



(12) 发明专利

(10) 授权公告号 CN 101401070 B

(45) 授权公告日 2012. 07. 04

(21) 申请号 200780007478. 5

G06F 15/163 (2006. 01)

(22) 申请日 2007. 02. 28

G05B 19/04 (2006. 01)

(30) 优先权数据

0600448-5 2006. 03. 02 SE

(56) 对比文件

US 6950863 B1, 2005. 09. 27,

US 2002/0116453 A1, 2002. 08. 22, 全文.

(85) PCT申请进入国家阶段日

2008. 09. 02

审查员 王晶

(86) PCT申请的申请数据

PCT/SE2007/000188 2007. 02. 28

(87) PCT申请的公布数据

W02007/100291 EN 2007. 09. 07

(73) 专利权人 ABB 股份有限公司

地址 瑞典韦斯特罗斯

(72) 发明人 S·萨尔伯格

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 王岳 王忠忠

(51) Int. Cl.

G06F 9/46 (2006. 01)

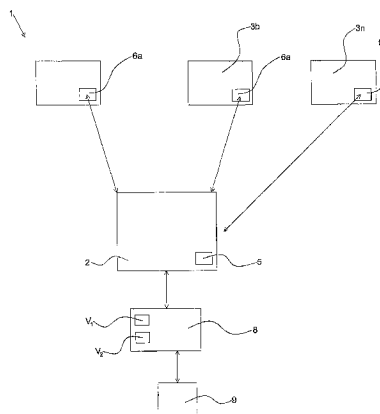
权利要求书 2 页 说明书 6 页 附图 2 页

(54) 发明名称

用于管理不同版本应用程序的方法以及自动化系统

(57) 摘要

本发明涉及一种在自动化系统中用于管理至少两个版本的应用程序的方法。该系统包括一个与通信服务器 2 双向连接的控制器 8, 其通过存储其上的应用程序进行处理过程的自动化。该通信服务器 2 按顺序与至少两个客户机双向连接。通过该方法改善了通信服务器与多台客户机之间的通信。当使用一个新版本的应用程序时, 不需要重新配置就能显示图形和图像。本发明也涉及这样一种系统。



1. 一种用于在自动化系统中管理不同版本的应用程序的方法,所述系统包括通过存储于其中的应用程序进行处理过程自动化的控制器(8),所述控制器(8)与通信服务器(2)双向连接,并且所述通信服务器(2)按顺序与至少两个客户机(3a, 3b, ..., 3n)双向连接,其特征在于包括以下步骤:

在所述控制器(8)中执行至少两个版本( $V_1, V_2, \dots, V_n$ )的所述应用程序,

使用所述至少两个版本( $V_1, V_2, \dots, V_n$ )中相应版本的所述至少两个客户机(3a, 3b, ..., 3n)中的至少一个,请求这些版本所共有的并与所述处理过程相关联的变量的值,所述值采用相同的变量名称进行请求,

所述通信服务器(2)为所请求的变量检索相应的值,并将所述值传送到相应的客户机(3a, 3b, ..., 3n),

其中每个所述客户机(3a, 3b, ..., 3n)通过定义它所处的环境向所述通信服务器(2)指定它所使用的版本( $V_1, V_2, \dots, V_n$ ),

其中所述通信服务器(2)基于各客户机(3a, 3b, ..., 3n)所给出的指定,将每个客户机(3a, 3b, ..., 3n)与相应的版本( $V_1, V_2, \dots, V_n$ )相链接。

2. 如权利要求1所要求的方法,其中所述通信服务器(2)是OPC服务器,并且所述客户机是OPC客户机。

3. 如权利要求2所要求的方法,其中每个OPC客户机(3a, 3b, ..., 3n)都具有相应的OPC处理机(6a, 6b, ..., 6n),所述OPC处理机(6a, 6b, ..., 6n)包含着关于其客户机(3a, 3b, ..., 3n)工作环境的信息。

4. 如权利要求3所要求的方法,其中所述OPC处理机(6a, 6b, ..., 6n)将数据运送到指定各客户机(3a, 3b, ..., 3n)使用环境的所述OPC服务器(2)。

5. 如前述任一项权利要求1-4所要求的方法,其中所述控制器(8)在没有控制处理过程的版本被命令去控制所述处理过程时,向所述通信服务器(2)发出信号。

6. 如前述任一项权利要求1-4所要求的方法,其中所述通信服务器(2)为所请求的变量检索相应的值的步骤,包括以下步骤:

分配对应于所述至少两个版本( $V_1, V_2, \dots, V_n$ )中的每一个的唯一的标识符( $ID_1, ID_2, \dots, ID_n$ ),以及

在所述控制器(8)和所述通信服务器(2)之间发送的通信消息的通信包中包含所述唯一的标识符( $ID_1, ID_2, \dots, ID_n$ )。

7. 如权利要求6所要求的方法,其中所述通信服务器(2)使用所述唯一的标识符( $ID_1, ID_2, \dots, ID_n$ )从控制器(8)中的一个版本的应用程序请求数据。

8. 一种自动化系统,包括通过存储于其上的应用程序进行处理过程自动化的控制器(8),所述控制器(8)与通信服务器(2)双向连接,并且所述通信服务器(2)按顺序与至少两个客户机(3a, 3b, ..., 3n)双向连接,其特征在于该系统包括:

在所述控制器(8)中具有至少两个版本( $V_1, V_2, \dots, V_n$ )的所述应用程序,

在使用所述至少两个版本( $V_1, V_2, \dots, V_n$ )的相应版本的所述至少两个客户机(3a, 3b, ..., 3n)中,请求这些版本所共有的并与所述处理过程相关联的变量的值的装置,所述值采用相同的变量名称进行请求,

在所述通信服务器(2)中为所请求的变量检索相应的值的装置,以及将所述值传送到

相应客户机 (3a, 3b, ..., 3n) 的装置,

其中每个所述客户机 (3a, 3b, ..., 3n) 包括通过定义它们所处的环境向所述通信服务器 (2) 指定它所使用的版本 ( $V_1, V_2, \dots, V_n$ ) 的装置,

其中所述通信服务器 (2) 包括基于各个客户机 (3a, 3b, ..., 3n) 所给出的指定, 将每个客户机 (3a, 3b, ..., 3n) 与相应的版本 ( $V_1, V_2, \dots, V_n$ ) 相链接的装置。

9. 如权利要求 8 所要求的系统, 其中所述通信服务器 (2) 是 OPC 服务器, 并且所述客户机是 OPC 客户机。

10. 如权利要求 9 所要求的系统, 其中每个 OPC 客户机 (3a, 3b, ..., 3n) 都具有相应的 OPC 处理机 (6a, 6b, ..., 6n), 所述 OPC 处理机 (6a, 6b, ..., 6n) 包含着关于其客户机 (3a, 3b, ..., 3n) 工作环境的信息。

11. 如权利要求 9 所要求的系统, 其中所述 OPC 处理机 (6a, 6b, ..., 6n) 被设置为将数据运送到指定各客户机 (3a, 3b, ..., 3n) 使用环境的所述 OPC 服务器 (2)。

12. 如权利要求 8-11 任一项所要求的系统, 其中所述控制器 (8) 被设置为在没有控制处理过程的版本被命令去控制所述处理过程时, 向所述通信服务器 (2) 发出信号。

13. 如权利要求 8-11 任一项所要求的系统, 其中所述通信服务器 (2) 还包括:

存储装置, 用于存储与所述至少两个版本 ( $V_1, V_2, \dots, V_n$ ) 中的相应版本相关联的唯一的标识符 ( $ID_1, ID_2, \dots, ID_n$ ), 以及

用于在所述控制器 (8) 和所述通信服务器 (2) 之间发送的通信消息的通信包中包含所述唯一的标识符 ( $ID_1, ID_2, \dots, ID_n$ ) 的装置。

## 用于管理不同版本应用程序的方法以及自动化系统

### 技术领域

[0001] 本发明涉及自动化领域,特别是涉及一种如权利要求 1 的前序部分中所定义的用于管理不同版本的应用程序的方法,以及一种如在权利要求 11 的前序部分中所定义的自动系统。

### 背景技术

[0002] 自动控制或自动化在工业生产中是非常重要的,例如用于取消生产线中单调乏味的工作以及减少危险因素。通常,自动化还提高了效率以及生产的质量,并且对于满足目前对不受周围环境影响和经济化生产的需求也是一种极好的手段。

[0003] 当前,多数加工工业,例如食品生产、能量或电力生产工厂、制药和化学工业都已经或多或少地实现了自动化,也就是说,具有自动的、计算化的控制、操作和管理。为了满足不断变化的需求,需要不断地开发在这些计算化的机器和生产过程中使用的程序代码。因此,为改善现存的例行程序以及克服所使用的例行程序的缺陷,有必要对系统软件进行有规律的更新。

[0004] 当要执行这种更新时,当前所使用的版本就要被新的、更新后的版本所覆盖。这样就会带来严重的问题,如果新的版本包含设计缺陷或如果它不能发挥作用或如果它包含了无法预见的系统的不兼容性,例如导致了系统内部的不稳定性,那么就不应当完全下载该新的版本。如果需要恢复到旧的版本,之前的代码就不得不再次进行下载。生产就有可能暂停,直至准备好旧的版本并再次运行。这种生产中的中断是非常昂贵的,甚至取决于所讨论的这些工业的类型,这种中断还可能是危险的。

[0005] 例如当在控制器(该控制器例如是过程控制器或者可编程逻辑控制器(PLC))中执行应用程序并更新这种应用程序时,该控制器是多数自动化系统的一个至关重要的部分,就会出现上述问题。在一个真实环境中预见新版本的控制应用程序是否能够发挥作用是困难的,如果它不能发挥作用几乎不可能知道该应用程序的哪个部分不能令人满意的运行。

[0006] 从自动化处理过程中获得的结果通常显示给用户或操作员,并且能够跟踪处理过程的进展对用户来说是重要的。当改变版本时,能够将结果显示给用户也是重要的,更新应用程序导致的旧版本的应用程序被删除或覆盖所带来的问题是需要花费时间对该应用程序的不同部分进行重新配置。因此当旧版本的应用程序被删除或覆盖时,例如与该应用程序相关联的图形就是如此。这样在不同版本之间改变时就存在一段时间,例如在这段时间中所显示的图形不与当前版本相关联,而是与之前的版本相关联。取决于所讨论的自动化系统的不同,这个时间段可能只是一个不必要的干扰,但它也可能是危险的并能导致严重的后果。当需要在不同版本之间来回切换时,这种缺陷就会变得更加糟糕,这是因为每次转变都要重复一次这种问题。

### 发明内容

[0007] 一种改进的为更新应用程序的版本而执行的方法,是按顺序或按类似并行的方式执行两个版本,如在共同未决的专利申请“用于估计的方法、自动化系统和控制器”中所描述的,该申请已转让给本发明的申请人,并在提交本发明的同一天进行了申请。

[0008] 目前,当下载了新版本的应用程序并因此删除之前的版本时,在控制器和使用了之前的版本的用户工作站或客户机之间不进行通信,这是因为客户机会试图从已经不存在的之前的版本中检索数据。为了解决这一问题,当执行两个版本的应用程序时,出现了困难,即在同一控制器中执行这些版本时需要知道要从哪个版本中检索数据。

[0009] 本发明的目的就是要提供一种方法,以确保在单一控制器中执行两个或多个版本的应用程序时,能够从正确的版本的应用程序中检索数据。

[0010] 其中,该目的将通过权利要求 1 所要求的方法和权利要求 11 所要求的自动化系统来实现。

[0011] 根据本发明,在自动化系统中提供一种用于管理至少两个版本的应用程序的方法。该系统包括控制器,例如过程控制器或可编程逻辑控制器,用以通过存储于其上的应用程序进行处理过程的自动化。该控制器优选地与通信服务器双向通信,并且该通信服务器按顺序与至少两个客户机双向连接。根据该方法,在控制器中执行至少两个版本的应用程序。使用相应版本的客户机请求获得与该处理过程相关联的某个变量的值。所请求的变量在所有的版本中都以相同的变量名称进行请求。最后,通信服务器为所请求的变量检索相应的值,并将这些值传送到相应客户机中。通过本发明的方法,提供了一种对向/从存储在控制器中几个版本的应用程序进行通信的非常便利的方法,更特别地,能够与每个版本独立地进行通信,从而尽管是以相同的变量名称来请求数据,控制器也总能知晓哪个数据要检索,也就是与所时论的版本相关联的数据。例如,当执行新版本时,可以立即看到与应用程序相应的并描述了过程变量的值的图形,而不需要对其重新进行配置。当新版本开始控制该自动化系统内的处理过程时,所显示的图像或图形立即进行更新。这样,本发明保证了与所有版本的应用程序持续有效地进行通信。进而,能够以简单的方式执行这种通信。

[0012] 根据本发明的一个实施例,每个客户机向通信服务器指定它所使用的版本。这种指定可以采用任意适当的方式来实现,例如通过定义它们所处的环境(诸如:生产环境、估计环境、之前的环境)。

[0013] 根据本发明的一个实施例,通信服务器基于各客户机所给出的指定,将每个客户机与相应版本相链接。这种链接能够以任意想要的方式来实现,并且尽管是以相同的变量名称来请求数据的,也能够使该通信服务器向各客户机检索正确的数据。

[0014] 根据本发明的一个实施例,通信服务器,例如通过使用唯一的地址,从控制器中的特定版本的应用程序中请求数据。根据本发明的进一步实施例,以客户机的名义来请求数据。这可以使通信服务器以一种可靠的方式知晓要从哪个版本中检索数据,并因此能够传送正确的数据给客户机,而不用与客户机进行任何交互,这样,就为用户访问不同版本的应用程序提供了非常便利的方法。

[0015] 根据本发明的另一个实施例,当没有控制处理过程的版本被命令去控制该处理过程时,控制器就向该通信服务器发出信号。从而通知该通信服务器哪个版本控制处理过程,并且能够容易地对客户机所请求的数据进行重定向。也就是,监督该处理过程的客户机甚至在版本改变之后,仍能从控制该处理过程的版本中接收数据。

[0016] 根据本发明的又一个实施例,该通信服务器是 OPC 服务器。这是一种在自动化系统中普遍使用的公知服务器,由此本发明的方法可以在现存并广泛使用的系统中实现。

[0017] 根据本发明的其它实施例,为每个客户机提供 OPC 处理机。OPC 处理机将数据运送到指定各客户机执行哪个版本的 OPC 服务器上,使用这种 OPC 处理机的用户能够从正确的版本中检索正确的数据。

[0018] 本发明还涉及这样一种系统,由此获导与上述相似的优点。

#### 附图说明

[0019] 图 1 示出了实现本发明的自动化系统的概括图。

[0020] 图 2 示出了按照本发明的包含在方法中的步骤的流程图。

#### 具体实施方式

[0021] 在使用自动化系统的典型工厂中,控制系统用于监视并控制设备的参数,例如容器的高度、生产过程中液体的温度或者阀门的打开。在多数这种工厂中,存在着一个集中的控制区域,典型地,用户可以使用几个显示不同环境的操作站。每个操作站都具有图形用户接口,操作员可以通过现实设备的图形表现来监督处理过程。

[0022] 一种对更新控制处理过程的应用程序的版本进行改善的方法是在同一控制器中执行两个版本的应用程序,如在共同的未决专利申请“用于估计的方法、自动化系统和控制器”中所描述的,该申请已转让给本发明的申请人,并在此结合以作为参考。简而言之,按照所参考的共同的未决专利申请,描述了一种用于在自动化系统的控制处理过程的控制器中估计应用程序的方法。该控制器具有存储于其中的两个或多个版本的应用程序,实际的处理参数输入到所有不同的版本中,但只有来自当前正在控制处理过程的应用程序的结果才作为输出使用,用以控制该处理过程。因此,在启动新版本的应用程序之前,用户能够保证要安装的新版本的应用程序在实际的处理条件下是起作用的。

[0023] 当两个或多个版本的应用程序在控制器中执行时,该控制器例如是可编程逻辑控制器 (PLC),图形、不同对话图像(面板)、处理趋向的图形表示等等,必须从正确版本的应用程序中检索数据,如之前所提及的。监督自动化处理过程的操作员或用户,例如处于上述的集中控制区域之中,就必须从控制物理输入/输出(I/O)设备的版本中检索数据,按顺序向处理过程输出命令,其中该命令例如是影响阀门的打开或调整容器的高度的命令。另一用户可能想要估计新版本的应用程序,那就应当从该新版本中检索数据,而不是从控制该处理过程的版本中检索。当然,其它用户也因某些原因需要从另外的版本中检索数据。

[0024] 按照本发明,为实现自动化提供一种方法和系统,从而能够在控制器中执行简单地管理两个或多个不同版本的方法。

[0025] 为了便于透彻地理解本发明,某些构成部分或设备采用简短的描述。OPC(用于过程控制的 OLE)是由工业自动化产业工作组开发出来的标准,而 OPC 服务器是管理 OLE(对象链接和嵌入)应用程序的服务器。OPC 服务器通常提供一种使许多不同的软件包或应用程序从过程控制设备中存取数据的方法,这里的过程控制设备例如是可编程逻辑控制器。OPC 服务器使用 OLE 技术,与 OPC 客户机通信。在下文中,贯穿整个说明书始终,使用 OPC 服务器和 OPC 客户机分另作为通信服务器和客户机的实例,但这不是试图限制本发明的范

围。可以理解,任何通信服务器都可以使用,如同可以使用任何客户机一样。

[0026] 图 1 示出了实现本发明的自动化系统 1 的示意图。在该示意性的实施例中,OPC 客户机 3a, 3b, ..., 3n 可以是不同的用户工作站或操作员工作点(例如为个人计算机(PC)),以用于显示不同的环境。例如,OPC 客户机 3a 可以是具有生产环境的用户工作站,在该环境中,利用现行的版本控制处理过程;客户机 3b 可以是工程师工作站,其具有工程技术环境并且在新版本的应用程序取代现行版本之前对该新版本进行估计;以及客户机 3n 可以是另一个工作站,其具有先前的环境并且执行旧版本的应用程序。工作在先前的环境中的用户可能想要从其它版本中检索数据,例如可以是旧版本,该旧版本当前并没有控制 I/O 设备并且为使备份版本有效而使用。使用旧版本的用户能够在该旧版本不再控制该 I/O 设备之后,看到该旧版本的值,这有时候是便利的,并且对于安全复原功能来说也是必需的。OPC 客户机 3a, 3b, ..., 3n 连接到 OPC 服务器 2,并在它们之间存在着双向链接。

[0027] 当在控制器中执行几个版本的应用程序时,图形、不同的对话图像(面板)、处理趋向的图形表示等等,必须从正确版本的应用程序中检索数据,如之前所提及的。监督自动化处理过程的操作员可能想监督与这种处理过程相关联的图形。例如,用户会希望监督容器的高度,因此,用户就监视示出了这种高度的图形。在下面,容器高度作为一个待监视的变量的示意性的实例。可以理解,容器的高度只作为一个示意性实例来使用,当然也可以使用其它的处理过程和相关的变量。而且,还可以理解,容器高度(tank level)的值通常不会因版本的不同而不同,但是基于这些值所采取的动作可能是不同的;容器的高度例如可能会依据某一版本而提高,而依据另一版本而不会提高,因此,控制该高度的阀门在某一版本中被调节而在其它版本中不被调节。然而,为了简化说明,通篇使用变量名称“Tank\_level”。与容器高度相关的图形,例如图表,检索该变量“Tank\_level”。在生产环境中工作的客户机然后必须检索数据,也就是从控制物理输入/输出(I/O)设备的版本中检索“Tank\_level”的值,其中该设备按顺序向处理过程输出命令,这种命令例如是影响容器的高度的调整的命令。如上所提及的,其它用户可能想估计新版本的应用程序,他们就应当从这种新版本中检索数据而不是从控制处理过程的版本中检索。不监督当前处理过程的用户,例如估计新版本的用户,可能仍然想监视容器的高度以及与之相关的图形,该高度仍然用变量“Tank\_level”来表示。然而,变量“Tank\_level”的实际值当然可以是不同的,这是由于这些值是与不同版本的应用程序相关联的。这样,按照本发明,在不同的版本中使用相同的变量名称,但是按照所讨论的版本来检索相同的变量的值。这样,本发明就可以不用对与某个处理过程相关联的图形进行重新配置。新的变量当然也可以引入到新版本中,其它的变量也可以移除,但是版本间共有的变量具有相同的变量名称。

[0028] 根据本发明,为 OPC 客户机 3a, 3b, ..., 3n 中的每一个提供新的 OPC 处理机 6a, 6b, ..., 6n。OPC 处理机 6a, 6b, ..., 6n 规定了 OPC 客户机运行环境,从而使 OPC 服务器知晓从哪一个版本中获取数据;例如,OPC 客户机 3a 具有 OPC 处理机 6a,在该处理机 6a 中包含了与 OPC 客户机 3a 所运行的生产环境有关的信息以及与待使用的应用程序版本 Vx 有关的信息,也就是要检索的变量值。由于 OPC 客户机 3a 在生产环境中运行,可假定它就是控制处理过程的版本。相似的,工作在工程技术环境中的工程师应当从最新的版本中接收变量值,也就是被估计的版本;而工作在之前的环境中的工程师应当从旧版本中接收变量值。可以认识到,也可以相似地使用除了上述 3 个环境之外的其它环境。

[0029] OPC 服务器 2 采用双向的方式进一步连接到控制器 8,并用这些实体之间的通信是采用通信包来执行的。不同版本的应用程序  $V_1, V_2$  存储在控制器 8 中。控制器 8 包括输入 / 输出单元 9,并且版本  $V_1$  和  $V_2$  中的一个控制自动化系统 1 中的处理过程。在该示意性实施例中,只示出了两个版本  $V_1$  和  $V_2$ ;然而,可以理解,也能以相似的方式管理多于两个的版本  $V_1, V_2, \dots, V_n$ 。

[0030] 按照本发明,OPC 服务器 2 包括存储装置 5,其用于跟踪客户机  $3a, 3b, \dots, 3n$  采用了各版本的应用程序  $V_1, V_2, \dots, V_n$  中的哪一个。存储装置 5 可以是任何适当的数据存储装置,例如 ROM(只读存储器)、PROM(可编程只读存储器)、EPROM(可擦除 PROM)、闪存、EEPROM(电 EPROM)、SRAM(静态 RAM)或 DRAM(动态 RAM)。OPC 服务器 2 知道哪个客户机处于哪个环境,从而知晓哪个版本的值是要使用的,这使得 OPC 能够向各客户机传送某个变量(例如“Tank\_level”)的正确值。

[0031] 控制器 8 在不同的存储位置存储用于不同版本各变量值。OPC 服务器 2 包括对某个存储位置进行寻址的装置,其中该存储位置包含正确的变量值,OPC 服务器 2 可以包括用于存储这种信息的存储装置 5,从而使 OPC 服务器 2 通过寻址能够请求正确的值。存储装置 5 可以是任何适当的数据存储装置,例如是上面所提及的任意一种类型。寻址例如可以包括标识  $ID_1, ID_2, \dots, ID_n$ ,它们对于每个版本的应用程序  $V_1, V_2, \dots, V_n$  是唯一的。然而,也可以采用其它寻址方式。标识符  $ID_1, ID_2, \dots, ID_n$  可以是地址的值、全局唯一标识符(GUID)(通常对于系统中的每条数据,是数字串和 / 或字母串)、时间戳,或者使这些版本具有自身唯一的标识符  $ID_1, ID_2, \dots, ID_n$  的任意适当的标识方式。因此,不同的版本或者实际上不同版本的变量值,能够从所有的其它版本中唯一地区分出来。这些唯一的标识符  $ID_1, ID_2, \dots, ID_n$  包含在控制器 8 与 OPC 服务器 2 之间通信的通信包中。OPC 服务器 2 总是知道每个 OPC 客户机  $3a, 3b, \dots, 3n$  正在使用哪个版本,这样,控制器 8 就会总是将正确的版本的数据提供 OPC 服务器 2,其中该 OPC 服务器 2 按顺序与 OPC 客户机  $3a, 3b, \dots, 3n$  通信。所有的通信都在不需要客户以任何方式定义或规定从哪一版本中检索数据的情况下执行,也就是说,客户不必知晓控制器 8 中正在使用的特定版本的应用程序的唯一标识符;这对用户来说是完全透明的。

[0032] 例如,如果版本  $V_1$  是当前控制特定处理过程的版本,并且它正要与新版本  $V_2$  互换,控制器 8 就在两个版本  $V_1, V_2$  在控制器 8 中切换角色的同时,也就是新版本  $V_2$  接管 I/O 单元 9 的控制的同时,发送信号给 OPC 服务器 2。从而,OPC 服务器 2 无论什么时候都能知道哪个版本在控制控制器 8 的 I/O 单元 9。这样,OPC 服务器 2 能够向每个 OPC 客户机  $3a, 3b, \dots, 3n$  传送正确的数据。例如,执行使用相同变量名称的不同版本的客户机请求变量“Tank\_level”,OPC 服务器 2 指出对应每个版本要获取的相应值的不同存储位置。OPC 服务器 2 按顺序将这些从客户机  $3a, 3b, \dots, 3n$  接收的请求翻译成存储地址,控制器 8 从而能够检索正确的值并将它们传送给 OPC 服务器 2,OPC 服务器 2 再将它们提供给各客户机  $3a, 3b, \dots, 3n$ 。

[0033] OPC 客户机  $3a, 3b, \dots, 3n$  可以从控制器 8 中预订变量值(例如,为了显示某个处理过程的图像),而不需知道实际的版本信息,并仍然可以从控制器 8 中接收正确的数据。不需要再次预订就能实现这些功能,这样就不需要进行重新配置。假如在控制器 8 中不同版本的应用程序在切换角色,就可以在 OPC 服务器 2 中简单地对预订进行重定向。所有的

这些功能都不需要与 OPC 客户机 3a, 3b, ..., 3n 进行交互就能实现。

[0034] 通过本发明的方式,不需要为了从其它版本的应用程序中,获得例如处理过程中正在使用的数据来重新配置图形。

[0035] 图 2 示出了包括用于执行根据本发明的方法的步骤的流程图。在方法 100 中,在控制器 8 中至少执行两个不同版本的应用程序  $V_1, V_2, \dots, V_n$ , 见步骤 100。在随后的步骤中,见步骤 120,使用不同版本  $V_1, V_2, \dots, V_n$  的至少两个客户机 3a, 3b, ..., 3n 请求与某个处理过程相关联的相同变量的值,例如变量“Tank\_level”。其后,在步骤 130 中,通信服务器 2 为所请求的变量检索相应值,并将它们传送给相应的客户机 3a, 3b, ..., 3n。也就是说,每个版本的值被检索并被传送到发出请求的客户机 3a, 3b, ..., 3n。

[0036] 通过本发明的方式,能够在控制器 8 和 OPC 服务器 2 之间建立不同版本  $V_1, V_2, \dots, V_n$  的独立通信。

[0037] 在说明书中,假设在同一控制器中驻留着两个不同的版本。但是,在其它可选的实施例中,被管理的版本也可以驻留在不同的控制器中,而仍旧可以采用所描述的原则,用户不需要主动地参与到数据检索的活动中来,他们只需要定义他们所工作的环境和所使用的版本就可以。这种定义通过 OPC 处理机 6a, 6b, ..., 6n 可以很容易地实现。

[0038] 总之,本发明提供一种在控制器中执行几个不同版本的应用程序时处理所出现的问题的方法。该控制器能够支持用于不同版本的独立通信。按照本发明提供了新的 OPC 处理机,OPC 服务器能够跟踪哪个客户机与哪个版本相关联并且与两个不同版本进行通信,是可以互相同步且独立的。

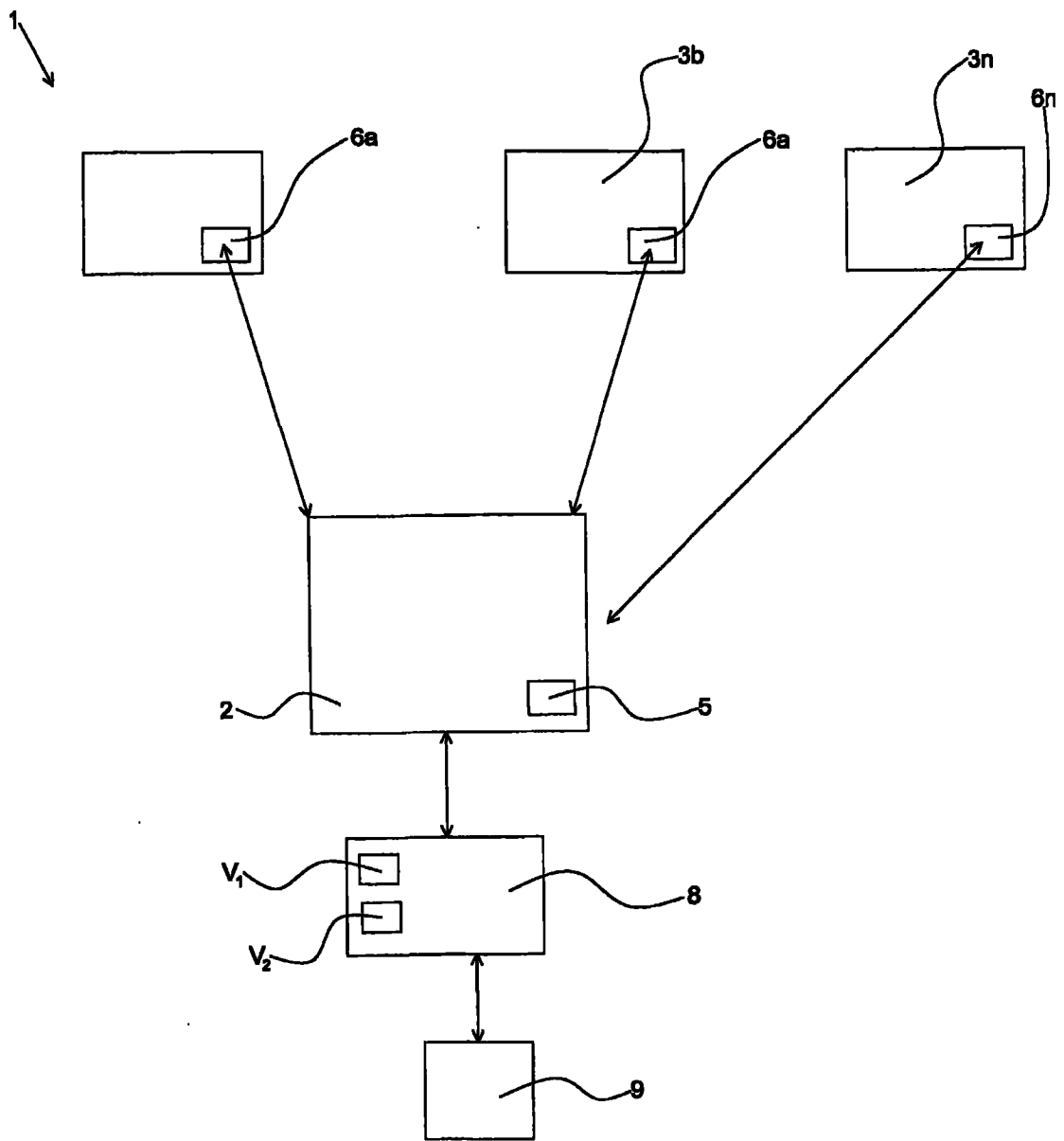


图 1

100  
↘

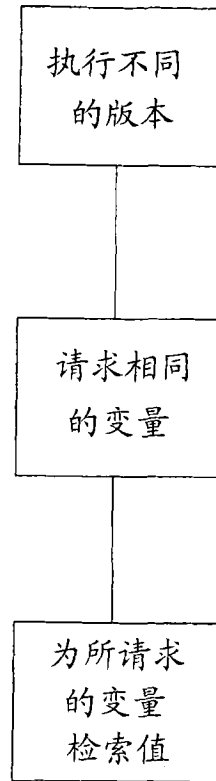


图 2