工程程序从而在仿真环境中仿真由技术设施106的一个或多个工程对象对所生成的工程程序的执行。

[0065] 自动化模块112进一步使得处理单元202基于所生成的第二编程块集合的仿真执行的结果来确定所生成的工程程序是有效的。自动化模块112进一步使得处理单元202基于所生成的工程程序有效的确定,将所生成的工程程序实时部署到一个或多个工程对象108A-N上。有利地,所生成的工程程序仅在确定所生成的工程程序有效之后被部署到一个或多个工程对象108A-N上。此外,该方法包括由处理单元202在显示设备上显示所生成的工程程序。有利地,使得用户能够将所生成的工程程序记入文档。

[0066] 存储单元206可以是配置用于存储数据库(诸如数据库118)的非暂时性存储介质,该数据库包括多个工程程序的服务器版本。

[0067] 通信接口208被配置用于在一个或多个客户端设备120A-N和工程系统102之间建立通信会话。通信接口208允许在客户端设备120A-N上运行的一个或多个工程应用将工程项目文件导入/导出到工程系统102中。在一实施例中,通信接口208与一个或多个客户端设备120A-N处的接口交互,以用于允许工程师访问与工程项目文件相关联的工程程序并且对存储在工程系统102中的工程程序执行一个或多个动作。

[0068] 输入-输出单元210可以包括能够接收一个或多个输入信号(诸如用以处理工程项目文件的用户命令)的输入设备、键盘、触敏显示器、相机(诸如接收基于手势的输入的相机)等。此外,输入-输出单元210可以是用于显示图形用户接口的显示单元,该图形用户接口可视化与经修改的工程程序相关联的行为模型,并且还显示与在图形用户接口上执行的每个动作集合相关联的状态信息。动作集合可以包括执行预定义测试、下载、编译和部署图形程序。总线214充当处理单元202、可存取存储器204和输入-输出单元210之间的互连。

[0069] 网络接口212可以被配置成处置工程系统102、客户端设备120A-N和技术设施106之间的网络连接、带宽和网络流量。

[0070] 本领域普通技术人员将领会,图2中描绘的硬件可以针对特定实现方式而变化。例如,除了所描绘的硬件之外或代替所描绘的硬件,还可以使用其他外围设备,诸如光盘驱动器等、局域网(LAN)、广域网(WAN)、无线(例如Wi-Fi)适配器、图形适配器、盘控制器、输入/输出(I/O)适配器。所描绘的示例仅出于解释的目的而提供,并且不意味着暗示关于本公开的架构限制。

[0071] 本领域技术人员将认识到,为了简单和清楚起见,本文中并没有描绘或描述适用于本公开的所有数据处理系统的完整结构和操作。取而代之,仅描绘或描述了工程系统102中对于本公开独特的或者对于理解本公开所必要的那些。工程系统102的构造和操作的其余部分可以符合现有技术中已知的各种当前实现方式和实践中的任何。

[0072] 图3是其中可以实现本发明的实施例的自动化模块112(诸如图2中所示的自动化模块112)的框图。在图3中,自动化模块112包括请求处置器模块302、对象行为模型生成模块304、分析模块306、修改器模块308、工程对象数据库310、验证模块312和部署模块314。图3是结合图1和图2来解释的。

[0073] 请求处置器模块302被配置用于接收生成与技术设施106的工业域相关联的工程程序的请求。例如,该请求是经由网络从工业环境100外部的一个或多个用户之一接收的。在替代实施例中,该请求是经由网络从一个或者一个或多个客户端设备120A-N接收的。请求处置器模块302进一步被配置成从与工业域集合相关联的多个编程块来确定与该工业域相关联的编程块集合。

[0074] 对象行为模型生成模块304被配置用于在存储器204中生成多个工程程序的本体模式。在优选实施例中,所生成的本体模式包括与对应于多个工程程序中的每个工程程序的变量集合之间的关系有关的信息。该变量集合包括在多个工程程序中定义和使用的多个数据和指针变量。在一个示例中,本体模式是多个工程程序的基于知识图的表示。

[0075] 分析模块306被配置用于分析与多个工程程序相关联的本体模式。具体地,分析模块306被配置用于检索与多个工程程序的每个编程块相关联的数据和控制参数。数据和控制参数是关键性能指标,关键性能指标包括涉及与技术设施106的工业域相关联的编程块集合的信息,例如传感器数据、致动器数据、环境数据、网络数据、任何自动化数据等。

[0076] 修改器模块308被配置用于基于对本体模式的分析的结果来修改多个工程程序的一个或多个代码语句。一个或多个代码语句是基于对本体模式的分析来修改的。该修改包包括任何改变,诸如编程块集合中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。

[0077] 工程对象数据库310被配置用于生成工程对象库,该工程对象库包括所生成的工

程程序、一个或多个工程对象108A-N、一个或多个工程对象108A-N之间的物理连接、以及与一个或多个工程对象108A-N和物理连接相关联的多个参数值。工程对象数据库310被配置用于利用工程程序的更新版本不断地更新工程对象库。此外,工程对象数据库310被配置用于在本地模式中维护工程对象库。

[0078] 验证模块312被配置成生成技术设施106的一个或多个工程对象108A-N的仿真实例。在一个示例中,仿真实例是一个或多个工程对象108A-N的数字孪生体。验证模块312被配置成通过在所生成的仿真实例上执行编程块集合从而在仿真环境中仿真由技术设施106的一个或多个工程对象108A-N对所生成的工程程序的执行。验证模块312被配置成基于所生成的第二编程块集合的仿真执行的结果来确定所生成的工程程序是有效的。此外,验证模块312被配置用于通过在所生成的仿真实例上执行编程块集合从而在仿真环境中仿真编程块集合的行为。仿真环境模拟了实际的技术设施106,诸如技术设施106。此外,仿真环境可以是实际技术设施106的虚拟设置。此外,验证模块312被配置用于基于仿真的结果来验证编程块集合的行为。仿真的结果可以指示工程程序或编程块集合(如果部署在技术设施106中的话)的成功或失败。

[0079] 部署模块314被配置用于基于该验证将工程程序实时部署到安装在技术设施106中的一个或多个工程对象108A-N上。有利地,所生成的工程程序仅在确定所生成的工程程序有效之后被部署到一个或多个工程对象108A-N上。

[0080] 图4A-图4B是图示了根据本发明实施例的在工程系统102中生成工程程序的示例性方法400的过程流程图。

[0081] 在步骤402,由处理单元202接收请求。该请求是从具有第二规范的第二工程程序生成具有第一规范的第一工程程序。该请求包括关于第一工程程序的第一规范的信息。第一和第二规范分别包括与关联于第一和第二工程程序的编码语言、编码约定、软件配置、处理速度限制、存储器限制和关键过程指标相关联的信息。请求针对技术设施106生成第一工程程序。第二工程程序被存储在技术设施106的存储器存储设备中。在一个示例中,存储器204被配置成存储第二工程程序。

[0082] 在步骤404,由处理单元202通过在具有第二规范的第二工程程序上应用自然语言处理算法来分析第二工程程序。自然语言处理算法的示例包括但不限于贝叶斯网络算法和条件随机场算法。自然语言处理算法被应用于第二工程程序上,以从第二工程程序中提取多个代码语句。多个代码语句可以在第二工程程序中编码为以自然语言、高级编程语言、汇编语言或编译代码编写的机器可读指令中的至少一个。多个代码语句包括与第二工程程序相关联的多个数据项和多个元数据项。多个数据项包括第二工程程序的多个变量、多个函数和多个代码语句。多个元数据项包括创建日期、作者姓名、其中部署第二工程程序的工业环境的域。第二工程程序的多个数据项和多个元数据项中的每一个可以具有与第二工程程序的其他数据项和元数据项的数据流和控制流相互关系。

[0083] 在步骤406,由处理单元202基于自然语言处理算法的应用来确定与第二工程程序相关联的多个数据项和多个元数据项。

[0084] 在步骤408,由处理单元202生成第一本体模式,该第一本体模式包括与第二工程程序中的多个数据项和多个元数据项之间的多个数据流和控制流相互关系相关联的信息。第一本体模式是通过对第二工程程序的分析来生成的。第一本体模式进一步包括关于在对

应于第二工程程序中的每个编程块的变量集合与第二工程程序中的多个代码语句之间的关系的信息。该变量集合包括在第二工程程序中定义和使用的多个数据和指针变量。第一本体模式是编程域的抽象,该抽象捕获多个编程概念(诸如类层次结构、命令行、数据变量、以及编程块集合的数据结构)的语义、以及在多个编程概念和多个程序属性(如第二工程程序的类层次结构、声明的成员变量、变量的数据类型、所使用的存储器段、所使用的系统功能和资源、到其他工业域对象的链接等)之间的一个或多个关系。在一个示例中,第一本体模式被实现为知识图。在这种情况下,知识图是通过在第二工程程序上应用解析器并由此捕获在多个数据项和多个元数据项之间的多个数据流和多个控制流相互关系来获得的。

[0085] 在步骤410,由处理单元202基于对第一本体模式的分析来生成第一本体模式的第一序列表示。为了生成第一序列表示,第一本体模式的知识图的多个节点由处理单元202串行化。在一个示例中,第一序列表示是包括由第一本体模式表示的信息的自然语言句子。在另一个示例中,第一序列表示是第一本体模式的基于Javascript对象简谱(JSON)的表示。

[0086] 在步骤412,由处理单元202训练机器学习模型,以将第一本体模式的第一序列表示转换成与第一工程程序相关联的第二本体模式的第二序列表示。第二本体模式包括与具有第一规范的第一工程程序的多个数据流和控制流相互关系相关联的信息。机器学习模型是基于对多个工程程序的分析来训练的。在一个示例中,多个工程程序可以包括工程程序的未优化版本和优化版本。在这种情况下,机器学习模型是在工程程序的未优化版本和优化版本上训练的,使得机器学习模型被训练成将工程程序的未优化版本的序列表示转换成工程程序的优化版本的序列表示。在另一个示例中,多个工程程序包括至少两个工程程序,该至少两个工程程序在功能上相同,但是在编码参数(诸如编码语言、编码长度或编码惯例)上不同。在这种情况下,机器学习模型被训练成将具有第一编码参数的工程程序的序列表示转换成具有第二编码参数的工程程序的序列表示。

[0087] 在步骤414,由处理单元202通过在第一序列表示上应用机器学习算法将第一序列表示转换成第二序列表示。

[0088] 在步骤416,由处理单元202通过比较第一序列表示和第二序列表示来确定第一序列表示与第二序列表示之间的多个差异。

[0089] 在步骤418,由处理单元202基于第一序列表示与第二序列表示之间的所确定的多个差异从第一本体模式生成第二本体模式。

[0090] 在步骤420,由处理单元202基于对第一本体模式的分析将第一序列表示与第二序列表示之间的所确定的多个差异中的每一个映射到第二工程程序的多个代码语句中的一个或多个代码语句。

[0091] 在步骤422,由处理单元202通过修改第二工程程序的所确定的一个或多个代码语句从第二工程程序生成第一工程程序。该修改包括任何改变,诸如一个或多个代码语句中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。由于第二工程程序是基于对第一本体模式的分析来修改的,因此基于对应于第二工程程序的变量集合之间的关系来修改一个或多个代码语句。

[0092] 有利地,针对技术设施106生成第一工程程序所需的代码开发时间和劳动显著地减少。在一个示例中,第一工程程序是第二工程程序的优化版本。有利地,针对技术设施106优化第二工程程序所需的时间和劳动显著地减少。

[0093] 图5A-图5C是图示了根据本发明实施例的在工程系统102中优化工程程序的示例性方法500的过程流程图。

[0094] 在步骤502,由处理单元202接收优化工程程序的请求。该请求包括关于要优化的工程程序的一个或多个参数的信息。一个或多个参数包括编码语言、编码惯例、软件配置、代码大小、存储器要求、处理能力要求、以及与工程程序相关联的关键性能指标。在一个示例中,接收到的请求是优化工程程序的代码长度。请求针对技术设施106优化工程程序。工程程序被存储在技术设施106的存储器存储设备中。工程程序包括多个代码语句,这些代码语句被编码为以自然语言、高级编程语言、汇编语言或编译代码编写的机器可读指令中的至少一个。

[0095] 在步骤504,由处理单元202通过未优化的工程程序上应用自然语言处理算法来分析工程程序。自然语言处理算法的示例包括但不限于贝叶斯网络算法和条件随机场算法。

[0096] 在步骤506,由处理单元202基于自然语言处理算法的应用来确定与工程程序相关联的多个数据项和多个元数据项。多个数据项包括工程程序的多个变量、多个函数和多个代码语句。多个元数据项包括创建日期、作者姓名、其中部署第二工程程序的工业环境的域。第二工程程序的多个数据项和多个元数据项中的每一个可以具有与工程程序的其他数据项和元数据项的数据流和控制流相互关系。

[0097] 在步骤508,由处理单元202确定本体模式。本体模式包括与工程程序中的多个代码语句之间的多个数据流和控制流相互关系相关联的信息。本体模式是通过对工程程序的分析来生成的。自动化模块112进一步使得处理单元202基于对本体模式的分析来生成本体模式的序列表示。

[0098] 在步骤510,由处理单元202通过在序列表示上应用机器学习算法来生成序列表示到优化的序列表示。机器学习算法被训练成将序列表示转换成优化的本体模式的优化的序列表示。机器学习算法是基于对多个未优化的工程程序和多个优化的工程程序的分析来训练的。优化的本体模式包括与多个数据流和控制流相互关系相关联的信息,该多个数据流和控制流相互关系与多个数据项和元数据项相关联,该多个数据项和元数据项与工程程序的优化版本相关联。

[0099] 在步骤512,由处理单元202通过将序列表示与优化的序列表示进行比较来生成序列表示与优化的序列表示之间的多个差异。

[0100] 在步骤514,由处理单元202基于序列表示与优化的序列表示之间的所确定的多个差异从本体模式生成优化的本体模式。

[0101] 在步骤516,由处理单元202基于对本体模式的分析将所确定的多个差异中的每一个映射到工程程序的一个或多个代码语句。

[0102] 在步骤518,由处理单元202通过修改工程程序的所确定的一个或多个代码语句从工程程序生成优化的工程程序。该修改包括任何改变,诸如一个或多个代码语句中的一个或多个变量、代码行、类、函数或注释的添加、删除、更新、替换或修订。由于工程程序是基于对本体模式的分析来修改的,因此基于对应于工程程序的变量集合之间的关系来修改一个或多个代码语句。换句话说,基于本体模式来优化工程程序,以针对技术设施106生成优化的工程程序。有利地,优化工程程序以便与技术设施106兼容所需的代码开发时间和劳动显

著地减少。

[0103] 在步骤520,由处理单元202接收修改工程程序的多个代码语句中的代码语句的请求。该请求是基于文本的用户指令、基于语音的用户指令、以及基于手势的用户指令中的至少一个。在一个示例中,诸如代码开发者之类的用户可以通过使用诸如键盘之类的数据输入设备将一个或多个编程行录入到工程程序的代码语句中来修改代码语句。自动化模块112进一步使得处理单元202基于接收到的请求来修改工程程序的代码语句。

[0104] 在步骤522,由处理单元202基于工程程序的多个代码语句中的该代码语句的修改来训练机器学习算法。

[0105] 在步骤524,由处理单元202生成用于优化的工程程序的仿真实例。在一个示例中,仿真实例是技术设施106中的一个或多个工程对象108A-N的数字孪生体。

[0106] 在步骤526,由处理单元202通过在所生成的仿真实例上执行工程程序从而在仿真环境中仿真由技术设施106的一个或多个工程对象对所生成的工程程序的执行。

[0107] 在步骤528,由处理单元202基于所生成的第二编程块集合的仿真执行的结果来确定所生成的工程程序是有效的。

[0108] 在步骤530,由处理单元202基于所生成的工程程序有效的确定将所生成的工程程序实时部署到一个或多个工程对象上。有利地,所生成的工程程序仅在确定所生成的工程程序有效之后被部署到一个或多个工程对象上。此外,该方法包括由处理单元在显示设备上显示所生成的工程程序。有利地,使得用户能够将所生成的工程程序记入文档。

[0109] 本发明可以采取计算机程序产品的形式,该计算机程序产品包括可从存储程序代码的计算机可用或计算机可读介质来访问的程序模块,该程序代码用于由一个或多个计算机、处理单元或指令执行系统使用或与其结合地使用。出于本描述的目的,计算机可用或计算机可读介质可以是能够包含、存储、传送、传播或传输用于由指令执行系统、装置或设备使用或与其结合地使用的程序的任何装置。该介质可以是电子的、磁的、光学的、电磁的、红外的或半导体系统(或装置或设备)、或者它们中的传播介质以及其本身的传播介质,因为信号载体不被包括在物理计算机可读介质的定义中,该物理计算机可读介质包括半导体或固态存储器、磁带、可移除计算机磁盘、随机存取存储器(RAM)、只读存储器(ROM)、刚性磁盘和光盘,诸如压缩盘只读存储器(CD-ROM)、压缩盘读/写、以及DVD。如本领域技术人员所已知,用于实现该技术的每个方面的处理单元和程序代码两者都可以是集中式的或分布式的(或其组合)。

[0110] 虽然已经参考某些实施例详细描述了本发明,但是应当领会,本发明不限于这些实施例。鉴于本公开,对于本领域的技术人员来说,在不脱离如本文中描述的本发明的各种实施例的范围的情况下,许多修改和变型本身将会存在。因此,本发明的范围由以下权利要求而不是前述描述来指示。在权利要求的等同意义和范围内的所有改变、修改和变型都被认为是在权利要求的范围内。在方法权利要求中要求保护的所有有利实施例也可以应用于系统/装置权利要求。

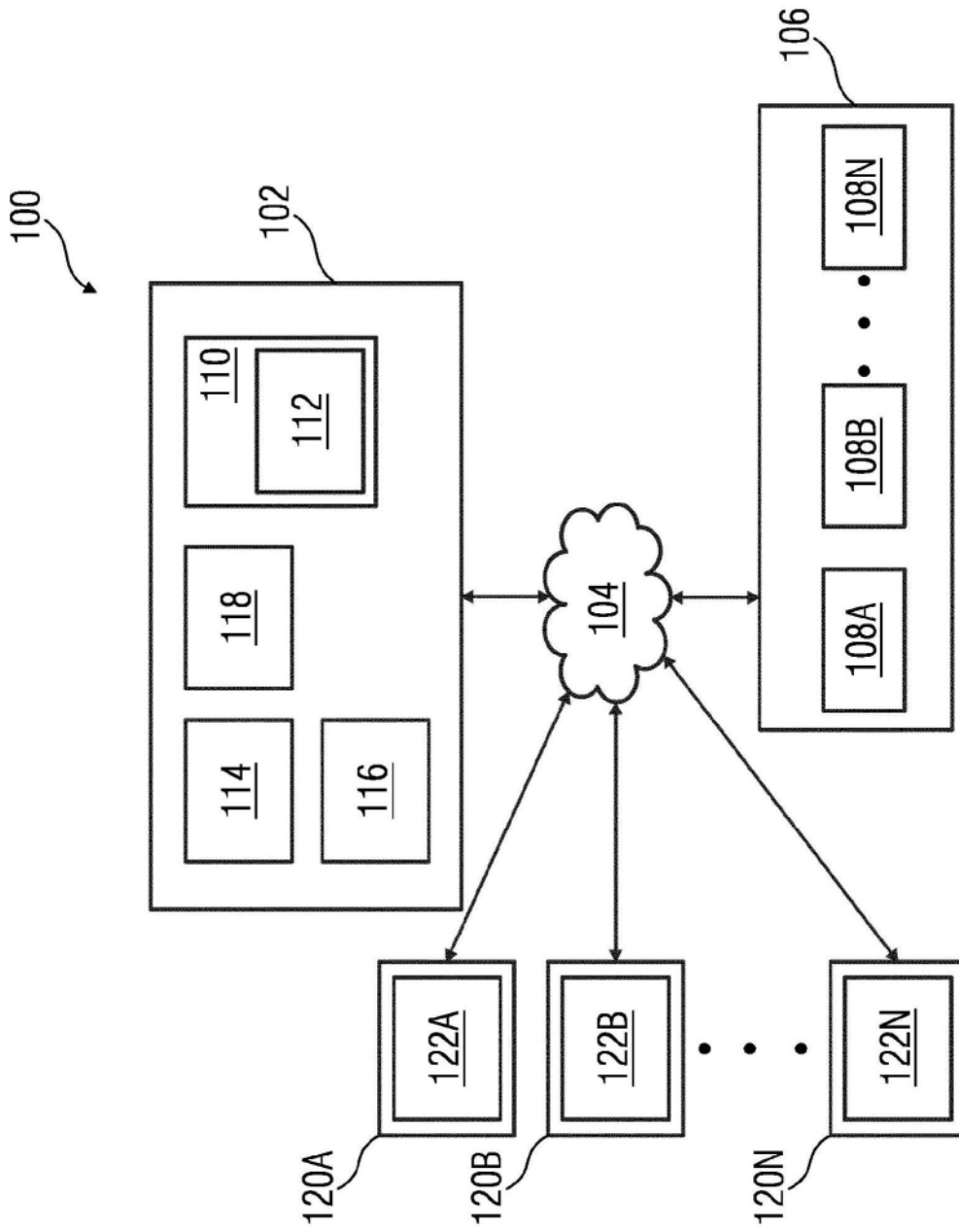


图1

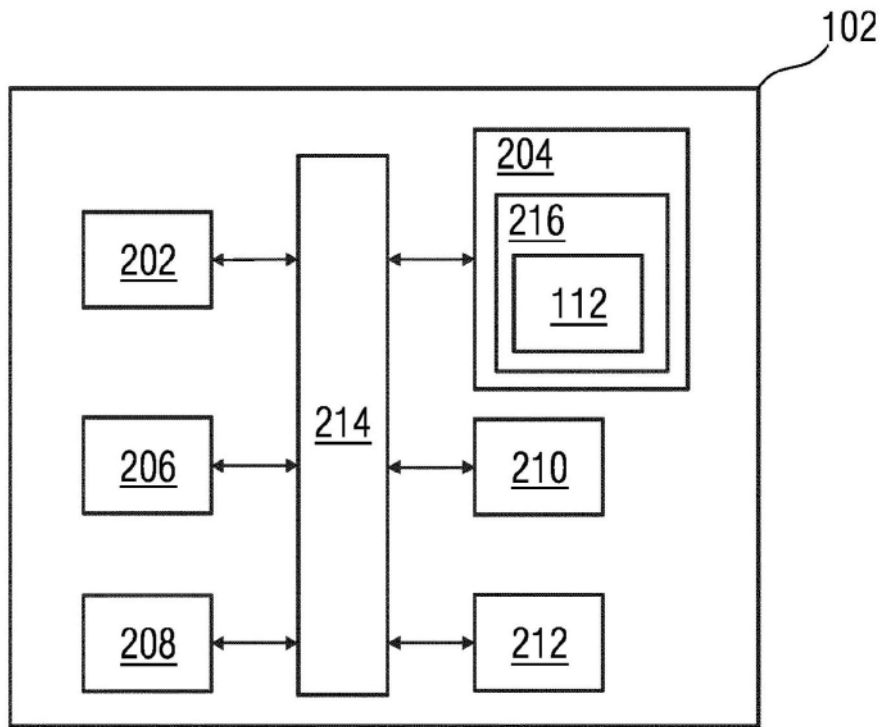


图2

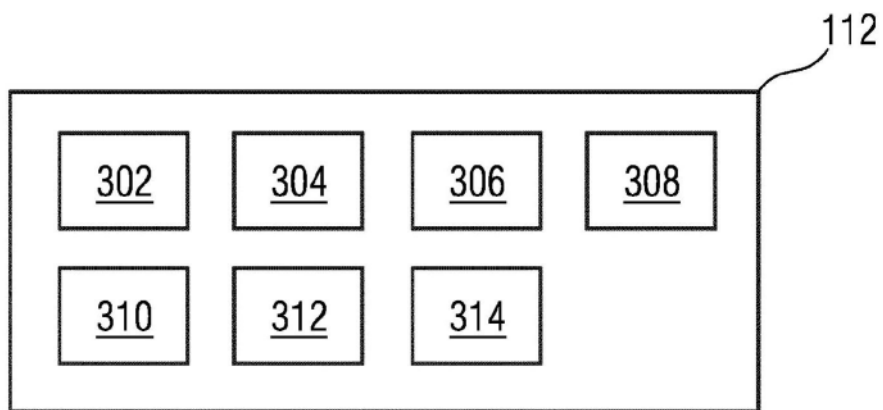


图3

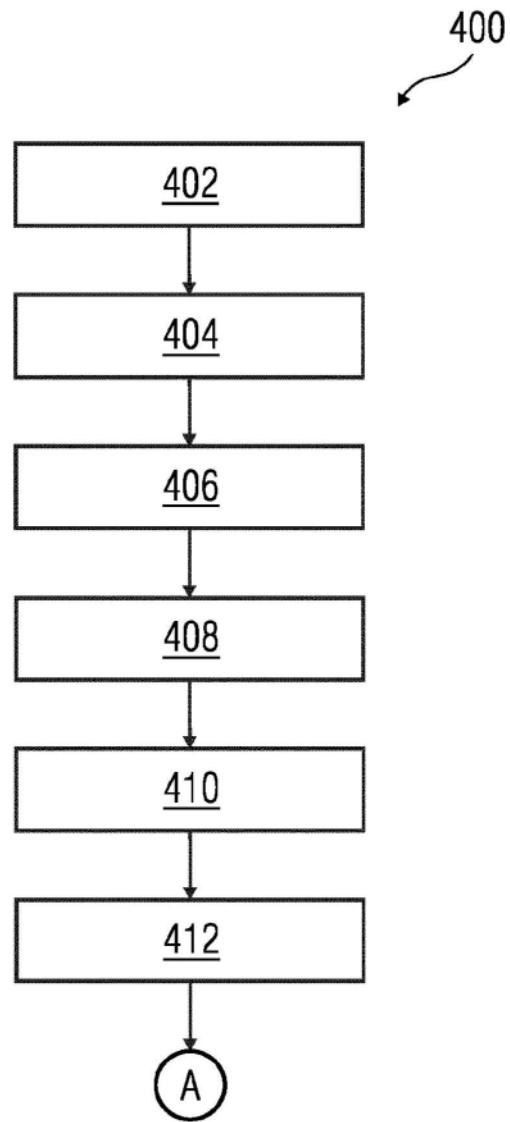


图4A

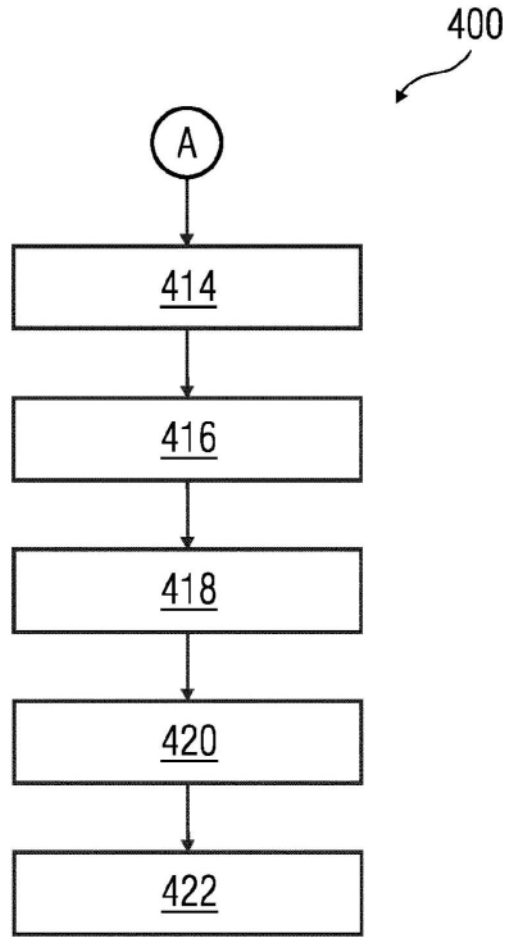


图4B

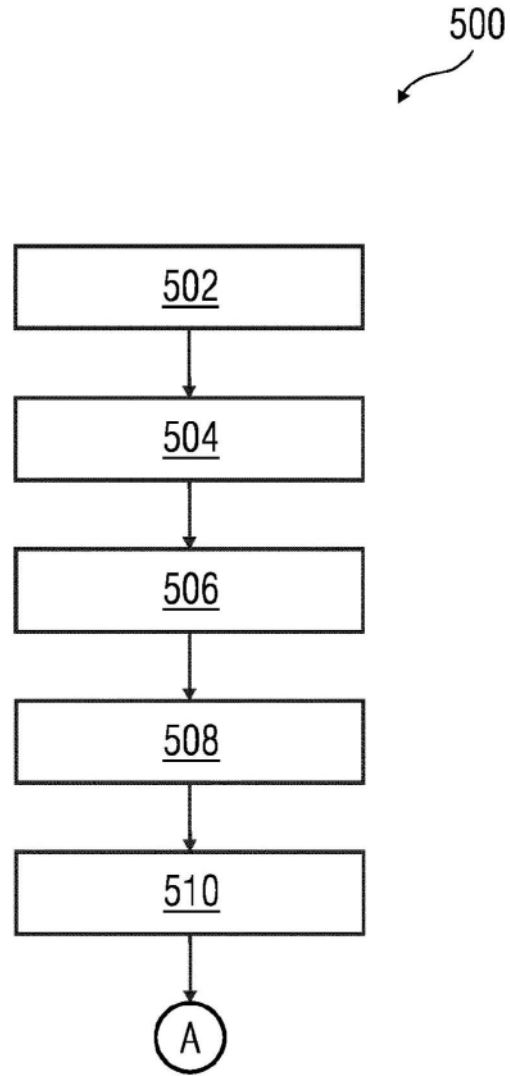


图5A

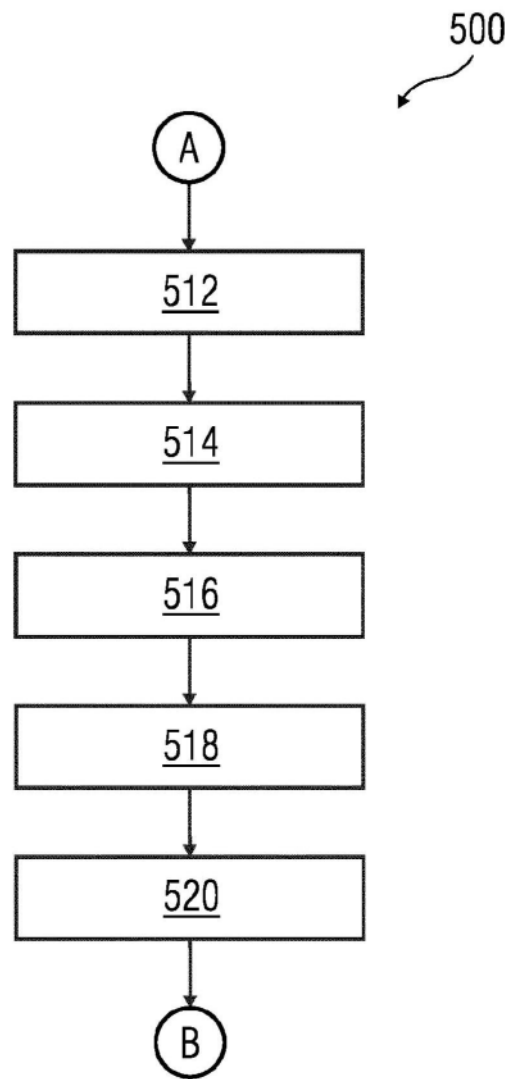


图5B

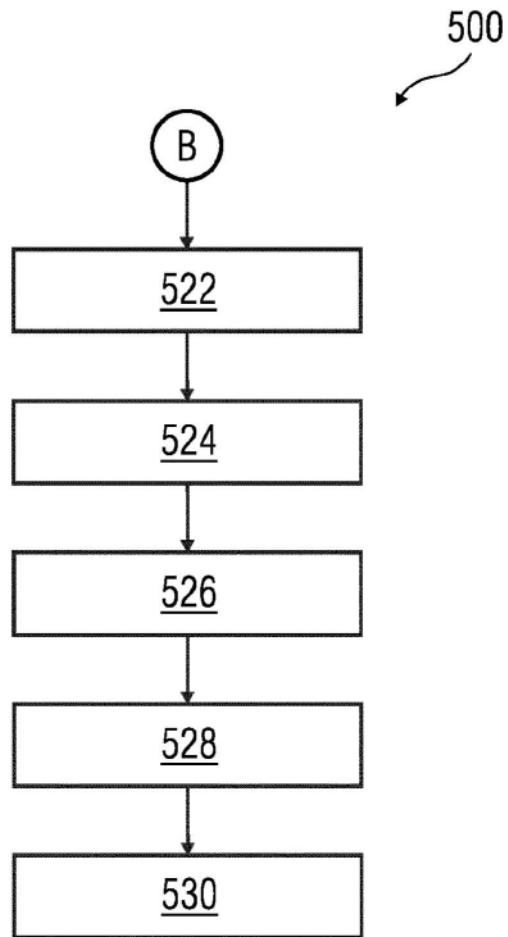


图5C