

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200810169386.3

[43] 公开日 2009 年 3 月 18 日

[51] Int. Cl.
G06Q 30/00 (2006.01)
G06F 9/44 (2006.01)
G05B 19/05 (2006.01)

[11] 公开号 CN 101388101A

[22] 申请日 2002.8.13

[21] 申请号 200810169386.3

分案原申请号 02818645.1

[30] 优先权

[32] 2001.9.24 [33] US [31] 09/961881

[71] 申请人 西门子能量及自动化公司

地址 美国佐治亚州

[72] 发明人 G·罗

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 魏军

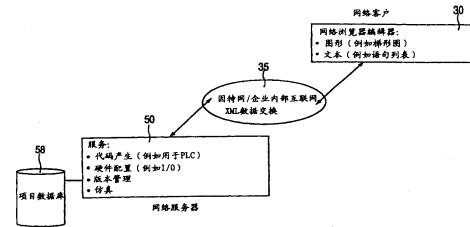
权利要求书 6 页 说明书 18 页 附图 27 页

[54] 发明名称

提供工程工具服务的方法

[57] 摘要

本发明披露了一种用于为可编程控制器生成程序代码和/或配置数据的方法和系统，以及它们在其上面进行工作的网络。在一个实施例中，所述程序代码生成于具有支持网络的工程工具的中央服务器。该工程工具在客户装置上工作，客户装置通过网络例如因特网与服务器通信。优选地，工程工具在客户装置的浏览器应用程序(或其它网络接口应用程序)中运行，并且用户借助客户装置在服务器上生成程序代码。通过工程工具的集中化，提供了服务于给拥有可编程控制器的顾客提供工程工具的新的商业范例，其中工程工具可以包含配置编辑器和具有版本管理功能的配置管理器。另外，本发明提供了新的用于程序代码合作工程的方法和工具。



1. 一种允许具有一客户装置的用户使用工程工具给可编程逻辑控制器编程的方法，该方法包括步骤：

 允许该用户的客户装置访问存在于网络上的服务器，该服务器不为该可编程逻辑控制器包括；

 在该服务器上保持一个支持网络的工程工具，使其能够被运行在所述客户装置上的浏览器应用程序访问；

 允许所述用户使用该支持网络的工程工具，该支持网络的工程工具适用于从用户接收程序代码，所述服务器适用于编译所述程序代码，被编译的程序代码适用于所述可编程逻辑控制器；及

 接收来自所述用户的费用，作为使用所述工程工具的交换。

2. 如权利要求1所述的方法，其中基于所述工程工具的使用情况计算所述费用的数值。

3. 如权利要求2所述的方法，其中基于所编写的程序代码的行数计算所述工程工具的使用情况。

4. 如权利要求2所述的方法，其中基于所述客户装置曾与服务器通信的时间计算所述工程工具的使用情况。

5. 如权利要求1所述的方法，进一步包括，在接收来自所述用户的费用之前，编译所述程序代码。

6. 如权利要求5所述的方法，其中所述费用的数值基于所述程序代码涉及的可编程逻辑控制器输入的数量。

7. 如权利要求6所述的方法，其中所述费用的数值基于所述程序代码涉及的可编程逻辑控制器输出的数量。

8. 一种用于出售编程服务的方法，该编程服务用于给可编程逻辑控制器开发程序代码，该方法包括步骤：

 允许用户使用适用于从该用户接收程序代码的工程工具，所述工程工具被保持在服务器上，该服务器不为该可编程逻辑控制器所包括，该程序代码为该可编程逻辑控制器所用；

在服务器上编译通过该工程工具开发的程序代码；及
向所述用户收取程序代码的费用。

9. 如权利要求8所述的方法，进一步包括，在接收来自用户的数值的步骤之前，把经过编译的程序代码下载到所述可编程逻辑控制器。

10. 如权利要求9所述的方法，进一步包括在所述服务器上将已经编译的程序代码的拷贝存档的步骤。

11. 一种用于给可编程逻辑控制器编程的方法，包括：

使用具有网络接口应用程序的客户装置访问被保持在服务器上的支持网络的工程工具，该服务器不为该可编程逻辑控制器所包括；

在该客户装置上的网络接口应用程序中运行该工程工具；

从该客户装置在该工程工具上接收程序代码，该程序代码为可编程逻辑控制器所用；

在该服务器上编译所述程序代码；

把经过编译的代码从所述服务器下载到可编程逻辑控制器；及

交换数值。

12. 如权利要求11所述的方法，其中所述网络接口应用程序包括web浏览器，及所述支持网络的工程工具包括支持web的工程工具。

13. 如权利要求11所述的方法，其中所述服务器与客户装置通过因特网通信。

14. 如权利要求11所述的方法，其中所述把经过编译的代码从服务器下载到可编程逻辑控制器的步骤包括步骤：

把经过编译的代码下载到所述客户装置，该客户装置与所述可编程逻辑控制器进行通信；及

把经过编译的代码从该客户装置下载到可编程逻辑控制器。

15. 如权利要求11所述的方法，其中所述把经过编译的代码从服务器下载到可编程逻辑控制器的步骤包括步骤：

把经过编译的代码下载到第二客户装置，该第二客户装置与所述可编程控制器连接；及

把经过编译的代码从该第二客户装置下载到可编程逻辑控制器。

16. 如权利要求15所述的方法，其中所述第二客户装置与服务器通过局域网通信。

17. 一种用于将程序代码下载到可编程逻辑控制器的方法，包括步骤：

 将该可编程逻辑控制器连接到网络；

 将程序代码从该网络下载到可编程逻辑控制器，其中该程序代码曾借助于被保持在服务器上的工程工具而被生成，所述工程工具可以被用户使用客户装置通过网络访问，该工程工具适用于从该客户装置接收该程序代码，该服务器不为该可编程逻辑控制器所包括，该服务器编译该程序代码，该程序代码为该可编程逻辑控制器所用；及

 付费。

18. 如权利要求17所述的方法，其中所述可编程逻辑控制器通过把所述可编程逻辑控制器与客户装置连接而连接到网络，所述客户装置连接到所述网络。

19. 如权利要求17所述的方法，其中将程序代码从网络下载到可编程逻辑控制器的步骤包括步骤：

 建立所述网络与所述客户装置之间的通信；

 将所述程序代码下载到客户装置；

 将所述可编程逻辑控制器与客户装置接口；及

 将所述程序代码从所述客户装置下载到可编程逻辑控制器。

20. 一种用于给具有可编程逻辑控制器的企业提供工程工具的方法，该方法包括步骤：

 给该企业提供服务器，该服务器允许客户访问所述工程工具，所述工程工具配置为可以被客户装置访问，所述工程工具被保持在所述服务器上，所述工程工具适用于从该客户装置接收程序代码，该服务器不为所述可编程逻辑控制器所包括，该服务器编译该程序代码，该程序代码为所述可编程逻辑控制器所用；

 在服务器上保持所述工程工具；及

 收取费用。

21. 如权利要求20所述的方法，其中基于所述工程工具的使用计算所述

费用。

22. 如权利要求20所述的方法，其中所述费用是固定费用。
23. 如权利要求20所述的方法，其中所述费用包括非货币价值的传输。
24. 如权利要求20所述的方法，进一步包括将所述服务器连接到企业内部互联网，该客户装置可通过所述企业内部互联网通信。
25. 如权利要求22所述的方法，进一步包括将所述服务器通信连接到局域网的步骤，该局域网通信连接所述可编程逻辑控制器。
26. 一种出售工程服务的方法，所述工程服务允许机构生成用于可编程逻辑控制器的程序代码，该方法包括步骤：

提供到被保持在服务器上的工程工具的访问，所述工程工具适用于接收为可编程逻辑控制器所用的程序代码，所述服务器不为所述可编程逻辑控制器所包括，所述程序代码为所述可编程逻辑控制器所用；

从客户装置接收用该工程工具生成的程序代码；
在服务器上编译该程序代码；及
提供到仿真软件的访问，以测试经过编译的程序代码。

27. 如权利要求26所述的方法，进一步包括从所述服务器发送已经编译的代码到客户装置。

28. 如权利要求26所述的方法，进一步包括将已经编译的代码从所述服务器发送到第二客户装置。

29. 如权利要求28所述的方法，其中所述第二客户装置与可移动存储装置耦合。

30. 如权利要求29所述的方法，其中所述可移动存储装置是磁盘驱动器。
31. 如权利要求29所述的方法，其中所述可移动存储装置是闪存装置。
32. 一种用于提供工程工具服务来配置I/O网络和设备的方法，该方法包括步骤：

在网络上保持服务器，所述服务器具有配置编辑器，该配置编辑器配置为可被客户装置上的网络接口应用程序调用，所述客户装置通过网络与所述服务器通信，工程工具的该配置编辑器部分被保持在所述服务器上，所述工程工具适用于从所述客户装置接收程序代码，该服务器编译该程序

代码，所述程序代码为所述可编程逻辑控制器所用，所述服务器不为可编程逻辑控制器所包括，所述配置编辑器适用于产生代码，所述代码适用于配置I/O网络和装置；

接收使用所述配置编辑器的数值；及

通过所述网络发送使用所述配置编辑器生成的配置数据。

33. 如权利要求32所述的方法，其中所述配置编辑器包括图形编辑器。

34. 如权利要求32所述的方法，其中所述配置数据包括波特率信息。

35. 如权利要求32所述的方法，其中所述配置数据包括有关将逻辑I/O信息映射到物理I/O信息的数据。

36. 一种用于生成配置数据的方法，包括步骤：

在网络上保持服务器；及

在该服务器上保持配置编辑器，所述配置编辑器在客户装置上运行，工程工具的该配置编辑器部分被保持在所述服务器上，所述工程工具适用于从所述客户装置接收程序代码，该服务器编译该程序代码，所述程序代码为所述可编程逻辑控制器所用，所述服务器不为该可编程逻辑控制器所包括，所述配置编辑器适用于产生代码，所述代码适用于配置I/O网络和装置。

37. 如权利要求36所述的方法，其中所述配置编辑器运行在所述客户装置上的网络接口应用程序中。

38. 如权利要求37所述的方法，其中所述网络接口应用程序是web浏览器。

39. 如权利要求36所述的方法，其中所述网络是因特网。

40. 如权利要求36所述的方法，其中所述配置编辑器包括图形编辑器。

41. 如权利要求40所述的方法，进一步包括在所述服务器上保持工程工具，该工程工具适用于给所述可编程逻辑控制器开发程序代码。

42. 一种提供工程工具服务的方法，包括步骤：

在服务器上保持工程工具，该工程工具适用于给可编程逻辑控制器接收程序代码和配置数据，该工程工具配置为运行在与所述服务器通信的客户装置上，所述服务器不为所述可编程逻辑控制器所包括，所述程序代

码为所述可编程逻辑控制器所用；

在服务器上编译通过工程工具生成的程序代码；及

在所述服务器上运行仿真软件以测试已经编译的程序代码。

43. 如权利要求42所述的方法，进一步包括接收数值。

44. 一种用于提供工程工具服务的方法，包括步骤：

在服务器上保持工程工具，该工程工具适用于给可编程逻辑控制器接收程序代码和配置数据，所述工程工具配置为运行在与所述服务器通信的客户装置上，所述服务器不为所述可编程逻辑控制器所包括，所述程序代码为所述可编程逻辑控制器所用；

在服务器上编译通过工程工具生成的程序代码；及

将一个测试可编程控制器与所述服务器接口，该测试可编程控制器配置为测试程序代码和配置数据。

45. 一种用于可编程控制器的版本管理方法，该方法包括步骤：

在服务器上存储多个程序代码的版本，所述程序代码适用于为可编程逻辑控制器而在所述服务器上被编译，所述程序代码在被保持在所述服务器上的工程工具之外被接收，所述程序代码为所述可编程逻辑控制器所用，客户装置通过网络可以访问该服务器，所述服务器不为所述可编程逻辑控制器所包括，该服务器编译该程序代码；

响应来自所述客户装置的命令，在所述多个程序版本中查找预先确定的程序中的变化；及

从与所述客户装置相关的用户接收数值，以交换在所述服务器上存储为所述可编程逻辑控制器所用的所述多个程序版本。

46. 如权利要求45所述的方法，其中所接收的数值是基于存储在所述服务器上的所述程序代码的版本数量。

47. 如权利要求45所述的方法，其中所接收的数值是基于存储所述多个程序代码的版本所用的空间。

48. 如权利要求45所述的方法，进一步包括将程序的第一版本与第二版本比较，做为在所述程序代码的版本中查找预先确定的所述程序代码中的变化的步骤的结果。

提供工程工具服务的方法

技术领域

本发明一般涉及可编程控制器领域。具体地，本发明提供一种方法，所述方法用于给用户提供鲁棒的工程工具服务，而无需以物理形式给用户提供工程工具。

背景技术

可编程控制器，包括可编程逻辑控制器（“PLC”）用于许多商业及工业应用。可编程控制器典型地监控输入，根据其如何编程来产生结果，并控制自动化程序或机器的输出。使用的最普通的可编程控制器之一是PLC。PLC包括输入模块或输入点，中央处理单元（“CPU”），及输出模块或输出点。输入接收来自不同类型的装置例如传感器的各种数字或模拟信号，并将其转换为可以被CPU使用的逻辑信号。CPU根据存储在存储器中的程序指令作出决定并执行控制指令。这些程序指令决定PLC对特定的输入会干什么。输出模块将来自CPU的控制指令转换为数字或模拟信号以便控制各种现场装置，例如执行机构或阀。

既然多数包括PLC在内的可编程控制器本质上属于计算机，它们以通（On）或断（Off）的形式（就是1或0）存储信息，称为二进制数字（就是位）。可编程控制器的程序包括一个或多个实现一项任务的指令。给PLC或其它控制器编程就是建立一系列指令的工作。编程通常也包括生成配置数据。配置可编程控制器包括将控制器的输入/输出（“I/O”）域映射到物理I/O。配置编辑通常是图形形式的。

查看程序有几种方法，例如，设计流程图，梯形逻辑（Ladder Logic），指令列表，或功能块图。梯形逻辑（“LAD”）是PLC使用的一种程序语言。如图1所示，梯形逻辑编码10使用与用在继电器逻辑框图中的机电元件相类似的图形符号来描述硬连线控制。在一典型的梯形逻辑框图中的左侧的垂直线通常表示一电源或带电导体。右侧的垂直线，表示在硬连线控

制线路图中的返回路径，可被省略。梯形逻辑框图是从左向右，从上向下阅读。梯线通常称之为网络。网络可有几个输入输出指令。由一系列在一个或多个平行的分支中的触点表示的输入指令执行比较或测试，且在梯线中通常是左调整的。由线圈表示的输出指令，因此其在每个输出分支中可能只有一个，执行一些操作或功能，且在梯线中通常是右调整的。如图1中所示的梯形逻辑编码10所示，IO.0, IO.1, 及 QO.0 表示第一指令组合。如果输入IO.0及IO.1都受到激励，则输出QO.0受到激励。输入可以是开关，按钮，或接点闭合。输出可以例如是线圈或灯泡。IO.4, IO.5, 及 QO.1 表示第二指令组合。如果输入IO.4或IO.5任一个受到激励，则输出QO.1受到激励。

指令列表 (“IL”) 提供一组指令的另一种形式，在图1中用20所示。操作，就是将要完成的工作，示于图的左侧。操作对象，就是被操作的对象，示于图右侧。LAD及IL具有相似的结构。在图1中的IL20中的那组指令执行与10中的LAD 相同的任务。

功能块图 (“FBD”) 提供一组指令的另一种形式，照图1中用30表示。各功能块有表明其特定任务的名称。功能通过矩形表示。输入显示在矩形的左侧，输出示于其右侧。图1中所示的功能块30执行与图1中所示的LAD10和IL20相同任务。

一般的可编程控制器，具体地说是PLC，以重复的步骤执行程序码，称之为扫描。扫描可以在CPU读取输入状态时开始。应用程序使用输入状态来执行。一旦程序完成，CPU执行内部诊断和通信用务。扫描循环更新输出后结束，然后重新开始。循环时间依赖于程序的大小，I/O的数量，和需要的通信量。

为了编写程序指令和生成配置数据及将代码和配置数据下载到可编程控制器，需要几种工具。如图1所示，编程装置，个人计算机1，与PLC对接典型地，专用电缆5，如Siemens® PC/PPI, 将PLC7连接到计算机的RS232口。在本发明之前，工程软件工具3，例如西门子STEP7®, 必须安装在PC上，以使PC可以用于给PLC编写程序指令。典型地，该工程工具在CD上或其它计算机可以读取的介质上销售。

图2概述了购买一份工程工具拷贝导致的典型步骤和缺点。工程工具

的用户通常购买一份软件拷贝并获得使用该工具的许可，步骤20。用户然后必须将该软件安装到他的个人计算机上，步骤25。用户只能开发应用软件，就是在具有该工程工具的计算机上编写可编程控制器的程序代码，步骤27。通常伴随该工程工具的许可协议限制用户在没有支付允许更多计算机的协议的情况下将工具安装在多于一台的计算机上。

上述软件或工程工具，如西门子STEP7®或MicroWin®，经常是控制器制造商开发的专用工具。典型地，开发这些工程工具需要数千人工时。经常设计和测试这些工具，以使其工作在特定的计算机操作系统上，例如Microsoft Windows®98。当应用该工具的计算机的操作系统改变时，该工具需要重新验证。PC的出售者常常仅在他们的PC上安装最新的微软操作系统。这迫使工程工具的出售者也支持新的操作系统，这通常意味着额外增加的数百或数千人工时的投资。在许多组织中，PC操作系统升级时没有注意该PC上存在的软件，入上述工程工具。

工程工具也经常随着时间更新。因此，这些工具的不同版本会同时存在。在一个大型的制造设备中可能不是所有的编程工具都使用相同的版本。这不仅增加了所有者的花费，而且当使用不同的编程装置为相同的PLC应用写程序代码时还会引起问题。经常安排一组工程师为PLC编程。这个组中的每位工程师可能独立负责应用的一个独立的方面。然后，这些不同的独立方面被组合成一个在PLC上工作的程序。如果工程师不都使用此工具的相同版本，从此工具某一版本产生的代码有可能与此工具另一更早期版本不兼容。

除了关于在编程装置上运行不同操作系统及不同版本的工程工具在操作系统上运行的问题外，用于可编程控制器的程序代码经常不是以集中的方式存档。在工厂中用于一个PLC的代码可能存储在一个膝上型电脑或台式个人计算机中，而用于另一PLC的代码可能存储在别处。如果一个PLC最初使用第一台PC编程，并且这台PC后来被第二台PC代替，以及如果这个PLC使用第二台PC重新编程，常常会没有办法恢复原始程序，这使得新程序不完善。

发明概要

本发明提供一种方法，用于给用户提供一给可编程控制器编程的工程工具。不是出售一工程工具的物理形式的拷贝，而是给最好具有一客户装置的用户提供到网络上的服务器的访问权限。该工程工具提供者或其它机构在服务器上保留一份支持网络应用的工程工具。该工具可以通过运行在客户装置上的浏览器应用软件进行访问。用户可以使用该工程工具给用户的可编程控制器创作程序代码。该工程工具提供者从用户处收费做为使用该工具的交换。有利地，本发明省去了工程工具提供者提供安装支持和服务的需要。

本发明提供了工程工具市场的一新的范例。在一个实施例中，给用户提供了给可编程控制器开发程序代码的工程工具的使用。使用该工具开发的代码被编译，并且向用户收取该程序代码的费用。在一些实施例中，在向用户收费以前，编译过的程序代码可以下载到可编程控制器。有一些情况下，可能希望在服务器上保存一份编译过的代码的拷贝，以及该代码的后续版本。该服务器也可以配备具有版本管理功能的配置管理器。这允许用户在一个版本中查找代码的特定属性，以及这些属性何时被改变。也可以允许用户比较代码的不同版本。

本发明也改进了给具有可编程控制器的机构开发用于其控制器代码的方式。不是购买和在编程设备上安装一工程工具，而是拥有一带有网络接口应用程序如网络浏览器的客户装置的工程师或其它用户，通过客户装置访问位于网络服务器上的工程工具。在一实施例中，网络是因特网的环球网部分，网络接口应用程序是网络浏览器。所述工程工具运行在客户装置中的网络接口应用程序上，并用来创建程序代码，其中可能包括配置数据。然后该工程代码在服务器上编译，然后编译后的代码从服务器上下载到所述可编程装置。有些地方，用户为使用工程工具创建的程序代码或使用工程工具而付费。

本发明提供了多种用于将在具有可访问工程工具的服务器上生成的程序代码下载到客户装置的方法。一种方法包括将可编程控制器连接到网络，从网络中将程序代码下载到可编程装置，并付费。在一个实施例中，程序代码生成于一具有可通过客户装置访问的工程工具的服务器。

本发明可以由需要高度保密的机构使用。在有些情况下，最好为拥有可编程控制器的企业提供其自己的具有支持网络的工程工具的服务器。该工程工具保持在服务器上，并允许客户通过客户装置访问此工具。企业为使用该工程工具付费。费用可以根据工具的使用次数或使用频率来计算。或者费用可以是一个固定(flat)费用。因为服务器公众无法访问，可以提供其它方法允许授权的企业人员访问该服务器。例如，服务器可以通过专用内部网访问或可以是局域网(LAN)的一部分。

附图简要说明

图1是现有技术中编程可编程控制器需要的工具的图解。

图2是表示现有技术中用于编程可编程控制器的方法和工具的步骤及缺点的方框图。

图3是本发明的一个实施例的图解，其中连接到一PLC的客户装置通过网络与服务器通信，所述服务器包括支持网络的工程工具，用于给PLC编写程序代码并配置I/O网络及设备。

图4是本发明的一个实施例的一般结构的图解，其中用于给可编程控制器编写程序代码，并用于配置I/O网络及设备的工程工具存在于服务器上并通过网络由客户装置访问。

图5是本发明另一可替换的实施例的图解，其中所述服务器还包括一网络服务层。

图6从客户和服务器的角度图示了按照本发明的在因特网上操作的支持网络的工程工具的组成，其中使用了图形编程技术。

图7图示了按照本发明的网络应用工程工具使用XML格式的另一可选择的方法。

图8图示了使用传统工具的应用程序接口通过网络启动根据本发明的传统工程工具。

图9是按照本发明支持网络的流程图编程工程工具的屏幕视图。

图10是按照本发明支持网络的梯形逻辑编程工程工具的屏幕视图。

图11是概述本发明一个实施例的方法步骤的流程图。

图12是概述按照本发明用于编程一可编程控制器的方法的流程图。

图13图示了按照本发明的一个系统的配置，其中一客户装置通过网络与服务器连接，所述服务器具有一支持网络的工程工具，以及一个通过网络连接于服务器的第二客户装置，所述第二客户装置在具有一个或多个可编程控制器的局域网（LAN）上。

图14图示了按照本发明的另一配置，其中可编程控制器直接与一具有支持网络的工程工具的服务器对接，所述服务器与客户装置进行通信。

图15图示了按照本发明的又一个结构，其中具有支持网络的工程工具的服务器在因特网或企业内部互联网上与客户装置连接，以及一个或多个可编程控制器通过因特网或企业内部互联网与服务器连接成网络。

图16是概述按照本发明用于提供工程工具服务的步骤的流程图。

图17是概述按照本发明用于离线生成程序代码步骤的流程图。

图18是按照本发明用于出售工程工具服务的一个新的商业范例的图解。

图19是按照本发明用于出售工程工具服务的另一个新的商业范例的图解。

图20是本发明一个实施例的图解，其中人可阅读的工程代码存储在服务器上并与运行在可编程控制器上的编译后的代码链接，并可访问一自动服务器。

图21示出了本发明在一家公司或商业实体中如何建立。

图22是按照本发明的一个合作工程系统的图解。

图23是概述按照本发明开发和测试可编程控制器和/或配置数据的程序代码的方法的流程图。

图24是概述按照本发明用于生成程序代码，并将程序代码安装到可编程控制器上的方法的流程图。

图25示出了本发明的一个实施例，其中位于服务器上的一工程工具组包括配置编辑器，并可通过网络被客户装置调用。

图26是概述由本发明实施的合作工程方法的流程图。

图27是概述由本发明实施的另一可选择合作工程方法的流程图。

图28是概述由本发明实施的又一可选择的合作工程方法的流程图。

本发明详细描述

除了其它方面，本发明还提供了用于在一中心服务器上编写程序代码（这通常也包括生成配置数据），及用于将编译后的程序代码连同任何配置数据下载到可编程控制器，如PLC，的系统和方法。如图3所示，一客户装置30例如一台包含一浏览器的个人计算机位于一公用或专用网络35中。所述浏览器例如可以是微软Internet Explorer[®]或Netscape Navigator[®]。网络35可以是因特网或者是企业内部互联网。在一个实施例中，PLC40，如Siemens[®]S7-200PLC，通过一接口电缆45例如Siemens[®]PC/PPI电缆连接到客户装置30。该电缆可以包括控制设置如波特率的双列直插式开关48。在此实施例中，客户装置30做为给PLC40编程的编程装置。但是，与现有技术中用于给PLC40编程的工程工具安装在编程装置上的情况不同，对程序代码进行写和编译所需的工程工具位于服务器50上，客户装置30通过因特网或企业内部互联网或其它的网络35可以访问服务器50。该工程工具最好能够支持网络，以在任何客户装置30中的浏览器应用程序中进行操作。

用户55用客户装置30通过网络35访问服务器50。客户装置30可以是很小的客户机，即，除了浏览器外它不包括其它任何软件。这样，客户装置30可以采用多种形式，包括个人计算机，支持网络的PDA，蜂窝电话，支持因特网的电视装置，和因特网设备，等等。本发明是可伸缩的，因为它可以在具有不同功能的不同的客户装置上进行不同模式的操作。例如，本发明可以在一具有有限功能和计算能力的支持网络的移动电话浏览器上操作有限的功能，或者可以在一台具有高速处理器和大容量存储器及强大的图形能力的先进的个人计算机上操作强大的功能。

按照本发明，工程工具，类似于诸如西门子STEP7[®]等工具的工程工具存在于服务器50上并支持网络。典型地，这些工程工具将使用图形编程方案，如梯形逻辑图，流程图编程，功能序列图，或功能块图。（参见图1）。优选地，它们配置为处理不同的程序语言，包括指令表编程和所有其它标准语言，例如所有IEC-1131标准语言（例如，LAD,FBD,SFC,IL和ST）。这些工具配置为运行在一浏览器应用程序中，这样使它们可以独立于客户机所运用的操作系统来操作。优选地，这些工程工具配置为使用HTML或

XML界面操作，这样它们可以在任何现有的浏览器中工作。这些工具也可以提供配置编辑器以将控制器逻辑I/O地址映射到一个物理I/O地址。

图4描述了用于支持网络的工程工具的一般的系统结构的一实施例。一网络客户装置30包括一标准的web浏览器。用于给可编程控制器生成程序代码的工程工具位于网络服务器50上并且能够在网络客户机上的浏览器应用程序中操作。该工程工具支持网络，并且在其最基本的形式中，其包括一个或多个可以运行在网络浏览器下的编辑器。有利的是，编辑器独立于操作系统并可以运行在最新的浏览器或操作系统和硬件中。网络服务器50也可以包括一编辑器，用于执行硬件配置。它也可以包括执行版本管理和运行仿真程序所需要的软件或硬件。网络客户30与网络服务器50都存在于网络35中，如互联网或企业内部互联网。用户在网络客户30中为可编程控制器创建程序代码和/或生成配置数据。这些代码和配置数据转换为XML文件并通过网络35传送到网络服务器50。程序代码和配置数据可以存储在与网络服务器50关联的项目数据库58中。项目数据库58可以是任何标准形式的存储介质。服务器50无需运行最新的平台（操作系统或硬件）。

图5描述了另一个实施例，其中网络服务层70与服务器50合并。另外的网络服务，例如仿真服务和工具，可以位于工程工具的最上层。这给工程工具的开发者提供了迅速和节省成本的方式来改进和开发新产品。工程工具的出售者可以提高和延长他们在开发工程工具上付出的投资的使用寿命。在这个实施例中，一基本工程工具的用户将仅仅通过鼠标点击即可获得供其使用的鲁棒的工程服务。

XML特别适合用于本发明，因为图形表示，例如通用PLC编程语言中所用的图形表示，很容易转换为XLM文件。这样，在客户30与服务器50之间优先地以XML文件的形式传输。

图6图示了使一个标准工程工具，例如使用图表编程技术的西门子STEP7®支持网络所需要的客户端，服务器端和因特网组件。尽管图形编程技术可用于使标准工程工具例如西门子STEP7®支持网络，但Microsoft®的.Net技术提供了一个使工程工具支持网络的更为有效的方法。如图7所示，.Net技术易于使用XML格式实现在网络服务应用程序99与网络服务器

之间的对接。它也允许客户一服务器之间进行XML格式的数据交换。

如图8所示，新的网络服务50可以建立在一个已经存在的工程工具77例如西门子STEP7®的应用程序接口（“API”）59的顶部。然后服务器50与客户装置30使用标准协议交换XML文件。

支持网络的、浏览器可执行的工程工具优选地具有用于自动项目不同阶段的用户友好功能。图9和10示出了典型的支持网络的工程工具在具有标准浏览器的客户装置中查看时的屏幕视图。图9示出了支持网络的流程图编程工程工具，以及图10示出了用梯形逻辑编程时支持网络的工程工具。典型地，这些工程工具包括各种功能，例如硬件的配置和参数化；通信的定义；编程；测试；启动和维护；记录与存档；及操作/诊断。这些工具也可以包括符号编辑器，用于定义符号名称，数据类型，及全局标记符注释。这些工具优选地支持全部编程语言，包括但不限于流程图编程，IL,ST,SFC,LAD,和/或FBD,以及由IEC指定的全部顺序编程语言。

如上面讨论的，工程工具提供的功能可以根据它们运行在其上的客户装置的类型缩放。例如，一个非常小的客户装置，例如一支持网络的移动电话，可能仅能够运行一个基于文本或有限图形编辑功能的工具。一个更复杂的客户装置，例如IBM600E Thinkpad®，可以支持更多的功能，和可以允许工程工具执行更多的功能，如在开发代码时的全图形编辑。这样，按照本发明，工程工具功能是在客户装置上执行还是在服务器端执行，取决于客户装置的性能，因特网或企业内部互联网速度，及其它参数。

如图3所示及图11的流程图所概括，在本发明的一个实施例中，一个用户（或几个用户）可以访问位于服务器50上的PLC编程工程工具38（或用于给其它可编程控制器编程的工程工具），步骤1000。在一个实施例中，服务器50位于因特网上的环球网部分。在另一个实施例中，服务器位于专用公司网络或公司的企业内部互联网上。用户55在客户装置30的网络浏览器上操作工程工具38，步骤1005，并在浏览器上创建用于可编程控制器的程序代码，步骤1010。可以使用标准安全措施限制对服务器和其上所创建的程序的访问。用户55可以在一次会话中创建一个程序，也可以在几次会话的时间段中创建程序。用户可以在浏览器应用程序中修改，编辑，并调

试程序代码，步骤1015。浏览器中生成的代码可以是图形方式。这些代码然后转变为一个文件，步骤1020，以及该文件通过网络传送到服务器，步骤1025。优选地，程序代码存储在服务器50上，步骤1030。

编辑软件，调试软件及编译器可以存在于服务器50上。这实际上允许任何PC,PDA,或其它支持网络的客户装置30做为编程装置。在用户55创建并编辑程序代码后，该代码可以在服务器上存储和编译，服务器优选地具有编译器。在一些实施例中，附加的服务器可以用于存档代码。此外，编译器也可以存在于可以访问具有工程工具的服务器的另一个服务器上。

代码创建、调试、并编译后，被下载到可编程控制器。如图12中的流程图所示，在一个实施例中，给可编程控制器编程包括两个步骤：在控制器与服务器之间建立通信，步骤2000，以及通过网络将程序代码下载到控制器，步骤2005。在一个实施例中，如图3中所示，代码可以下载到客户装置30，客户装置30然后又可以与可编程控制器40对接。在这个实施例中，已经编译的程序代码的一份拷贝可存在于客户装置或存档在客户装置上。

典型地客户装置30通过连接电缆45与可编程控制器40对接，如西门子PC/PPI连接电缆（见图3）。如图3所示，当PC做为客户装置30使用时，连接电缆允许位于可编程控制器上的串行接口与标准RS-232串行接口通信。当另一个装置例如PDA做为客户装置使用时，连接电缆与该装置上的通信端口对接。电缆可以如图3所示，包括DIP开关48，用于选择信息在PLC40与客户装置30之间传递合适速度（波特率）。客户装置30可以如图13所示，与多个可编程控制器40对接。在一些实施例中，在有大量的可编程控制器与一个客户装置接口的情况下，需要一个中继器（没有示出）与电缆连接。

在另一个可选择的实施例中，例如在图14中所示的例子，可编程控制器40可以直接或间接连接到网络，如LAN100,服务器位于该网络上，而无需通过客户装置30连接。在这个实施例中，可编程控制器40通过网络与服务器50进行通信，网络100可包括使用PCP/IP以太网协议，Profibus[®]网络，或实际上任何其它网络结构的标准网络。然后编译过的程序代码可以被直接下载到可编程控制器40。

在又一个实施例中，如图15中所示，可编程控制器40可以通过网络110

例如企业内部互联网或因特网与服务器50进行通信。在一些实施例中，可编程装置可以在使用例如TCP/IP或Profibus[®]或任何其它合适的协议的LAN100上相互链接。

无论可编程控制装置是否直接与服务器所在的网络对接，还是通过客户装置与服务器通信，一份程序代码的归档拷贝都可以存储在归档项目数据库120中，该数据库位于服务器50上或与服务器50关联的存储装置上。（见图14）。这样有利地给编译过的程序代码的备份拷贝提供了集中的存储。所述归档拷贝也可以存储在独立的服务器上或可通过网络使用的数据库中。

一些实施例中，希望离线，即在客户装置与包含工程工具的服务器断开连接的情况下创建用于可编程控制器的程序代码。离线编程允许用户在不与服务器连接的情况下创建和编辑程序代码。如图17中的流程图概要所示，用户在客户装置与服务器之间建立连接，该服务器包含支持网络地工程工具以及编程项目的数据库，步骤3000。接着用户在客户机上的浏览器应用中打开支持网络的工程工具，步骤3005。接着用户从项目数据库中检索出项目的拷贝，步骤3010，该项目的拷贝被传送到客户装置，步骤3015。客户与服务器之间的通信就此结束，步骤3020。用户现在可以离线自由编辑所检索的拷贝，步骤3025。当用户完成编辑，用户重新建立客户机与服务器之间的通信，步骤3030。接着编辑后的程序代码在3035被存储到原来的位置，并且可以被具有访问权限的其它用户访问。在一些实施例中，在检索出项目时，其它用户可被允许以其被检索之前的形式读取一份被检索出的项目的拷贝。依赖于客户装置的复杂程度和资源，离线编程技术和功能可以变化。

在客户装置配置不复杂时，可以提供基于文本的只能编辑的功能。当客户资源包括更高水平的复杂性（例如内存，图形能力，处理器速度，等等）时，可以提供更复杂的图形编辑/代码创建功能。这样，本发明可以缩放。例如，在一个运行于简单的支持网络的装置的实施例中，只能给用户提供文本编辑功能。而在另一个运行于更复杂的客户机的实施例中，可以给用户提供在包含全部图形编辑功能的网络浏览器中工作的工程工

具，所述图形编辑功能当用户连接到服务器时存在。

除了依赖客户资源（例如，内存）外，其它因素，如客户操作系统及安全问题，会影响支持网络的工程工具的功能。为了离线工作，装置的操作系统需要支持一些在客户装置上安装所述应用程序的方式。这通常不是现代的PDA的问题，但却可能是一些小装置如便携式电话的问题。因为安全的原因，网络浏览器应用程序通常被限制访问系统资源，如驱动器和系统内存。微软公司，雷德蒙·华盛顿提供了用于允许运行在其网络浏览器中的应用程序以通过应用程序中ACTIVE-X控制插件的方式访问客户系统资源的软件。对于那些希望更好安全性的用户，微软公司给用户提供了在其浏览器中屏蔽下载ACTIVE-X控件的功能，或者仅接受来自信任的站点下载，或仅接受原始认证的ACTIVE-X控件。

在一个实施例中，根据需要给客户装置提供特殊的功能。例如，语法检测在客户装置的用户需要这项功能时才提供。这允许用户最初下载最小的功能到客户装置，并仅在需要的时候添加附加功能。

通常情况下，当生成和编译程序代码时可编程装置无需连接到可编程控制器上。但是，一些实施例中希望可编程控制器与服务器和客户装置同时在线。在这个实施例中，将变化内容下载到可编程控制器。另外，可以监视输入和输出元件的状态。在一些实施例中，服务器可以安装诊断软件或是具有允许监测可编程控制器的功能。在其它的实施例中，服务器能够模拟输入和输出并且能够使之与可编程控制器通信。这允许将多样和鲁棒的测试算法集中创建并存储。在一些实施例中，鲁棒的测试算法仅可以被经授权的用户访问并且该用户可以为他们的使用付费。

本发明在工程工具产业中提供了一个新的范例。在本发明之前，工程工具属于被制造和出售的产品（见图2）。按照本发明，支持网络的工程工具提供了一新的商业范例，其中工程工具的生产者提供服务。（见图18和19）。在一个实施例中，例如图18中的流程图所示，允许用户访问包含工程工具的服务器，所述工程工具用于给可编程控制器开发和编译程序代码，步骤200。为了编写程序代码的目的，可以使用户自由访问工程工具。用户能够在与服务器通信的客户装置上开发代码，步骤205。用户可以在

无需付费的情况下创建和修改程序代码。这有利于给即使很小的商业业务提供访问复杂和最新的工程工具。仅当用户完成代码和编译后，才会要求付费，步骤207。在一些实施例中，所需费用会在程序代码编译时计算出来。在另外的实施例中，用户在将编译后的代码下载到PLC之前完成付费。(见图19，步骤220)。给代码付出的费用可以基于编译后的代码的大小，编写代码需要的时间，代码被设计要处理的输入和/或输出数量，或各种其它可以测量的参数。因为工程工具位于中心服务器而不是客户装置，限制和控制对工具的访问是简单的。此外，对于客户服务代表或销售人员来说，给顾客提供附加的工程工具用途是简单的事情，在此所述的附加用途有利的是来自商业预期。例如，当顾客购买了新的PLC或与已有的PLC接口的新的现场装置时，销售人员可以给其提供一个有限时段自由使用支持网络的工程工具来编程和配置PLC。

从建立和/或供应工程工具的机构的角度看，该新的商业范例可以采取图16所示的流程图的形式。工具生产者或提供者在服务器上保持支持网络的工程工具，客户可以通过网络例如因特网访问所述服务器，步骤1601。允许客户通过网络访问该工具，步骤1605。在一些实施例中，可以使用密码或其它安全措施限制只允许授权的用户访问；而在其它的实施例中，公众可以访问该工具，但没有必要访问使用该工具创建的代码。按照本实施例，工程工具提供者，例如通过网络，收到来自用户的文件，所述文件包含使用支持网络的工程工具生成的程序代码，步骤1610。然后该程序代码被编译，步骤1615，或可以被保存，并使创建程序代码的机构可以访问它们，或经授权的机构访问它们。最终，该代码被编译并下载到可编程控制器，步骤1620。

因为编码创建在中心服务器上，该编码可以以人可以阅读的形式和编译后的形式存档(见图20)。这提供了一种方法，用于当新的版本不能以期望的方式执行时在新的可编程控制器上重新安装现有的软件或者恢复旧版本的代码。在一实施例中，用户为下载一份存档拷贝付费。这笔费用可以低于用户下载第一份拷贝所付出的最初费用。

此外，因为在一些实施例中，人可阅读形式的代码(也称源代码)存

档并与机器可读代码链接，以便进行支持网络的监视，仿真，和控制自动设备。例如，如图20中所示，支持网络的PLC300可以包括一嵌入式网络服务器310。PLC可以连接到网络350，如因特网或企业内部互联网。PLC可通过本发明的支持网络的工程工具编程。用于PLC的程序代码可以人可阅读的形式和编译后的形式存在于工程服务器400上。人可阅读的形式存储在存储器或数据存储装置410中，且该代码链接到也存在于存储器装置420上的编译形式，装置420与工程服务器关联。因为工程服务器可以通过因特网或企业内部互联网与PLC访问，因此在相同网络上可以保持另外一个服务器（物理上与工程服务器分离的服务器，或是物理上是工程服务器或PLC的一部分的虚拟服务器），例如一台自动控制服务器。所述自动控制服务器甚至也可以是PLC的一部分，例如当PLC包含嵌入式网络服务器时。复杂和鲁棒的仿真，监测和控制软件可以存在于所述自动控制服务器上，并且可以支持网络，以便通过在因特网或企业内部互联网上的客户装置访问。因为运行在PLC上的软件的程序代码的人可阅读的版本通过工程服务器可以获得，所以鲁棒仿真，诊断，控制，及监视软件包可方便地访问该人可阅读的代码，并提供运行这些软件包的有价值的信息。例如，因为人可阅读的代码容易访问，因而便于访问受控于PLC的各种现场装置的标记信息。因此本发明允许从车间到控制室的复杂和鲁棒的通信与数据传送。

如图5所示，保持具有集中式支持网络的工程工具的服务器给工程工具的开发者提供了一个发布新的服务和产品的平台。例如，在本发明之前，购买用于优化程序代码的仿真工具的费用阻止了许多客户在这些工具上的投资。在一实施例中，由于程序代码在中心服务器上存储和生成，这样可以被很多客户访问，因此在服务器上安装仿真软件的费用可以由很多客户分担。并且，在一实施例中，客户仅需为使用仿真工具付费。另外，诊断工具可以相同的方式在服务器上安装和共享。在有些实施例中，服务器和工具的所有者可以根据使用，为客户保存，或多种其它方案而得到补偿。

因为工具位于中心，因此工具的许可变得简单。工具可许可给公司，公司里的某些个人，或这些个人自己。可使用加密技术和/或密码保护来实现对工具的访问，并可以保持合法的许可为条件。在一些实施例中，可

能希望提供有限时段的免费访问工具。有些情况下，免费访问的量可与可编程控制器的购买联系起来。在有限时段提供免费的现场服务器对把服务器长期租赁给客户的销售也是有益的。在这个实施例中，给用户提供了服务器进行有限时段的免费试验。

在另一个实施例中，如图21中所示，公司可以选择租借或拥有它们自己的服务器。对于一些大型的制造公司和/或其它机构，安全是其主要关心的，限定只有公司雇员可访问服务器是有利的。因为工程工具位于服务器上而不是每位用户的PC，所以版本管理，更新和其它软件维护工作变得简单。此外，工程工具相对于用户编程装置上的操作系统独立工作。因为只有公司雇员可以访问服务器，所以专有信息可以存储在服务器上，并且可以被工作在不同地点的编程小组访问。

本发明也有利地提供了用于合作完成工程的系统和方法。在一个实施例中，多个用户可共同访问和开发用于一个或多个可编程控制器的程序代码。如图22所示，多个用户可访问一个服务器。支持网络的工程工具2200位于服务器50上，并且能够在客户装置30上的浏览器应用程序中运行。服务器可以包含存储器，用于在数据库2210以及配置管理器2220中存储工程项目。存储在项目数据库2210中的项目可以是分散的或是相互关联的。典型地，创建一个特定项目或一段代码的用户被赋予了对该项目的全部读和写的权力。如图22中所示，第一用户5000可以访问第一项目2241。一旦访问第一项目2241，第一项目的一份活动的拷贝2250就在服务器上创建，其中所述第一项目2241典型地是人可读形式的存储程序代码的文件。借助于支持网络的工程工具2200，第一用户5000可以修改第一项目2250的活动的拷贝。系统中的其它用户5001，5002，5003，可能需要查看项目1的拷贝。这是第二和第三用户处理相关项目时经常遇到的情况。因为一份活动的拷贝正处于工作进程中，使得有可能第一用户5000根本无法保存他对该活动的拷贝所作的修改，因此准许第二和第三用户有权读取该活动的拷贝是不利的，同时上述也不是典型的。与此不同，更好的经常是给予第二和第三用户5002，5003读取位于项目数据库中的归档拷贝的权力。在一些实施例中，有利的可以给第四用户5001提供对活动拷贝的读的权力或写的权力。

这在第四用户5001是一位客户服务代表、一名顾问或是用户的管理人员的情况下非常有用。访问存储在数据库中的项目的归档拷贝与访问活动拷贝一样，通常受控于配置管理器。在本发明的一个实施例中，一个项目创建者通过项目配置管理器2220给其它用户赋予权力。常常至少一个其它用户，例如超级用户或IT管理人员，具有赋予权力的能力。

在一个实施例中，如图22中描述的一个例子，一个大型的编程任务被划分成几个小的项目。每个项目在项目数据库中作为文件存储。每个用户主要对一个单独的项目负责。例如第一用户5000负责项目1, 2241。这样第一用户5000，通过配置管理器，被赋予对第一项目2241读和写的权力。其它用户，如第二和第三用户5002、5003可能负责其它项目或相关项目，并因此被赋予对项目1的较少的权力，例如只读的权力。第二用户5002的任务是创建项目2, 2242，并因此被赋予对项目2的全部权力，但是用户1和用户3仅被给予了对项目2的一份归档拷贝读的权力。用户3, 5003的任务是创建项目3, 2243，并因此被赋予了对项目3的读写的权力，但是对项目1和2的归档拷贝只有只读的权力。同样，第一用户5000被赋予对项目2和3的文档拷贝的只读权力。一个项目超级用户，例如第四用户5001，具有管理权，包括有权对活动拷贝以及对文档拷贝进行读和写，有合并不同项目（活动拷贝或文档拷贝）成为一段程序代码的权力，有编译程序代码的权力，及有其它查看程序项目需要的权力。

本发明因此提出许多方法，用于对可编程控制器合作编程，及为I/O网络和装置生成配置数据，这些通常是编程过程的一部分。图26—28，概括示出了本发明提供的合作编程的方法和技术中的几个例子。虽然在这些图中示出了特例，但是这里概要示出的方法步骤只是举例说明，并未穷举本发明提供的广泛而鲁棒的合作工程方法。

本发明也提供了方便的程序代码存档和版本管理。版本管理器或版本管理系统可以合并到配置管理器或受控于配置管理器。在新的编译过的程序代码下载到可编程控制器（直接从服务器下载到控制器或通过客户装置从服务器下载到控制器）之前，一份现有代码的拷贝存档于服务器上。在一些实施例中，程序代码在其被编译之前以人可阅读的形式存档。编译后

的代码与人可阅读的代码可相互链接，这样人可阅读的代码可基于编译后的代码访问（见例如图20）。

位于服务器上的版本管理工具具有多种优点并为工程工具供应者提供了新的商业模式。许多优点中的其中之一是其允许客户查找程序的特殊变化或比较程序的两个版本。这是在食物和药品管理局（“FDA”）的行业规定中非常希望的。例如，在食品工厂，FDA要求对于PLC程序的全部变化都具有文档记录。版本管理工具可提供这种功能。在一些实施例中，版本管理工具是配置管理器的一部分并位于具有工程工具的服务器中。在一个典型的情况下，用户对PLC程序（例如版本1.0）进行修改，例如增加了混合物中糖的成分。修改后的程序与文件一起在服务器上存为新的版本（例如版本1.1）。如果在其后的某些时间点，产品确定为太甜，后面的用户可以通过查找PLC中影响甜度或增加到产品中糖的数量的变化对其进行修正。版本管理器优先地支持这种功能。通常，不可能恢复到原始版本（例如1.0），因为存在后续的变化，但是用户可以通过改变运行在该PLC上的已有版本创建一个新版本。

除了提供查找功能外，版本管理工具还可以提供比较功能，这允许将一个版本与另外一个版本进行比较。该比较功能可以工作于基于图形和文本的编程语言，且最好高亮显示变化。

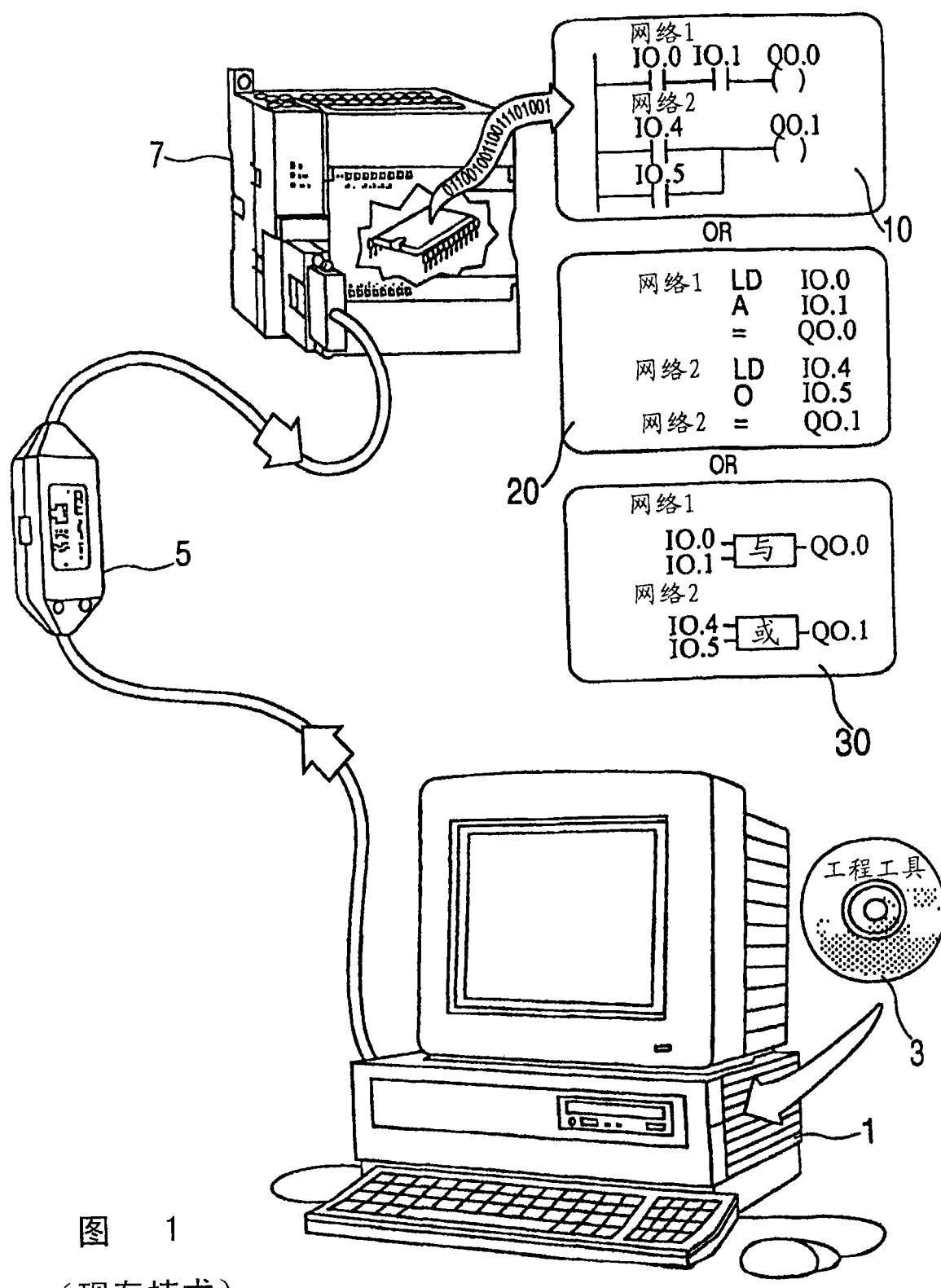
本发明也提供了用于测试在中心服务器上开发的程序代码的方法和系统。如图23所示和概述的，包括工程工具的服务器被客户装置访问，步骤9900。然后在客户装置上运行工程工具，步骤9905，优先地在客户装置的浏览器应用程序中运行，然后在服务器上创建程序代码，步骤9910。随后可以通过在服务器上运行仿真软件测试程序代码，步骤9915。该仿真软件可以模仿实际的可编程控制器。在其它实施例中，服务器可以与测试可编程控制器接口，步骤9920。优先地，测试可编程控制器配置为测试在服务器上用工程工具产生的程序代码和/或配置数据。程序代码测试完成后，可以被下载到实际的可编程控制器，步骤9925，这常常称为目标可编程控制器。

图24中概要示出了下载本发明生成的代码的一个方法。在服务器上生

成程序代码和/或配置数据，步骤9000。然后编译代码，步骤9005。代码然后被下载到客户装置，步骤9010。并拷贝到与客户装置关联的存储装置上，步骤9015。该存储装置可采用多种形式，并可以是可移动存储装置，例如一磁盘驱动器或一闪存单元。存储装置可与一电子装置接口或通信耦合，步骤9020。然后拷贝代码到该电子装置，步骤9025。然后代码从该电子装置被下载到目标可编程控制器，步骤9030。

如图25所示，本发明也可以包括一个配置编辑器9100，用于生成存在于服务器9200上的代码或数据，以配置I/O网络和装置。配置编辑器通常是工程工具9300的一部分，或是工程工具组合9350的一部分。本领域技术人员可以认识到多数情况下这里描述的工程工具可以包括配置编辑器，以及程序代码的生成可以包含配置数据的生成。配置编辑器可采用多种形式，但通常是图形形式，并优选地适于被客户装置9500通过网络例如因特网9600调用。图形配置编辑器可以显示I/O网络与在网络上的装置的布局。用于I/O网络和装置的配置数据典型地被下载到网络中的可编程控制器9700。在一些情况下，数据可以下载到网络中的智能装置。配置数据常常包括例如波特率等信息及有关将逻辑映射到物理I/O的信息。

虽然上面描述了一些具体的实施例，但是本发明也可以具体表示为多种不同的形式以致无法在这里全部描述，并且不应该受到上面描述的实施例的限制，上面描述的这些实施例用于对本发明进行说明但不是全部。例如，本领域技术人员也将认识到虽然上面以特定的次序描述了本发明的方法步骤，但是这些步骤的次序并不是关键的，并且本发明可以通过执行不同次序的步骤得以实施。例如，实施本发明时是否以特定的次序执行方法步骤常常是不重要的。例如，客户在把编译过的数据和配置数据从中心服务器下载到PLC之前或之后付费没有区别。



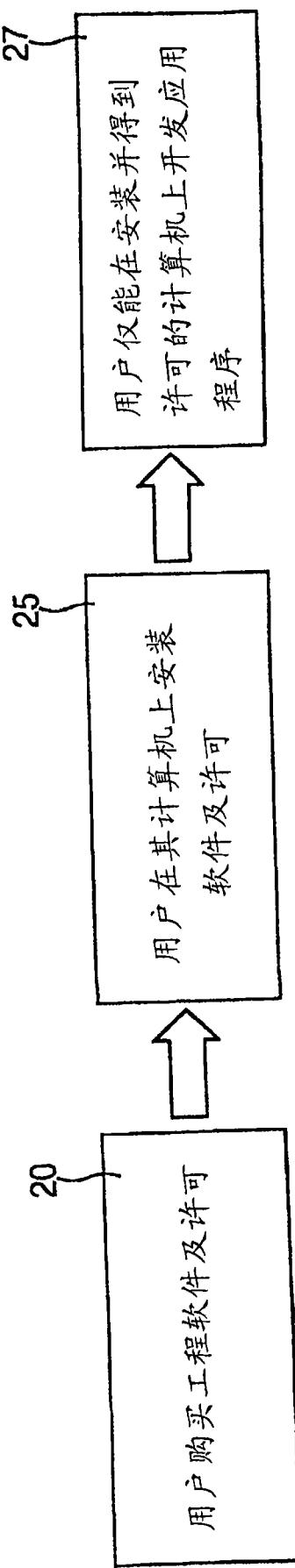


图 2
(现有技术)

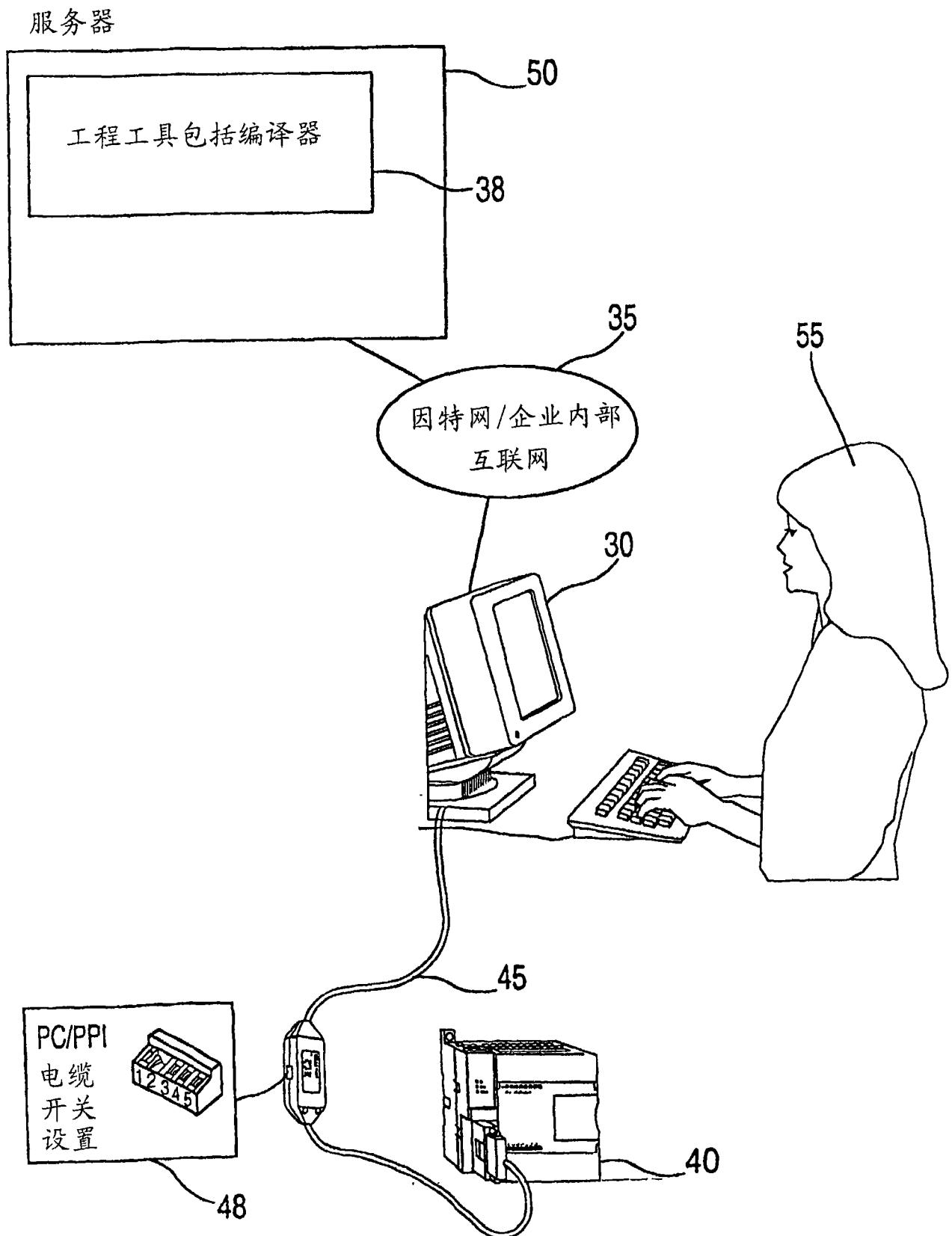


图 3

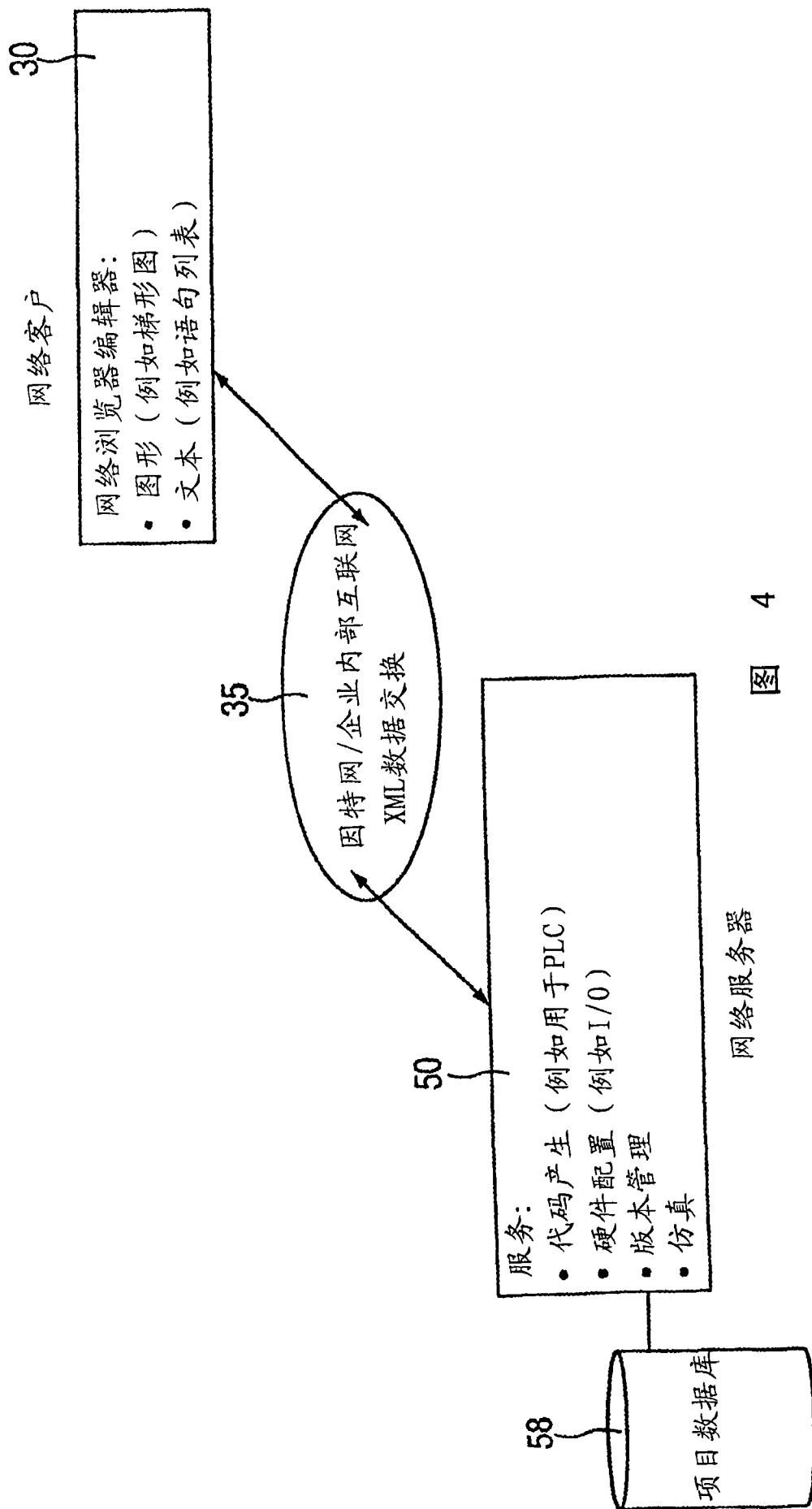


图 4

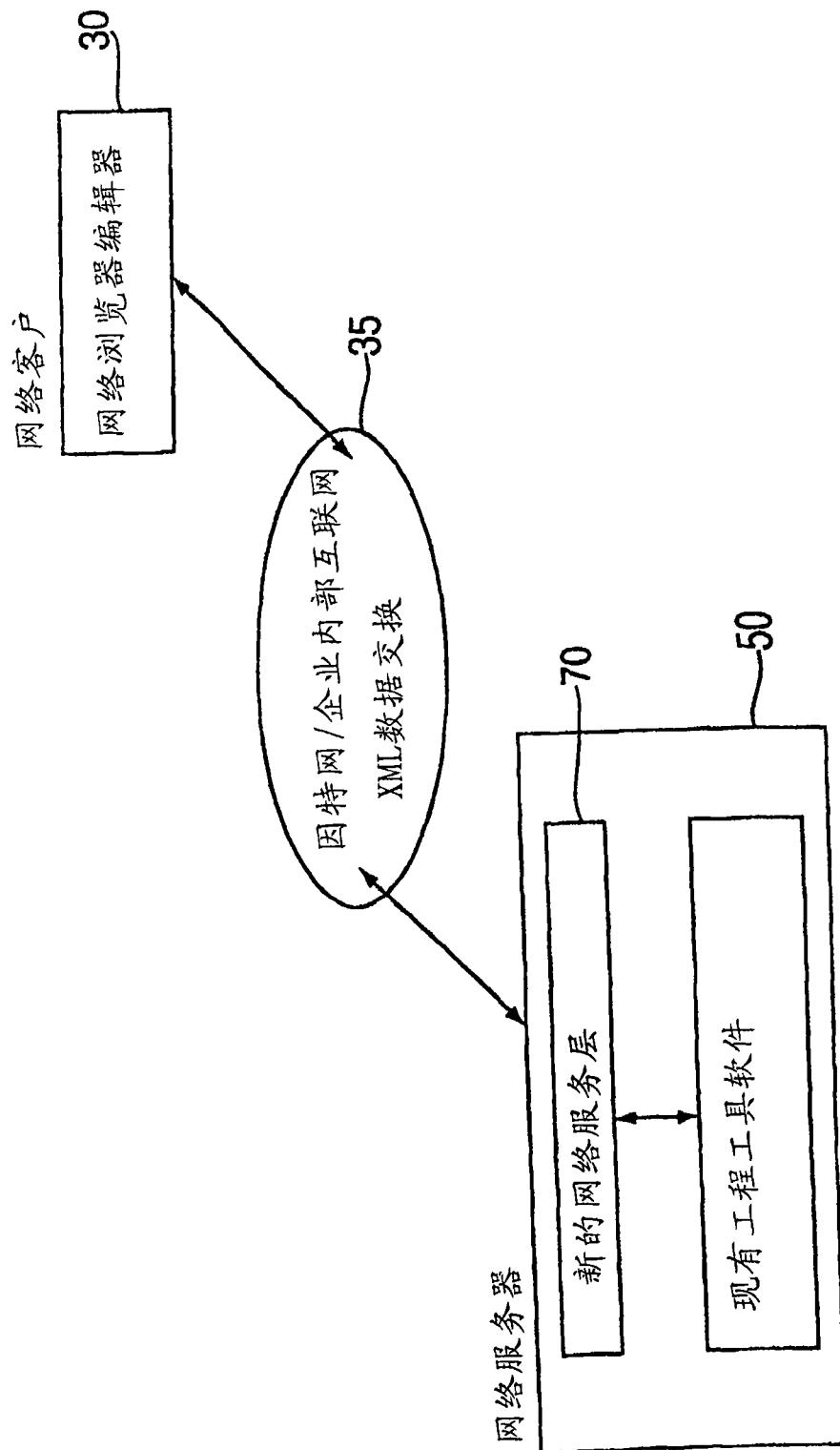
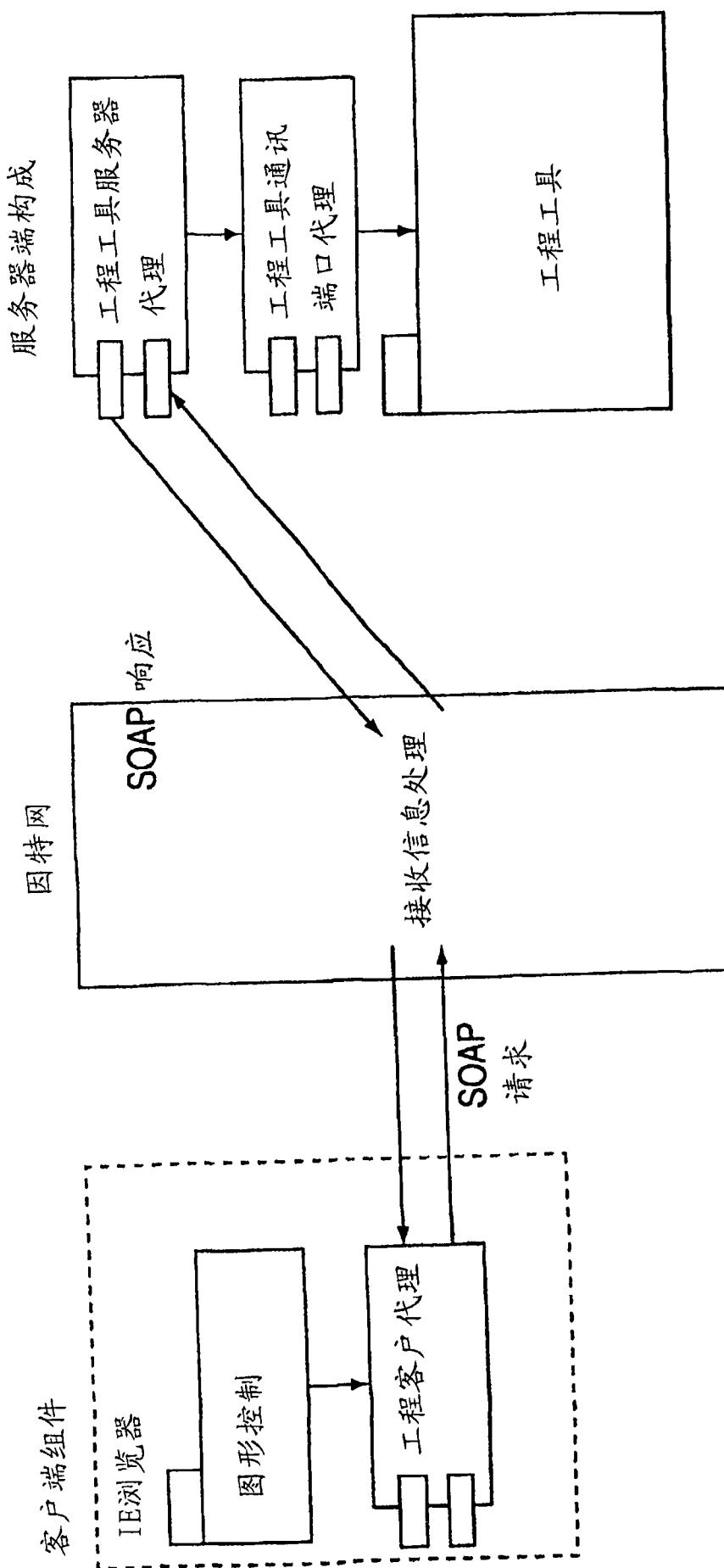


图 5



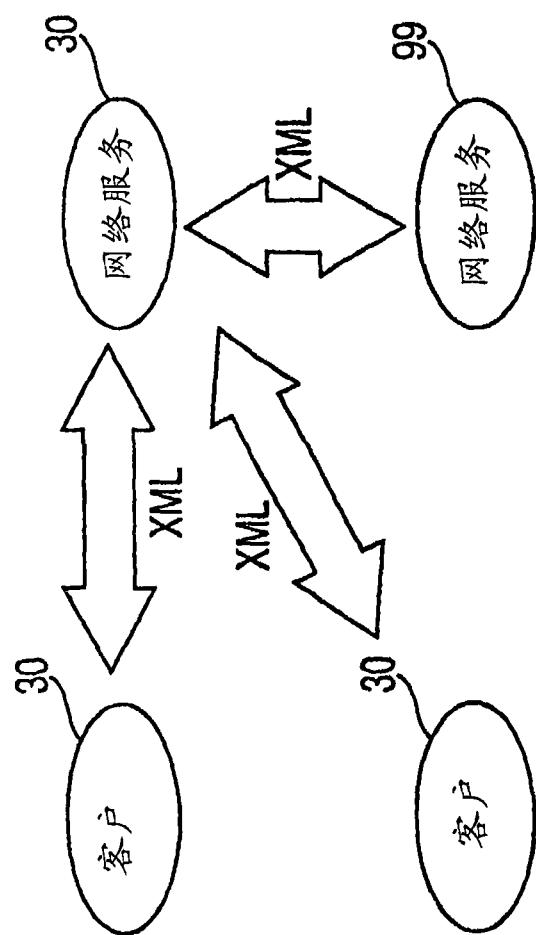


图 7

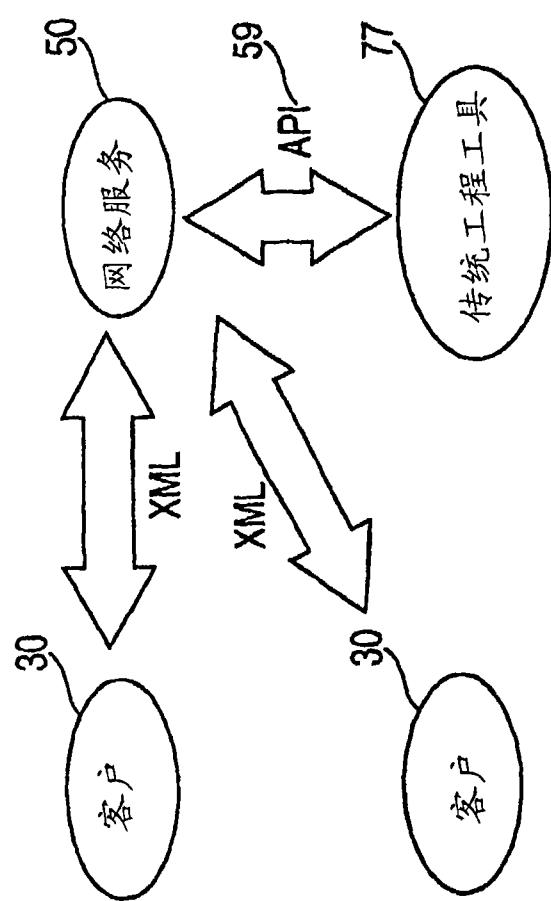
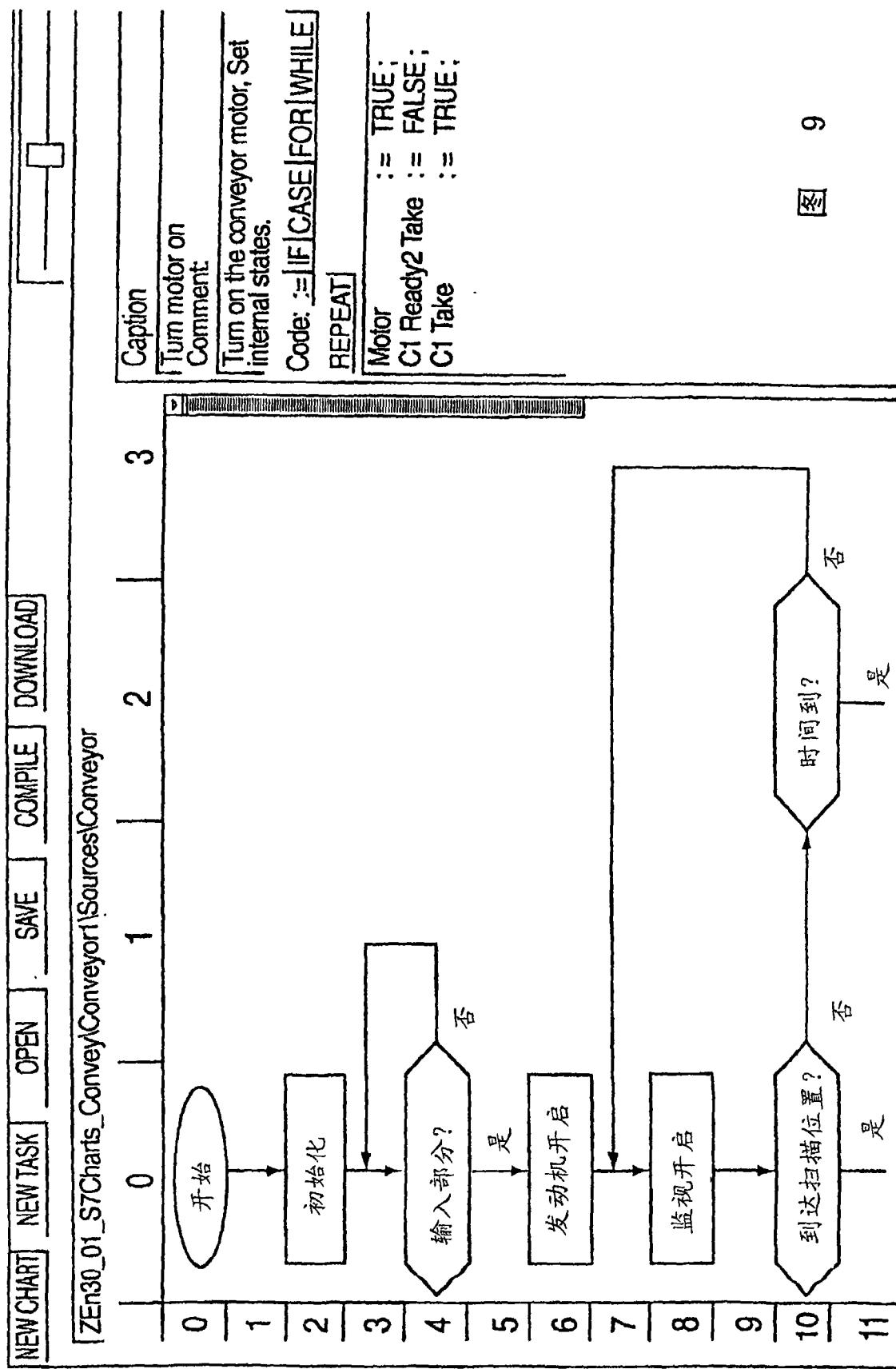
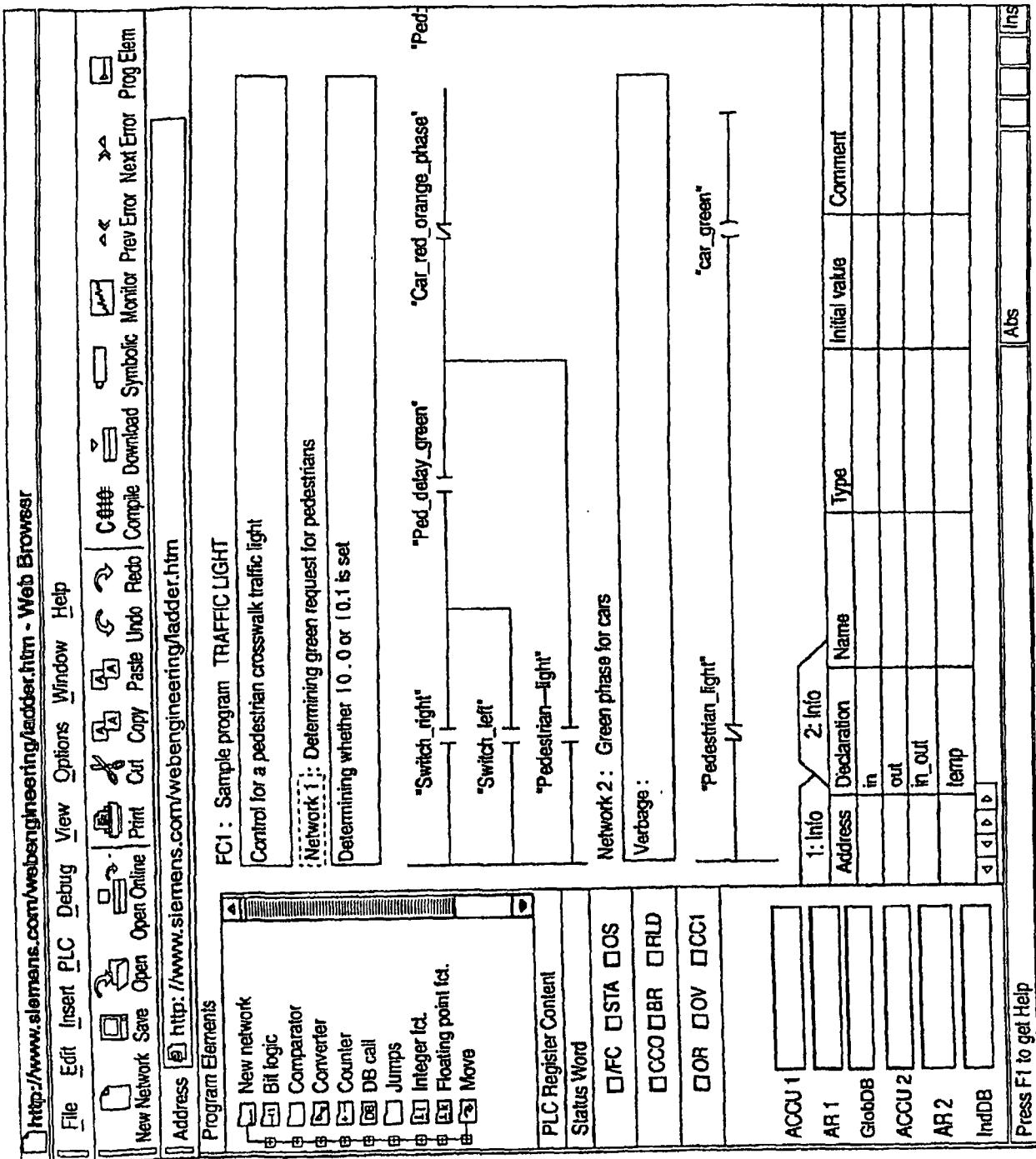


图 8





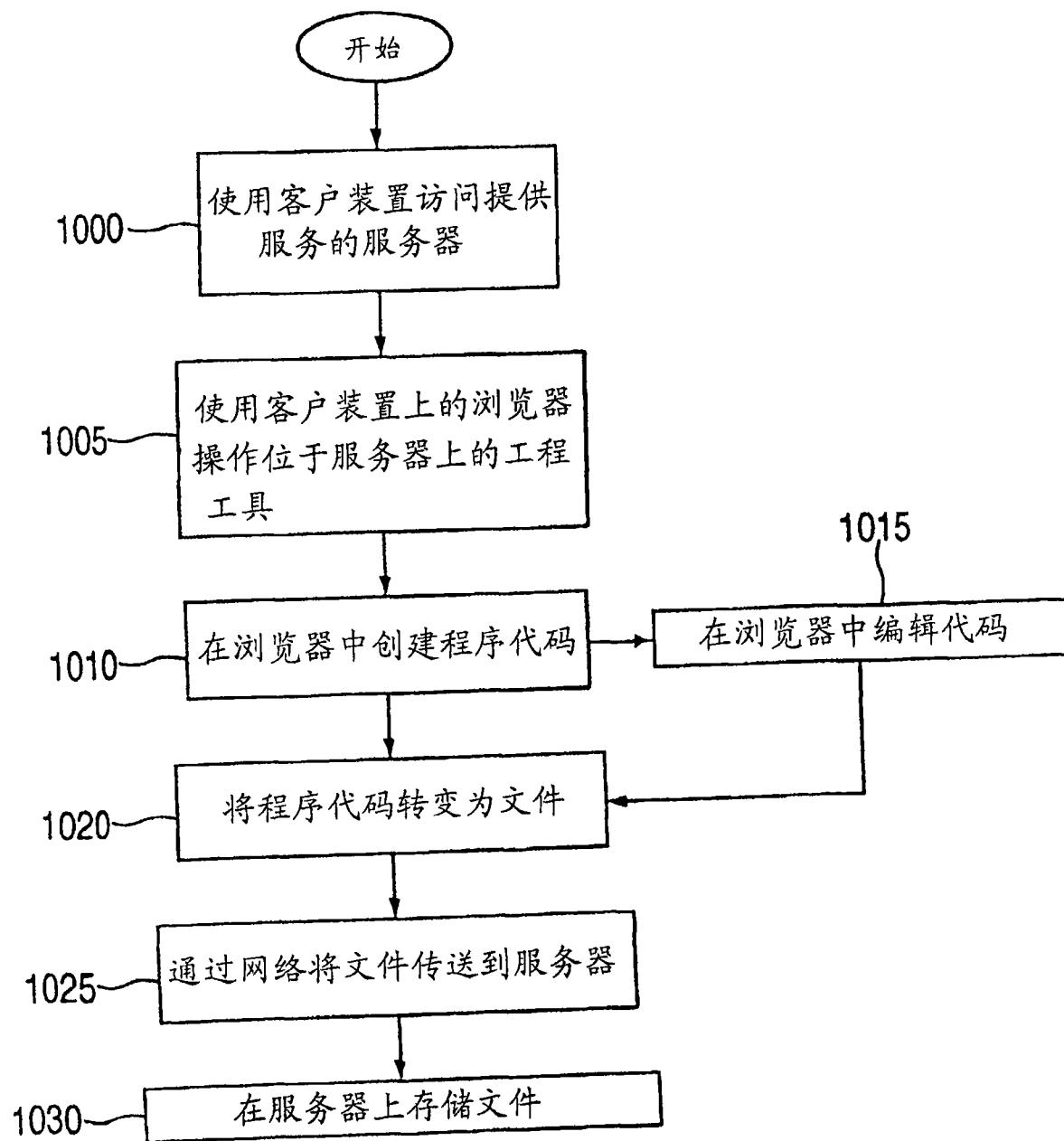


图 11

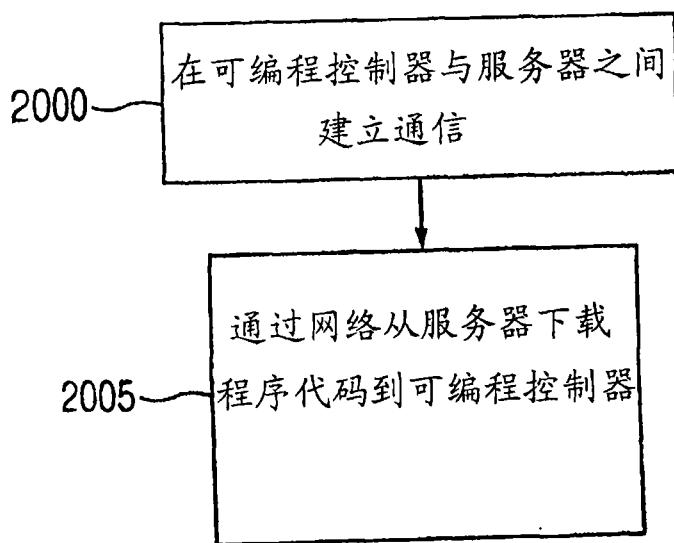


图 12

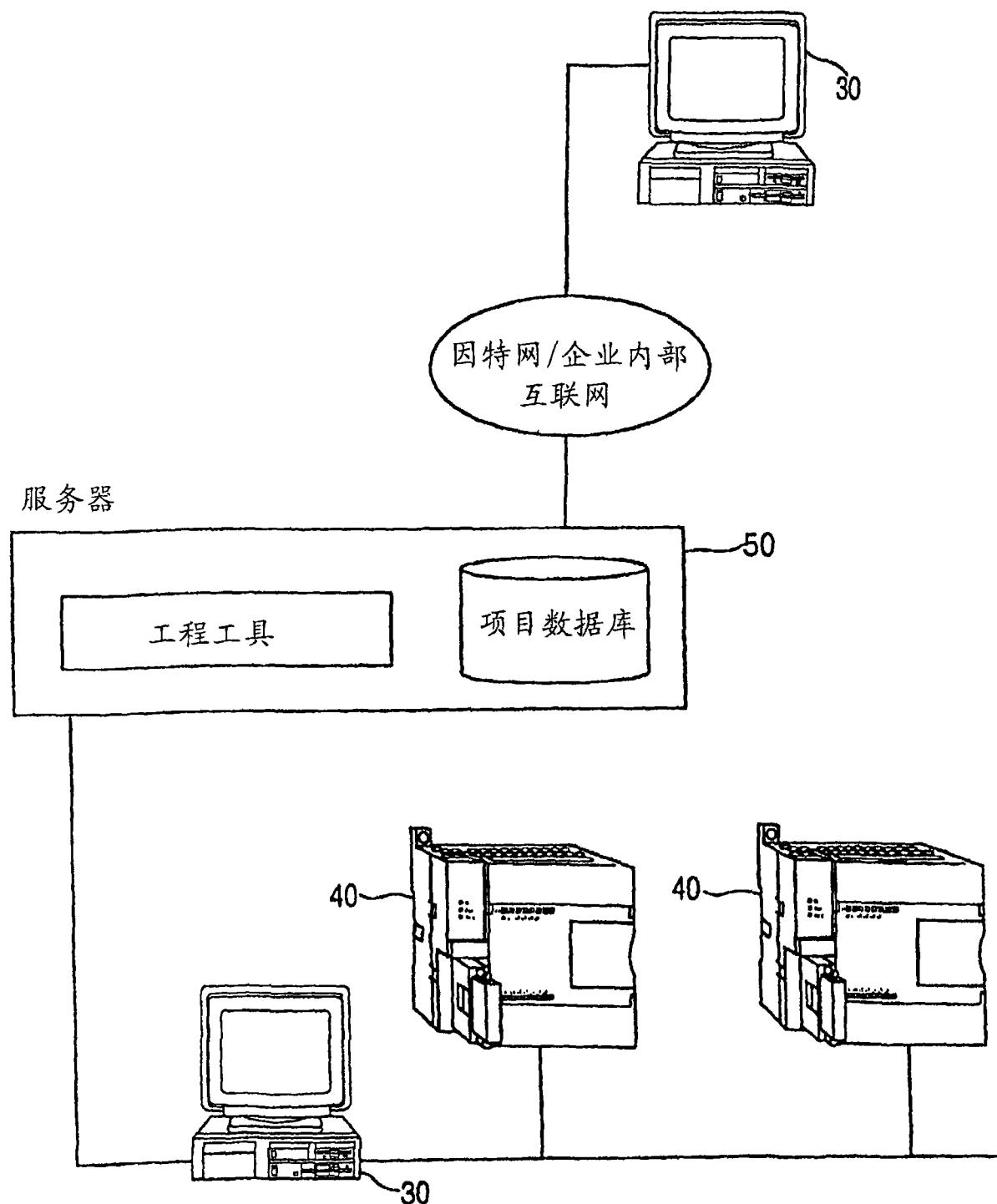


图 13

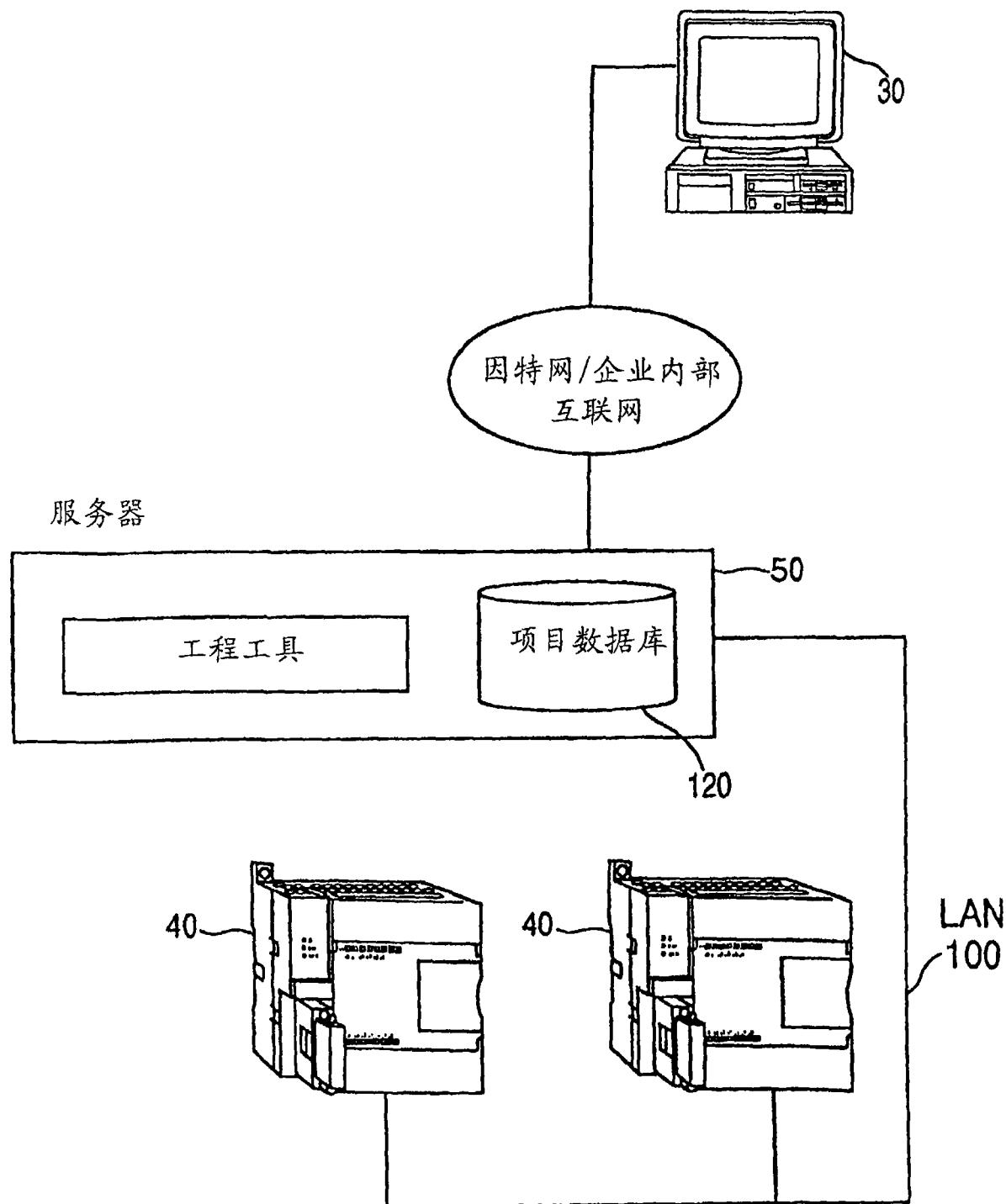


图 14

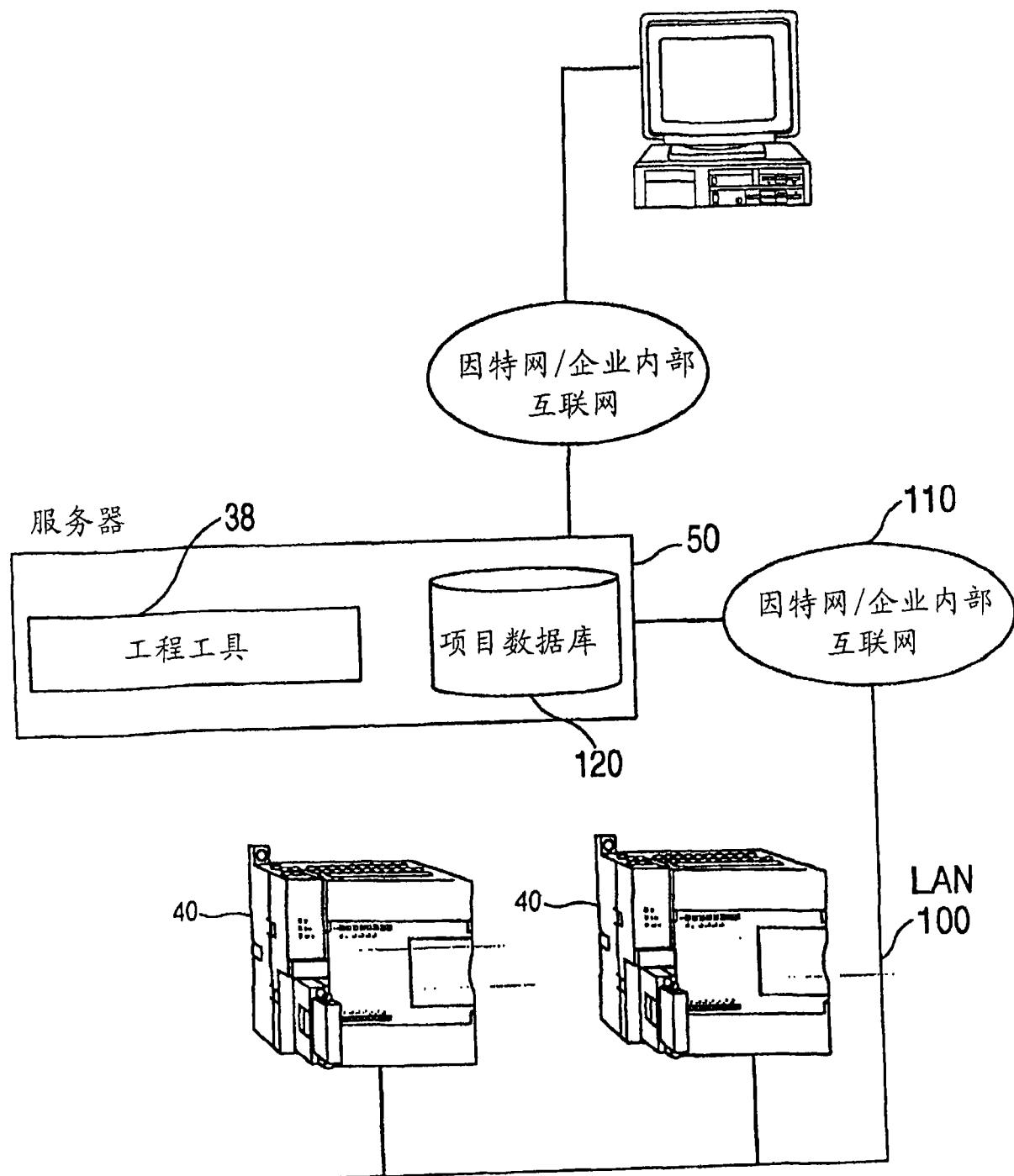


图 15

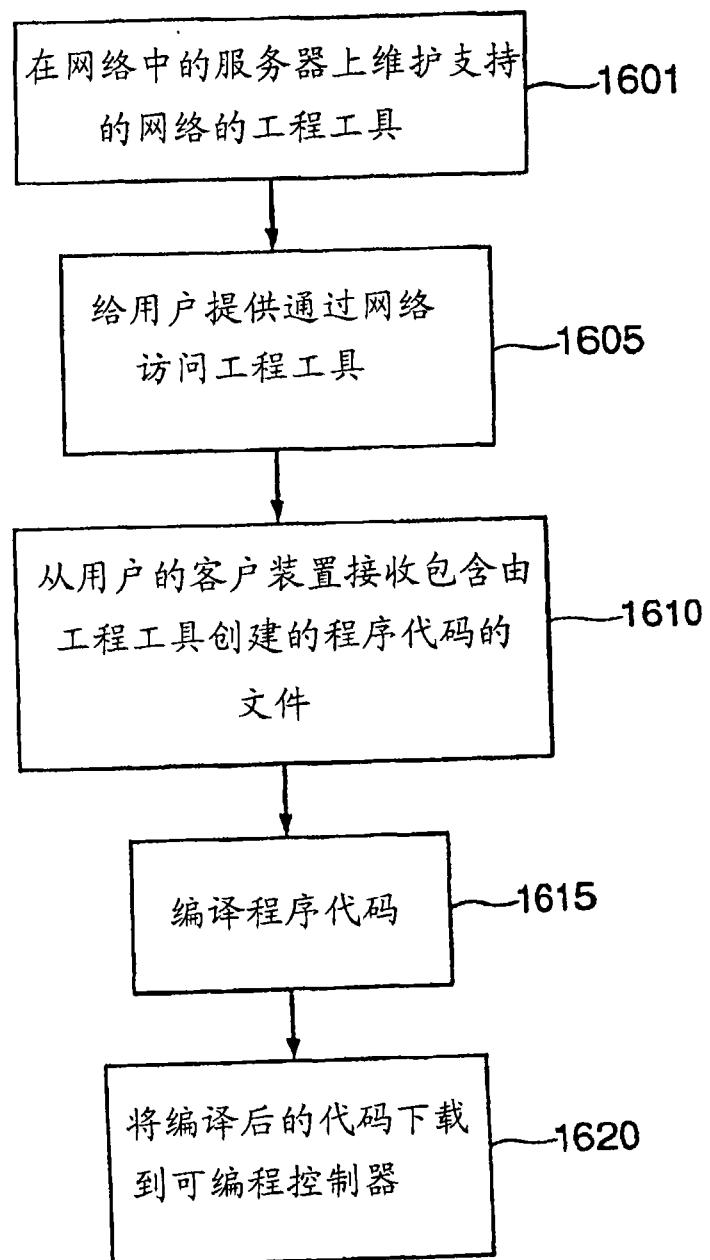


图 16

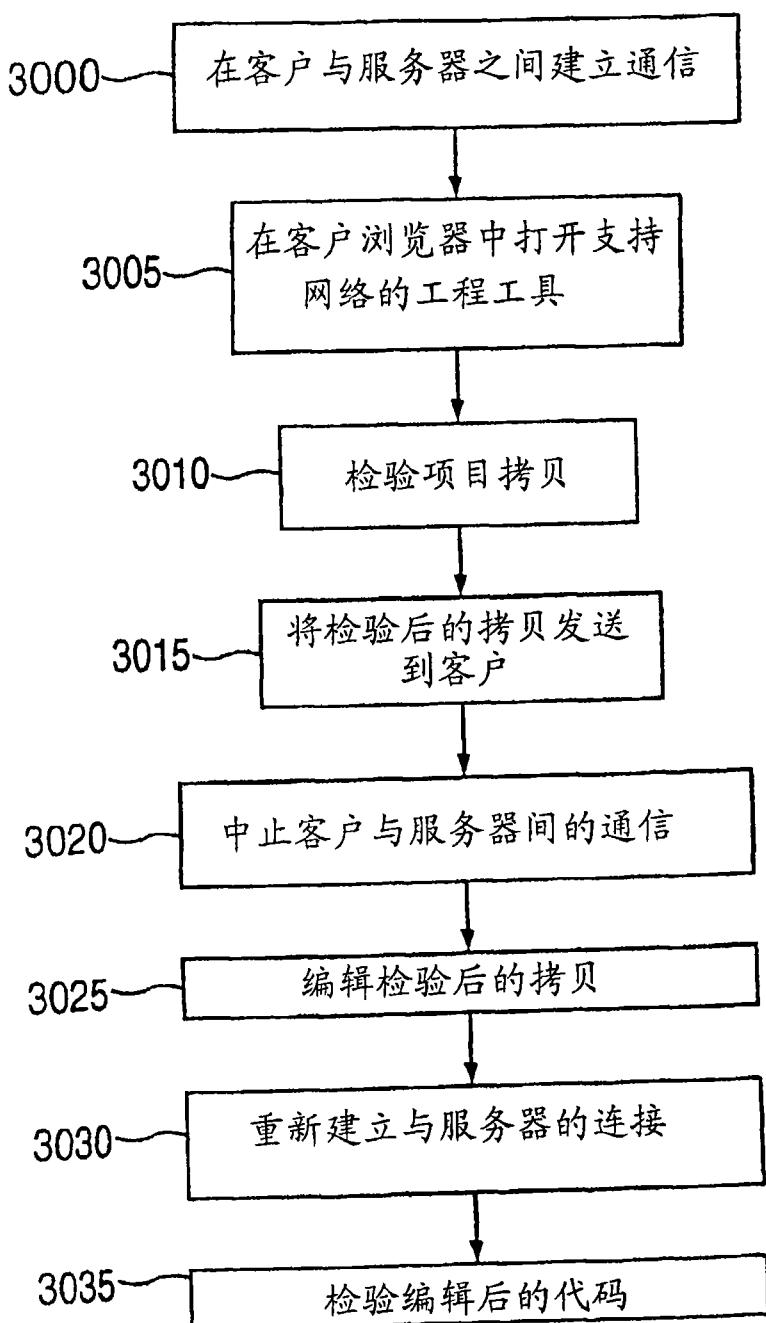


图 17

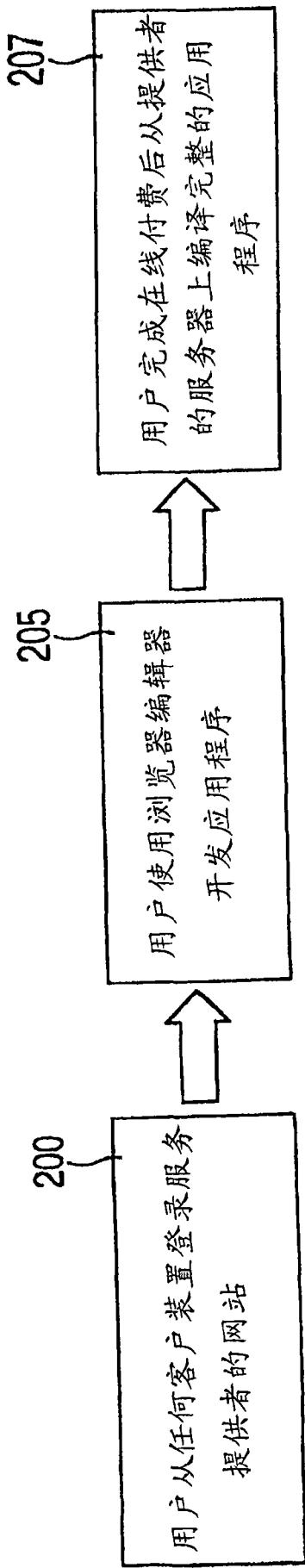


图 18

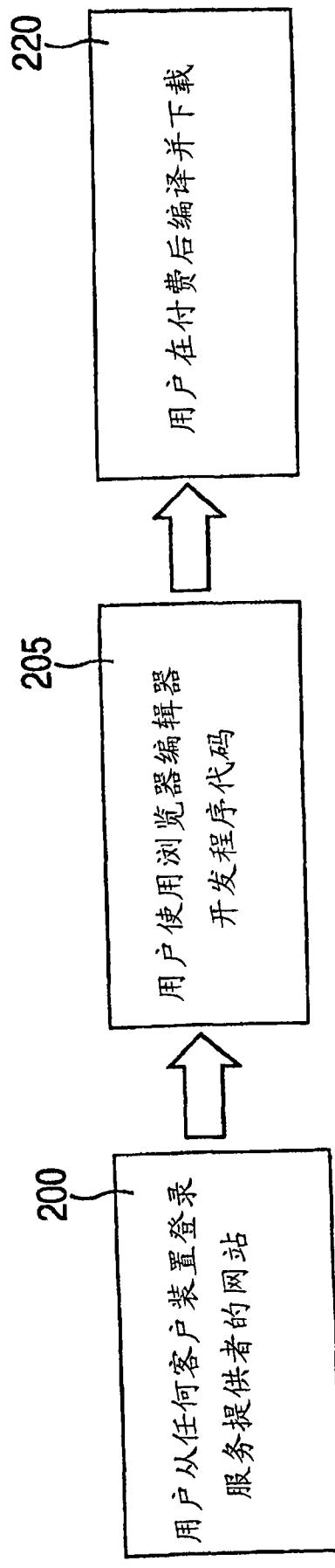


图 19

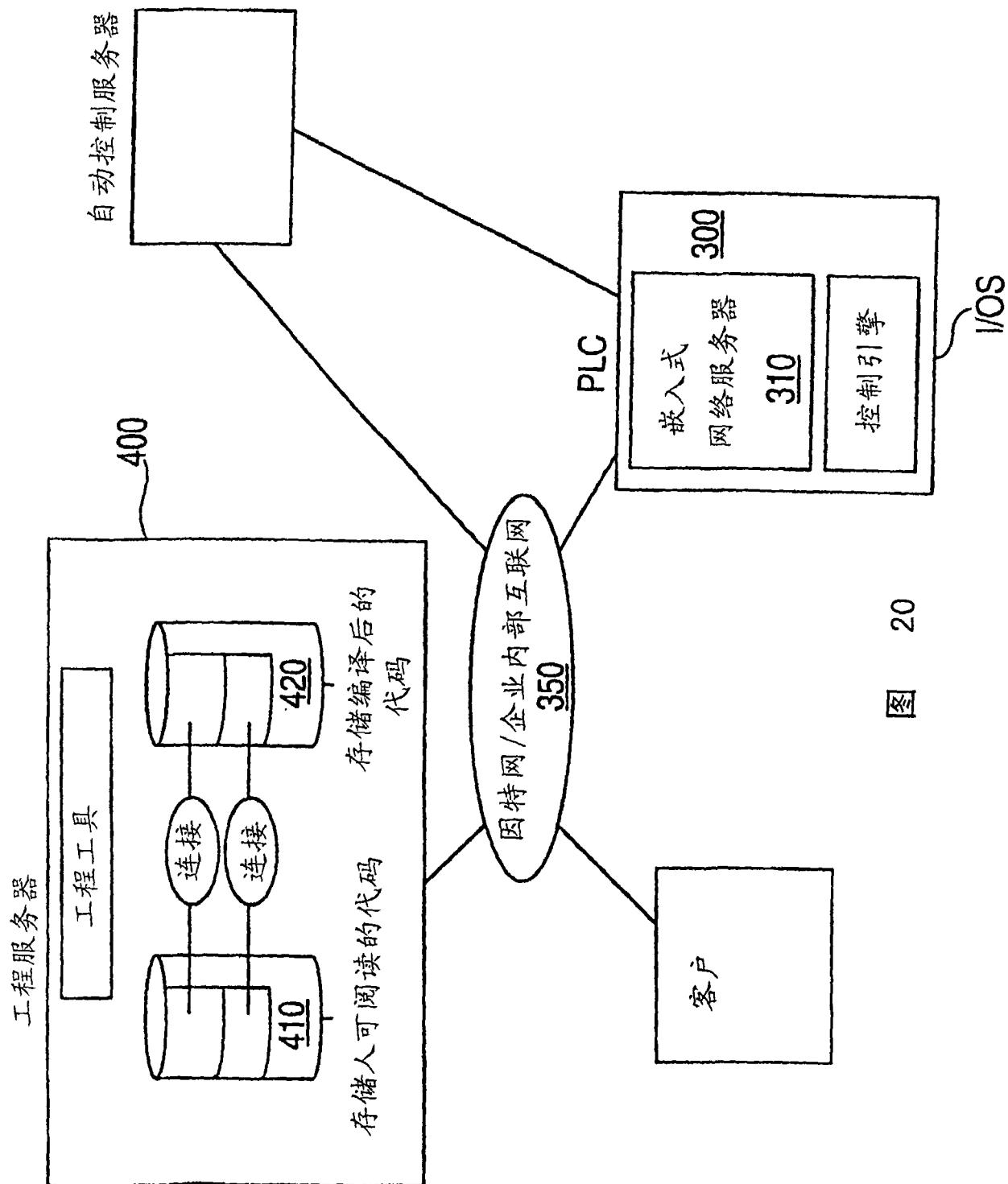
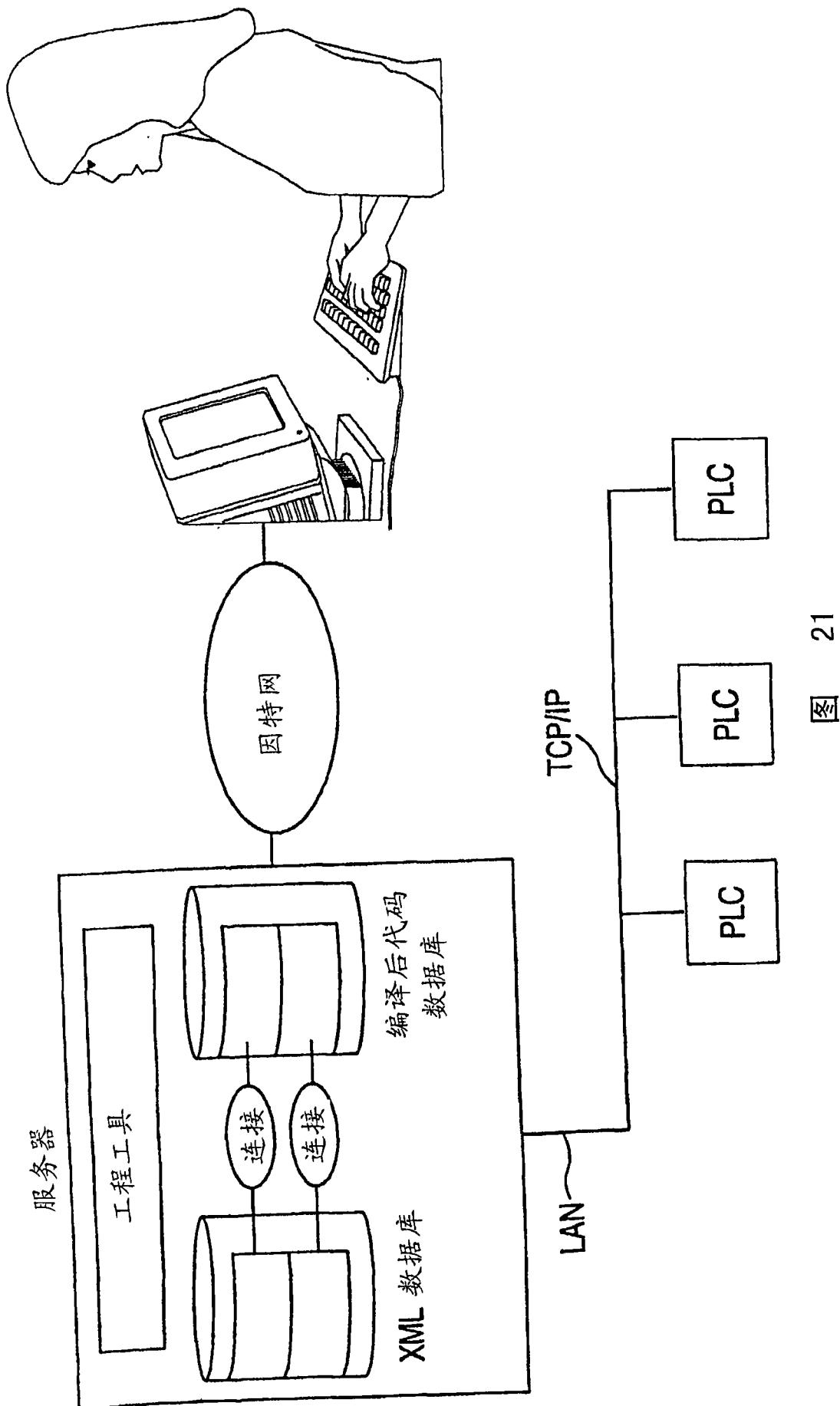


图 20



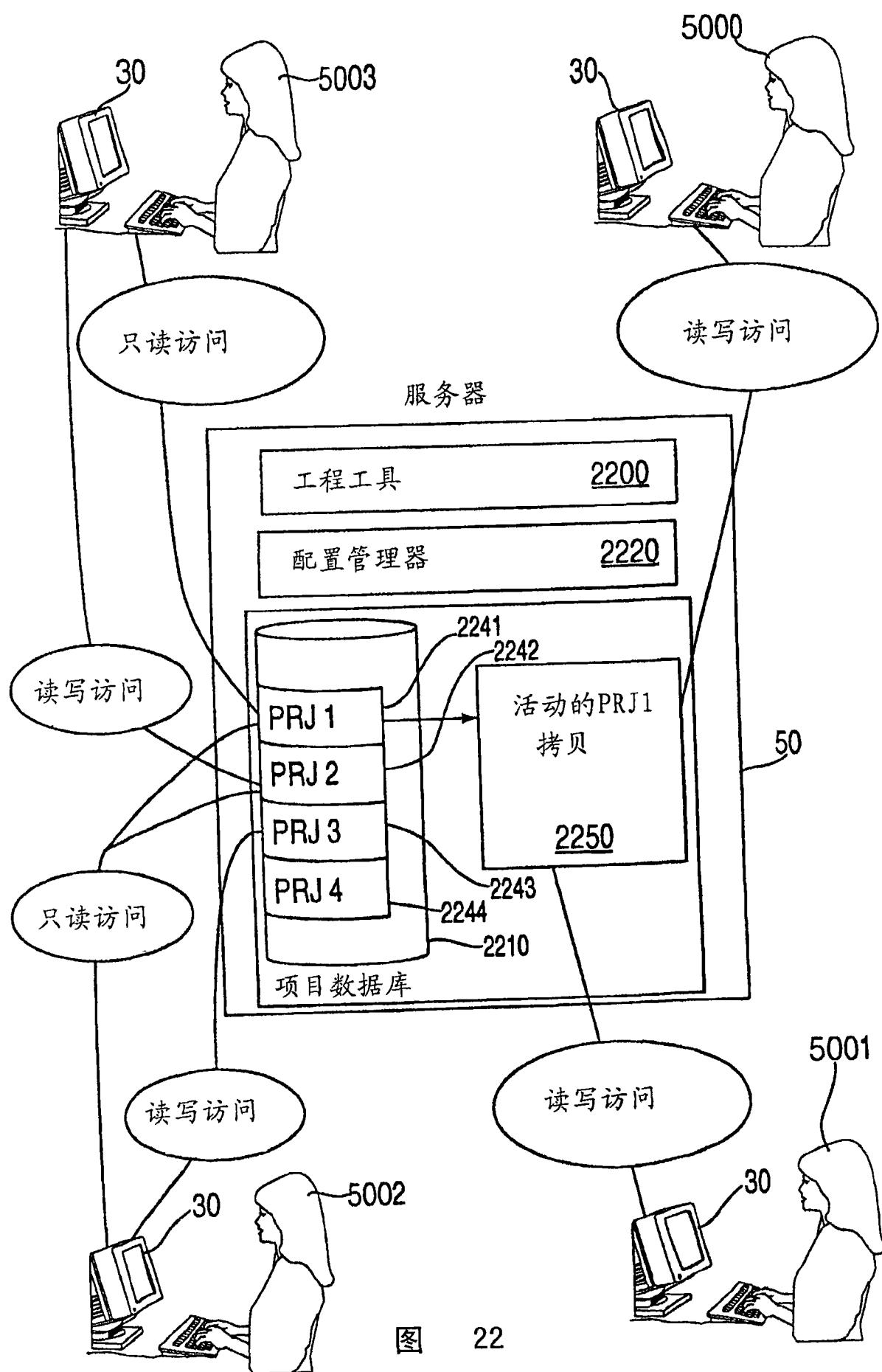


图 22

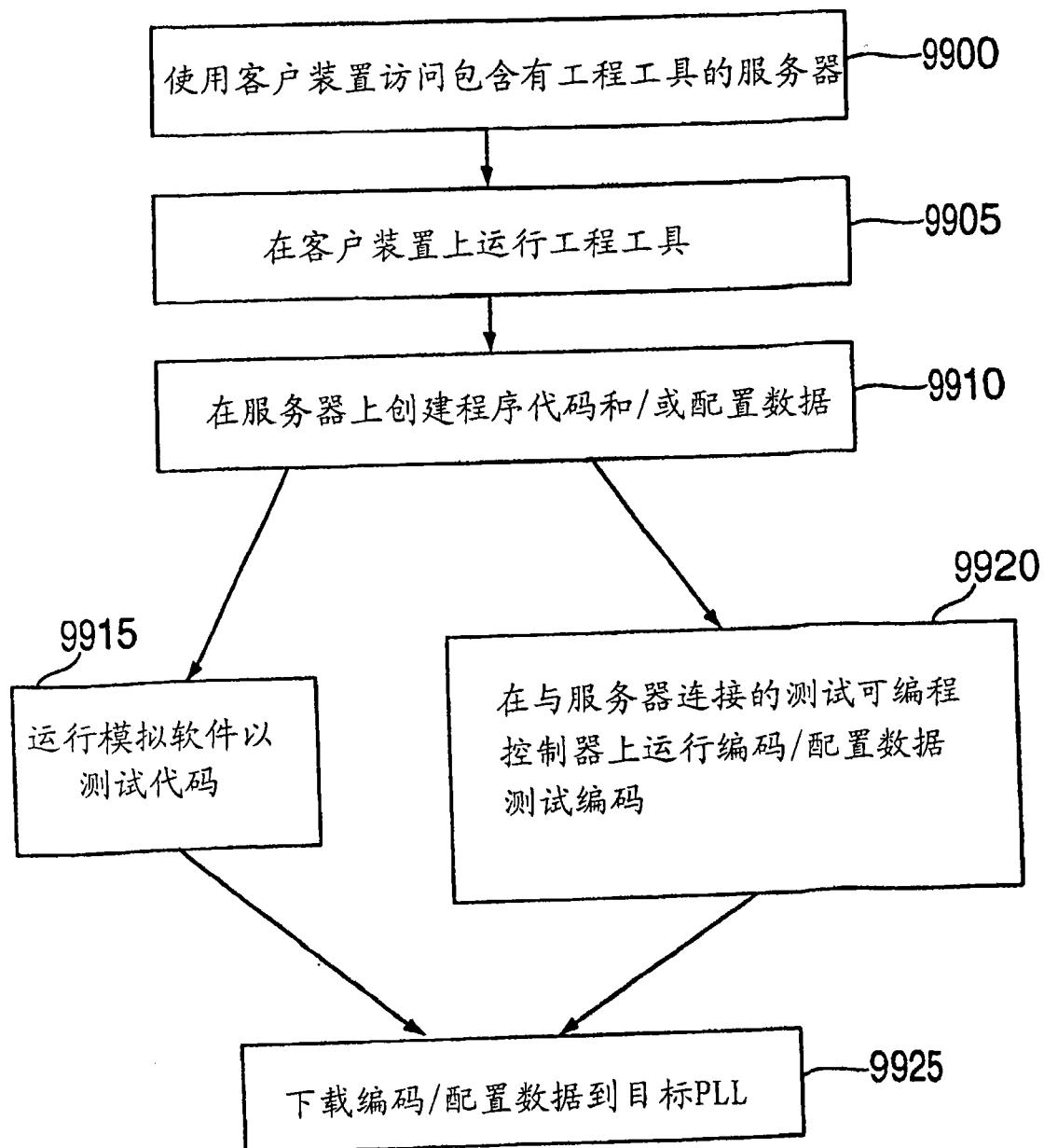


图 23

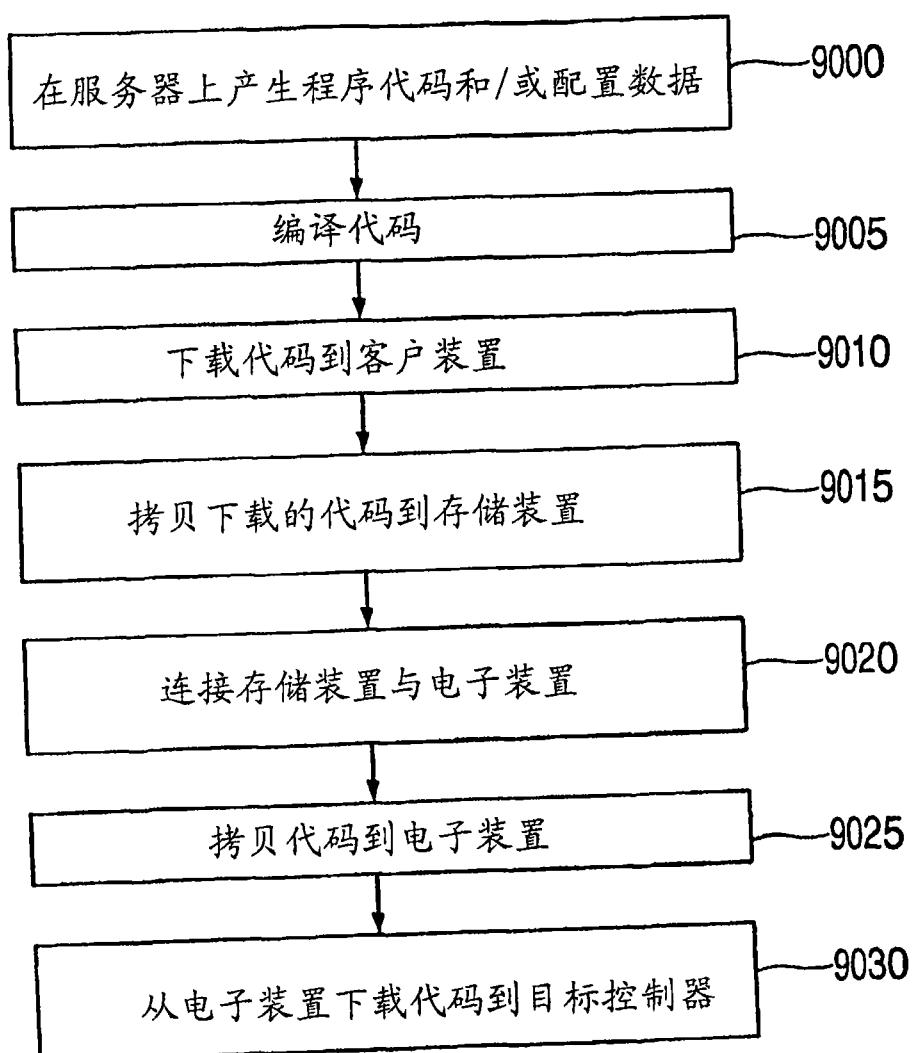


图 24

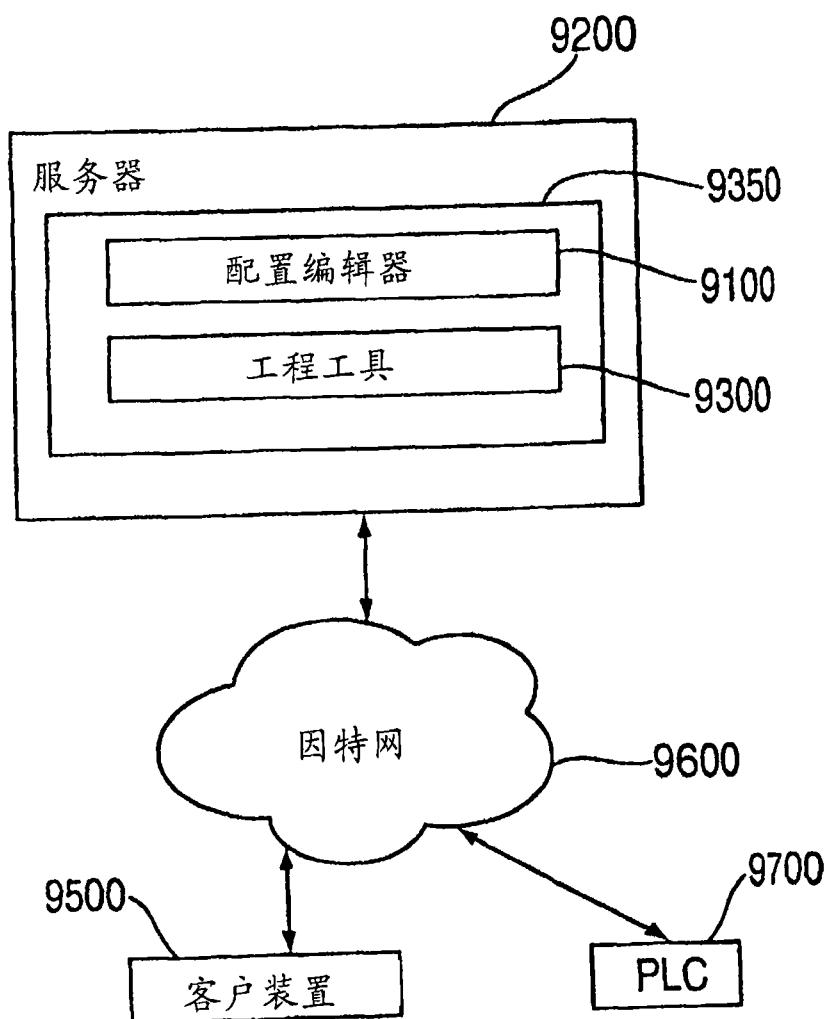


图 25

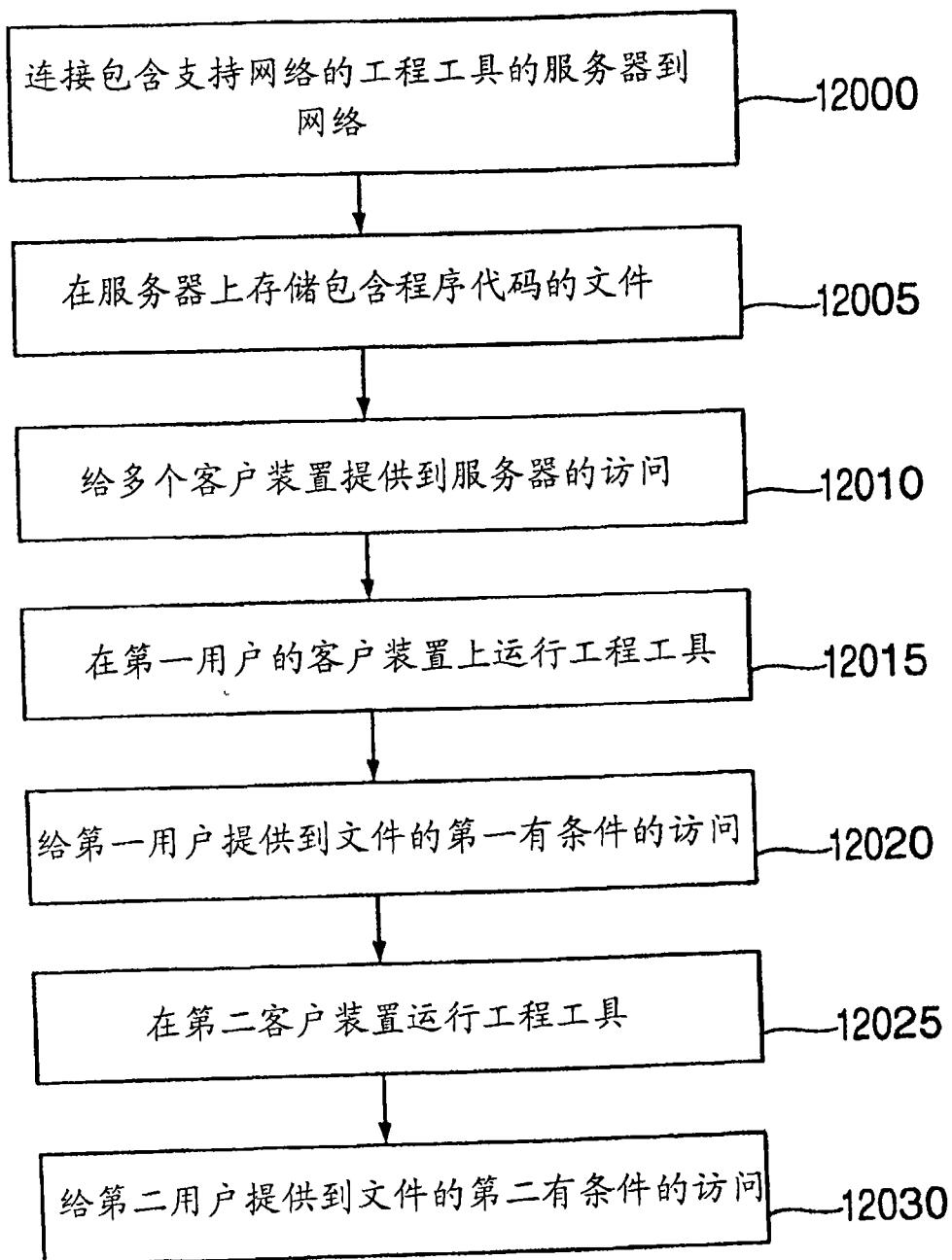


图 26

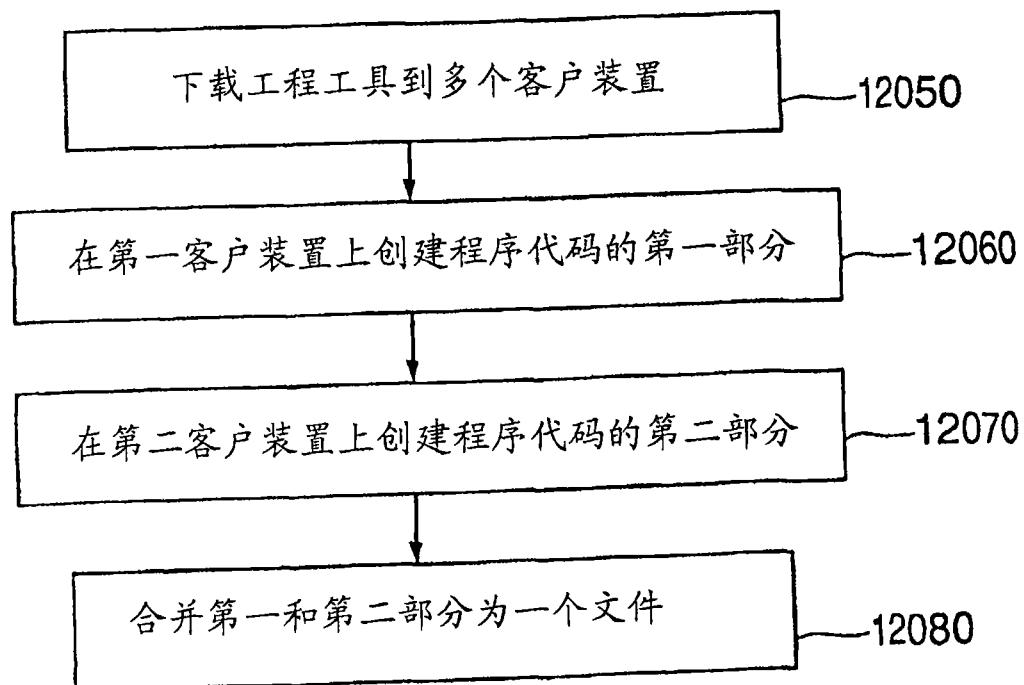


图 27

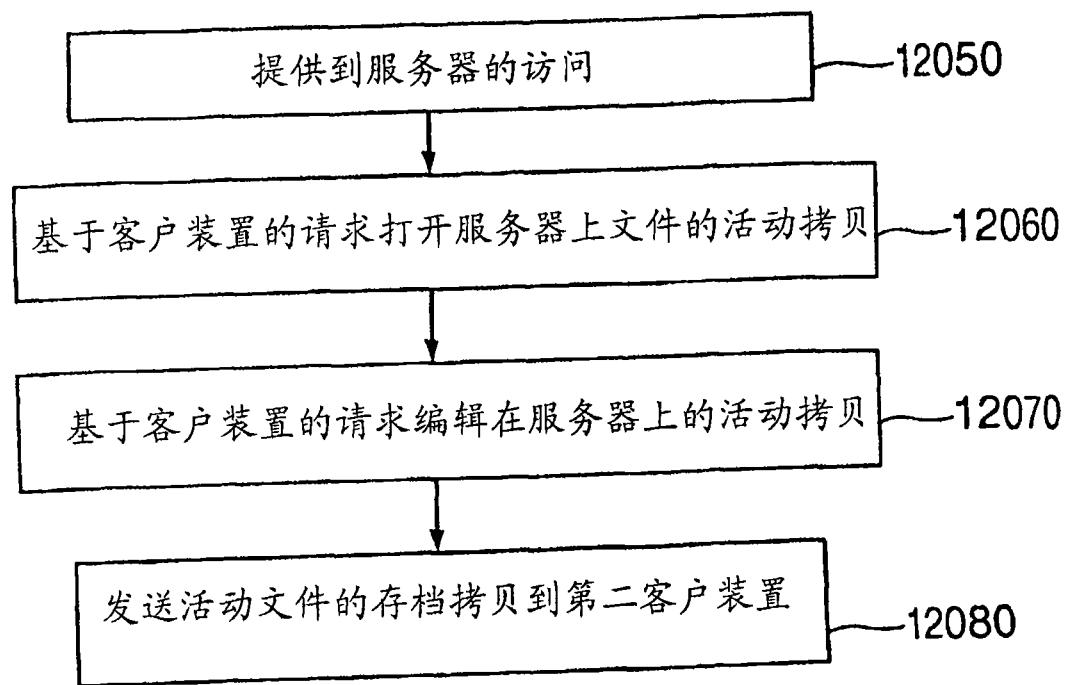


图 28