



(12)发明专利

(10)授权公告号 CN 103597464 B

(45)授权公告日 2017.06.09

(21)申请号 201280028934.5

(22)申请日 2012.06.12

(65)同一申请的已公布的文献号
申请公布号 CN 103597464 A

(43)申请公布日 2014.02.19

(30)优先权数据
13/159,139 2011.06.13 US

(85)PCT国际申请进入国家阶段日
2013.12.12

(86)PCT国际申请的申请数据
PCT/US2012/042104 2012.06.12

(87)PCT国际申请的公布数据
W02012/174022 EN 2012.12.20

(73)专利权人 微软技术许可有限责任公司
地址 美国华盛顿州

(72)发明人 C·鲁多夫 M·哈芒德

R·安德森 E·尼森 J·南那伽
A·殷加斯

(74)专利代理机构 上海专利商标事务所有限公
司 31100

代理人 陈小刚

(51)Int.Cl.
G06F 9/54(2006.01)

(56)对比文件
US 2008/0307043 A1,2008.12.11,
US 7555471 B2,2009.06.30,
US 2008/0040484 A1,2008.02.14,
US 2008/0307043 A1,2008.12.11,
CN 101873311 A,2010.10.27,
US 2006/0265662 A1,2006.11.23,
US 2008/0082604 A1,2008.04.03,

审查员 邹黎君

权利要求书2页 说明书13页 附图10页

(54)发明名称

将解释运行时应用适应到多个客户机的方法、系统及装置

(57)摘要

描述了将解释运行时引擎适应到多个客户机的技术。一种装置可包括被安排成执行web客户机的逻辑设备。该web客户机可包括客户机适配器,该客户机适配器用于检测针对客户机用户界面的用户事件、将与该用户事件相关联的对用户事件属性的改变发送给服务器应用、从服务器应用接收图形用户界面(GUI)无关的对象和经更新的用户事件属性、以及使用从服务器应用接收到的该GUI无关的对象和经更新的用户事件属性来更新客户机用户界面中的所渲染的图像。对其他实施例也予以描述并要求保护。

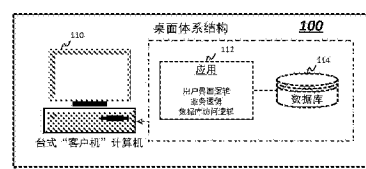


图 1A

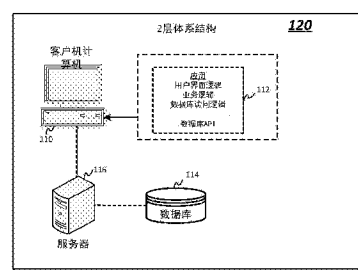


图 1B

1. 一种计算机实现的方法,包括:

接收影响在客户机设备上渲染的客户机用户界面中的用户界面元素的控制指示,其中所述控制指示由用户事件属性集合来定义;

将所述用户事件属性集合中的与所述客户机用户界面的呈现格式中的改变相对应的至少一个用户事件属性发送给在服务器上运行的服务器应用,所述服务器应用能响应于所述至少一个用户事件属性来执行脚本以生成图形用户界面 (GUI) 无关的对象;

处理从所述服务器应用接收到的具有经更新的用户事件属性的GUI无关的对象;以及

基于从所述服务器应用接收到的经更新的用户事件属性通过更新所述客户机用户界面的呈现格式来更新所述客户机用户界面中的渲染的图像。

2. 如权利要求1所述的计算机实现的方法,其特征在于,包括接收具有经更新的用户事件属性的GUI无关的对象,所述经更新的用户事件属性包括以下中的至少一个:具有一个或多个用户界面元素的对象元数据、或具有一个或多个元组的属性/值集合,其中每一元组包括用户界面元素的标识符、所述用户界面元素的属性、以及所述属性的值。

3. 如权利要求1所述的计算机实现的方法,其特征在于,包括在一消息中向在所述服务器上执行的服务器应用发送多个用户事件的多个已改变用户事件属性。

4. 如权利要求1所述的计算机实现的方法,其特征在于,包括基于从所述服务器应用接收到的所述GUI无关的对象和所述经更新的用户事件属性来更新所述客户机用户界面中所渲染的图像的一个或多个用户界面元素。

5. 如权利要求1所述的计算机实现的方法,其特征在于,包括:

在客户机适配器的先前实例及其相关联的包括GUI相关的对象的所渲染的图像已被解体时,创建客户机适配器的新实例;

将所述客户机适配器的新实例重新连接到所述服务器应用;

从所述服务器应用接收所有GUI无关的对象的最后已知状态;以及

在所述客户机适配器的新实例中同步从所述服务器应用接收到的所有GUI无关的对象的最后已知状态。

6. 一种计算机实现的系统,包括:

用于接收影响在客户机设备上渲染的客户机用户界面中的用户界面元素的控制指示的装置,其中所述控制指示由用户事件属性集合来定义;

用于将所述用户事件属性集合中的与所述客户机用户界面的呈现格式中的改变相对应的至少一个用户事件属性发送给在服务器上运行的服务器应用的装置,所述服务器应用能响应于所述至少一个用户事件属性来执行脚本以生成图形用户界面 (GUI) 无关的对象;

用于处理从所述服务器应用接收到的具有经更新的用户事件属性的GUI无关的对象的装置;以及

用于基于从所述服务器应用接收到的经更新的用户事件属性通过更新所述客户机用户界面的呈现格式来更新所述客户机用户界面中的渲染的图像的装置。

7. 一种计算机实现的装置,包括:

逻辑设备;以及

能在所述逻辑设备上操作的web客户机,所述web客户机包括客户机适配器,所述客户机适配器用于检测影响在所述装置上渲染的客户机用户界面中的用户界面元素的由用户

事件属性集合来定义的用户事件、将与所述用户事件属性集合中的与受影响的用户界面元素相关联的经改变的用户事件属性发送给服务器应用、从所述服务器应用接收具有经更新的用户事件属性的图形用户界面 (GUI) 无关的对象、使用从所述服务器应用接收到的GUI无关的对象和经更新的用户事件属性来更新所述客户机用户界面中的所渲染的图像、以及在用于渲染所述客户机用户界面的客户机适配器和包括GUI相关的对象的所渲染的图像已解体时,通过至少使用所述GUI无关的对象以与所述服务器应用同步来恢复所述客户机用户界面的渲染。

8. 如权利要求7所述的装置,其特征在于,所述经更新的用户事件属性包括对象元数据和属性/值集合,所述对象元数据包括一个或多个用户界面元素,并且所述属性/值集合包括一个或多个元组,每一元组包括用户界面元素的标识符、所述用户界面元素的属性、以及所述属性的值。

9. 如权利要求7所述的装置,其特征在于,所述客户机适配器包括以下各项中的至少一个:用于控制所述客户机用户界面的用户界面管理器、或用于基于从所述服务器应用接收到的GUI无关的对象和经更新的用户事件属性来更新所渲染的图像的渲染引擎。

10. 如权利要求7所述的装置,其特征在于,所述web客户机用于:

在所述客户机适配器的先前实例及其相关联的包括一个或多个GUI相关的对象的所渲染的图像已被解体时,创建所述客户机适配器的新实例;其中所述客户机适配器的新实例用于重新连接到所述服务器应用、从所述服务器应用接收所有GUI无关的对象的最后已知状态、以及在所述客户机适配器的新实例中同步从所述服务器应用接收到的所有GUI无关的对象的最后已知状态。

将解释运行时应用适应到多个客户机的方法、系统及装置

[0001] 背景

[0002] 客户机-服务器体系结构是为在两个基本实体(称为服务器和客户机)之间划分应用程序的计算任务或工作负荷的分布式应用结构。服务器是资源或服务的提供方。客户机是资源或服务的请求方。服务器是运行与客户机共享其资源的一个或多个服务器程序的物理或逻辑设备。客户机是通常不共享其任何资源而是从服务器请求内容或服务功能的物理或逻辑设备。客户机和服务器通常在单独的硬件上通过计算机网络进行通信。然而,在某些情况下,客户机和服务器两者可驻留在同一系统中。客户机因此发起与等待传入请求的服务器的通信会话。

[0003] 客户机-服务器体系结构的一种形式是多层体系结构,通常被称为n层体系结构。n层体系结构是其中应用程序的某些方面被分解成多个层的客户机-服务器体系结构。例如,使用中间软件在用户和数据库之间对数据请求进行服务的应用采用多层体系结构。n层应用体系结构向开发人员提供用于创建灵活且可重用的应用的模型。通过将应用分解成多个层,开发人员只需修改或添加特定层(或层次),由此避免了重写整个应用的需要。

[0004] n层体系结构在开发和修改应用程序时提供了许多优点。然而,在针对其中存在大量客户机的基于web的环境来实现n层体系结构时存在困难。每一客户机可能利用不同的web技术,包括不同的web浏览器、web服务以及web应用。此外,web技术被设计成与许多不同类型的底层硬件和软件体系结构一起工作,包括具有不同输入/输出(I/O)组件、形状因子、功率要求、处理能力、通信能力、以及存储器资源等的各种设备。因此,可能难以跨这些许多异构设备和体系结构来实现一个或多个层。此外,应用程序的web版本可能与应用程序的非web版本不兼容,由此造成了对各自的分开的软件体系结构的需求。本发明的改进正是针对这些和其他缺点而被需要的。

[0005] 概述

[0006] 提供本概述是为了以简化的形式介绍将在以下详细描述中进一步描述的选择的概念。本概述并非旨在标识所要求保护的主题的关键特征或必要特征,也不旨在用于帮助确定所要求保护的主题的范围。

[0007] 各实施例一般涉及适用于执行不同类型的应用程序(诸如,例如商用业务线应用程序)的客户机-服务器体系结构。一些实施例尤其涉及具有应用程序的多个层(或层次)的n层客户机-服务器体系结构,包括至少一个呈现层。在一个实施例中,例如,3层客户机-服务器体系结构可包括使用以下技术来实现的呈现层:该技术被设计成在适应解释运行时引擎应用以与许多不同类型的客户机一起操作时,分隔并增强用户事件的图形用户界面(GUI)渲染。

[0008] 在一实施例中,例如,一种装置可包括被安排成执行web客户机的逻辑设备。该web客户机可包括客户机适配器,该客户机适配器用于检测针对客户机用户界面的用户事件、将与该用户事件相关联的经改变的用户事件属性发送给服务器应用、从服务器应用接收图形用户界面(GUI)无关的对象和经更新的用户事件属性、以及使用从服务器应用接收到的该GUI无关的对象和经更新的用户事件属性来更新客户机用户界面中的所渲染的图像。对

其他实施例也予以描述并要求保护。

[0009] 通过阅读下面的详细描述并参考相关联的附图,这些及其他特点和优点将变得显而易见。应该理解,前面的概括说明和下面的详细描述只是说明性的,不会对所要求保护的各方面形成限制。

附图说明

[0010] 图1A示出了常规的台式机应用体系结构。

[0011] 图1B示出了常规的2层应用体系结构。

[0012] 图1C示出了常规的3层应用体系结构。

[0013] 图2示出了根据一个实施例的具有多个客户机和客户机适配器的增强式n层客户机-服务器体系结构的框图。

[0014] 图3示出了根据一个实施例的具有单个客户机和客户机适配器的增强式n层客户机-服务器体系结构的框图。

[0015] 图4示出了根据一个实施例的具有针对客户机和客户机适配器的图形用户界面(GUI)无关的对象的增强式n层客户机-服务器体系结构的框图。

[0016] 图5示出了根据一个实施例的增强式n层客户机-服务器体系结构的第一逻辑流程。

[0017] 图6A示出了根据一个实施例的GUI无关的对象的逻辑示图。

[0018] 图6B示出了根据一个实施例的特定GUI无关的对象的逻辑示图。

[0019] 图7示出了根据一个实施例的增强式n层客户机-服务器体系结构的第二逻辑流程。

[0020] 图8示出了根据一个实施例的适用于增强式n层客户机-服务器体系结构的计算体系结构的实施例。

[0021] 图9示出了根据一个实施例的适用于增强式n层客户机-服务器体系结构的通信体系结构的实施例。

[0022] 详细描述

[0023] 各个实施例一般涉及适用于执行不同类型的商用业务线应用程序的客户机-服务器体系结构。一些实施例尤其涉及增强式n层客户机-服务器体系结构,其中n是表示任何正整数的变量。增强式n层体系结构可包括应用程序的多个层(或层次),包括至少一个呈现层。在一个实施例中,例如,增强式n层体系结构可被实现为包括呈现层、应用处理层、和数据管理层的3层体系结构。呈现层一般实现用户界面逻辑,诸如处理输入/输出操作。应用处理层一般实现应用或业务逻辑,诸如根据一组应用规则来处理数据。数据管理层一般实现数据存储和访问,诸如定义数据模式、存储数据、处理数据查询,等等。

[0024] 增强式n层客户机-服务器体系结构可包括使用以下技术来实现的呈现层:该技术被设计成便于使用解释运行时引擎来分隔并优化应用中的GUI渲染和用户事件。它允许使解释运行时引擎应用从基于2层客户机-服务器的体系结构适应到被托管的3层环境,同时减少对该解释运行时引擎应用的改变。

[0025] 图1A、1B和1C以背景的方式示出了应用开发的三个常规体系结构,以突出增强式n层客户机-服务器体系结构的各个实施例的优点。图1A示出了常规的桌面体系结构。图1B示

出了常规的2层体系结构。图1C示出了常规的3层(或n层)体系结构。

[0026] 图1A是桌面体系结构100的示例,其中,应用程序112的所有部分(或应用层)是在客户机计算机110(例如,台式计算机)上实现的。应用程序112可包括实现例如用户界面(UI)逻辑、业务逻辑和数据访问逻辑的各种应用层。应用程序112可存储应用数据,并从也被实现在客户机计算机110上的数据库114中访问应用数据。

[0027] 图1B是2层体系结构120的示例,其中数据库114现在处于客户机计算机110的远程。在2层体系结构120中,应用程序112及其各组成应用层仍存在于客户机计算机110上。然而,数据库114已被从客户机计算机110移动到了数据库服务器116。运行在客户机计算机110中的应用程序112经由数据库应用程序接口(API)向与数据库114通信地耦合的数据库服务器116发送对数据的请求。所请求的数据随后被返回给在客户机计算机110上执行的应用程序112。

[0028] 图1C是3层体系结构130的示例。在3层体系结构130中,应用程序112可被划分成在各自的客户机计算机110和服务器122上执行的分布式应用程序112、124。应用程序112可以实现具有UI逻辑的应用层。应用程序124可以实现具有业务和数据库访问逻辑的应用层。在客户机计算机110中运行的应用程序112向正在执行应用程序124的服务器122发送数据。应用程序124可随后执行业务逻辑,并向与数据库114通信地耦合的数据库服务器116发送对数据的请求。所请求的数据和所执行业务逻辑的结果随后被返回给应用程序112,并被渲染在客户机计算机110中。应注意,数据库服务器116可以与服务器122位于一处,或者是服务器122的一部分。换言之,硬件体系结构可以使得单个服务器122用作应用和数据库服务器两者。2层体系结构和3层(或n层)体系结构之间的区别因素是应用层中的一些或许多被移出客户机计算机110,并被分布在一个或多个其他服务器116、122之间。

[0029] 在开发和修改应用程序时,n层体系结构(诸如,3层体系结构130)相对于2层体系结构120来说,可以提供许多优点。例如,单个层可以被修改或添加,而不会导致对整个应用程序的完全重写。然而,在针对其中存在大量客户机的基于web的环境来实现n层体系结构时存在困难。每一客户机可能利用不同的web技术,包括不同的web浏览器、web服务以及web应用。此外,web技术被设计成与许多不同类型的底层硬件和软件体系结构一起工作,包括具有不同输入/输出(I/O)组件、形状因子、功率要求、处理能力、通信能力、和存储器资源等的各种设备。由此,可能难以在不需要广泛定制呈现层以适合于每一客户机的独特配置的情况下跨这许多异构设备和体系结构来统一地实现给定应用层(诸如,呈现层)。此外,应用程序的web版本可能与应用程序的非web版本不兼容,由此造成了对各自的分开的软件体系结构的需求。

[0030] 在各个实施例中,增强式n层体系结构提供了一框架,该框架使得能够将2层客户机-服务器应用体系结构迁移到3层应用体系结构,该3层应用体系结构将瘦客户机用于应用程序的呈现层。在一个实施例中,例如,每一客户机设备可以实现web客户机形式的瘦客户机。web客户机通常指用web技术实现的瘦客户机应用,诸如例如在客户机计算机中操作的web浏览器。它也可以指将浏览器增强为支持来自站点或服务器的定制服务的插件和助手应用。在此对web客户机的任何引用也可以指web浏览器的功能。

[0031] 图2示出了客户机-服务器系统200。在一个实施例中,客户机-服务器系统200可以包括增强式n层客户机-服务器系统。增强式n层客户机-服务器系统可以将应用程序划分成

多个层,包括呈现层。呈现层可以使用以下技术来实现:该技术被设计成便于使用解释运行时引擎来分隔和优化应用程序中的GUI渲染和用户事件。它允许使解释运行时引擎应用从基于2层客户机-服务器的体系结构适应到被托管的3层环境,同时减少该解释运行时引擎应用所需的改变。

[0032] 如之前参考图1A所描述的,许多应用遵循2层应用体系结构,在2层应用体系结构中,应用被组织成两个相互关联的组件——数据库服务器和客户机应用。数据库服务器可以托管系统和公司数据以及扩展的业务逻辑,该扩展的业务逻辑允许服务器处理较繁重的操作中的一些,这些操作在客户机处执行起来将非常耗时。同时,客户机应用可以执行以下功能:分发UI、提供数据项确认、及渲染报告,以及其他功能。

[0033] 在图2中示出的说明性实施例中,客户机-服务器系统200可包括服务器202和多个客户机204、206。当在不同的硬件平台上被实现时,服务器202和客户机204、206可以通过网络250彼此通信。当在同一硬件平台上被实现时,服务器202和客户机204、206可以经由合适的总线技术和体系结构彼此通信。虽然出于简洁的目的,图2中仅示出了单个服务器202和两个客户机204、206,但应理解,客户机-服务器系统200可以如给定实现所需的,实现任何数量的服务器和客户机。各实施例不限于该上下文。

[0034] 在一个实施例中,服务器202可以包括实现服务器应用210的电子设备。服务器应用210可以包括任何类型的服务器应用,诸如商用业务线应用。商用业务线应用的示例可包括但不限于会计应用、企业资源规划(ERP)应用、顾客关系管理(CRM)应用、和供应链管理(SCM)应用等。这些商用业务线应用有时被称为“中间层”应用,因为它们一般由商用企业网络中的服务器或服务器阵列来执行,而不是由诸如台式计算机的客户机计算机来执行。特定示例可包括由华盛顿州雷蒙德市的微软公司制造的Microsoft®Dynamics GP。Microsoft Dynamics GP是商用会计软件应用。商用业务线应用的另一特定示例可包括由华盛顿州雷蒙德市的微软公司制造的Microsoft Dynamics®AX。Microsoft Dynamics AX是商用ERP软件应用。然而,各实施例不限于这些示例。

[0035] 当服务器202执行服务器应用210的代码时,服务器202形成解释运行时引擎212。解释运行时引擎212实现服务器应用210的多个应用层,在客户机-服务器系统200中被称为应用逻辑214、数据库逻辑216和服务器呈现逻辑218。服务器应用210可以由通过网络250从客户机204、206处接收的、信号或消息形式的控制指示来控制 and 操作。

[0036] 在一个实施例中,客户机204、206可各自包括实现相应的web客户机230、240的电子设备。web客户机230、240各自可包括例如在相应的客户机204、206上执行的web浏览器的实例。web浏览器还可包括被设计为增强web浏览器以支持来自服务器202的定制服务的插件、web应用以及助手应用。在此对web客户机230、240的任何引用也可以指web浏览器的功能。

[0037] 客户机204、206可以包括相应的客户机适配器232、242。客户机适配器232、242中的每一个可被配置为与给定客户机204、206一起使用。以此方式,服务器应用210和解释运行时引擎212在被使用不同的web技术的不同客户机访问时不需要被修改。

[0038] 客户机适配器232、242可包括相应的客户机呈现逻辑238、248。客户机呈现逻辑238、248可被设计成将用户界面元素或视图呈现在客户机204、206的输出设备(诸如例如数字显示器)上。客户机呈现逻辑238、248可被设计成根据为服务器应用210实现的n层体系结

构与在服务器202上执行的服务器应用210的应用逻辑214、数据库逻辑216和服务器呈现逻辑218协作。

[0039] 客户机适配器232、242和相应的客户机呈现逻辑238、248可以与服务器呈现逻辑218协作以允许经由不同的客户机204、206来访问服务器应用210。每一客户机204、206可以将服务器呈现逻辑218的不同版本实现成相应的客户机呈现逻辑238、248,以适合于客户机204、206的特定配置。这可以在不必重写服务器呈现逻辑218,并且更重要地不必重写业务逻辑214和数据库逻辑216的情况下被实现。此外,服务器呈现逻辑218和客户机呈现逻辑238、248可以按减少网络250的通信流量和开销的方式交互,由此提高了速度和性能,同时减少了与通信延迟相关联的等待时间。

[0040] 服务器应用210可分开地或同时与客户机适配器232、242(或每一个的各单独版本)通信。同时操作的场景可包括在用户需要帮助并且管理员需要查看用户的web客户机视图的第二版本时。

[0041] 在各实施例中,服务器呈现逻辑218和客户机呈现逻辑238、248可以按利用图形用户界面(GUI)无关的对象260的高效方式来进行交互。GUI无关的对象260允许诸如GUI屏幕(例如,Microsoft Windows®Forms)之类的GUI元素在桌面环境和web环境之间自由移动。GUI无关的对象260允许服务器应用210作为后台中等待可经由传统的操作系统表单或web客户机表单接收的用户事件的服务来运行,并仍能执行脚本事件,而与传送该对象的表单的类型无关。

[0042] 除了可以影响应用逻辑事件的用户事件属性之外,GUI无关的对象260还可包含可以影响客户机适配器232、242的GUI相关的渲染的用户事件和任何用户事件属性及其他类型的信息。GUI无关的对象260由解释运行时引擎212生成并被发送到客户机适配器232、242,该GUI无关的对象260随后经由相应的客户机呈现逻辑238、248被渲染在客户机用户界面中。

[0043] 图3示出了n层客户机-服务器系统300的特定实现。客户机-服务器系统300可包括服务器302和客户机304。服务器302可以表示例如参考图2描述的服务器202。客户机304可以表示例如参考图2描述的客户机204、206之一或两者。

[0044] 在客户机-服务器系统300中示出的说明性实施例中,服务器302可以实现服务器应用310。在一个实现中,例如,服务器应用310可以使用微软的Dexterity®编程语言及其他合适类型的编程语言来被代码化。在被实现为微软Dexterity应用时,服务器应用310一般可被划分成两个不同的元素。第一元素是解释运行时引擎312,该引擎解决应用环境的诸如与操作系统(OS)进行通信和经由文件管理器316来建立和管理与数据库320的连接之类的技术方面。第二元素是应用字典313,该应用字典托管应用逻辑315,诸如,应用规则、业务规则、表单、报告、资源、元数据以及使能对用户命令和输入进行响应的应用代码。应用代码的示例可包括sanScript代码、Microsoft Visual Studio®插件、Microsoft Visual Basic®Application(VBA)、Microsoft Dexterity Continuum,等等。该体系结构将应用逻辑315与UI的样式改变和平台进步(诸如,例如平台OS的升级)隔离。

[0045] sanScript代码用于控制应用如何操作。sanScript代码一般被写在小片段或脚本中,这些小片段或脚本被附加到应用字典313中的对象,诸如域、菜单、屏幕和表单。当用户与该应用中的该特定对象交互时,脚本被运行。例如,被应用于“按下”按钮的脚本将在用户

点击该按钮时运行。

[0046] 如图所示,客户机304可以包括web客户机330。web客户机330可以表示例如web客户机230、240之一或两者。web客户机330可以递送面向用户界面和用户交互的组件和服务集合,包括供与服务器应用310一起使用的用户输入和轻量用户界面控件。然而,为了实现到3层体系结构的平滑迁移,需要克服由web客户机体系结构的引入造成的多个技术挑战,以启用高效的web客户机界面。

[0047] 在此描述的各实施例的目标是减少对现有代码和GUI元数据的所需修改。为了解决前述挑战中的一些,各个实施例涉及用于将用户界面管理器318和操作系统渲染引擎322从解释运行时引擎312中解耦出来的技术。用户界面管理器318是控制给定GUI系统中的各个用户界面元素(诸如,GUI屏幕)的放置和外观的系统软件。操作系统渲染引擎322是用于显示内容的系统软件。解释运行时引擎312是服务器应用310的执行版本。

[0048] 对表单(或屏幕)的使用是任何Microsoft Dexterity应用的核心组件。表单是用户将用其与服务器应用310进行交互的机制。当服务器应用310被实现成例如Microsoft Dexterity应用时,Microsoft Dexterity屏幕通常包括与对该屏幕的控制相关联的sanScript代码。给定屏幕的预期功能和在脚本解释器314的指导下的控制(例如,保存事务、公告批处理),sanScript代码响应于用户事件而执行。

[0049] 在服务器应用310的非web版本中,UI由用户界面管理器318管理,该用户界面管理器318进而与操作系统渲染引擎322通信以使实际的Microsoft Dexterity屏幕与之前由开发人员布局的控制元素一起显示在显示屏上。

[0050] 然而,为了促进向客户机-服务器系统300的web客户机3层体系结构的转移,可以将用户界面管理器318和操作系统渲染引擎322从解释运行时引擎312的功能中解耦出来。这允许web客户机332将用户界面管理器336和渲染引擎338的客户机版本实现在客户机304上。这还允许在服务器302上执行的解释运行时引擎312产生供web客户机332使用的GUI无关的对象360。利用GUI无关的对象360,经典的客户机可以继续提供典型的GUI屏幕(例如,Microsoft Win32®屏幕),同时还允许客户机304的web客户机330提供同一屏幕的基于web的表示,而无须改变服务器应用310的任何底层应用逻辑315。

[0051] 将用户界面管理器318和操作系统渲染引擎322从解释运行时引擎312中解耦出来允许屏幕(表单)在非web(例如,桌面或Win32)环境和web环境之间自由移动。随着用户界面管理器318和操作系统渲染引擎322被解耦,服务器应用310可以作为后台中等候可经由传统的Win32表单或web客户机表单来接收的用户事件的服务来运行,并仍能执行脚本事件,而不管提交该对象的表单的类型。

[0052] 为了促进该解耦,首先将服务器应用310的GUI相关的处理层和GUI无关的处理层分开。代替这两个层之间的直接通信,使用GUI无关的对象360来展示渲染和事件元数据。除了可以影响应用逻辑事件的用户事件属性之外,GUI无关的对象360还可以包含可以影响客户机适配器332的GUI相关的渲染的任何用户事件属性。GUI无关的对象360随后被发送至(GUI相关的)客户机适配器332,该对象被渲染在客户机304的显示器上的客户机用户界面屏幕中。一些客户机适配器332的示例可包括但不必限于Microsoft Silverlight®、HTML、Win32GDI、.Net表单等。

[0053] 图4示出了n层客户机-服务器系统400的特定实现。客户机-服务器系统400可包括

服务器402和客户机404。服务器402可以表示例如参考图2、3描述的服务器202、302。客户机404可以表示例如参考图2、3描述的客户机204、206、304之一或全部。

[0054] 在服务器402上,可存在包括解释运行时引擎412的服务器应用410,该解释运行时引擎412可负责执行一个或多个应用层,或与运行一个或多个应用层的其他组件耦合在一起。解释运行时引擎412还可包括脚本解释器414、文件管理器416以及用户界面管理器418。脚本解释器414可以与文件管理器416和服务器用户界面管理器418通信。文件管理器416还可以与数据库420通信。

[0055] 在客户机404上存在执行客户机适配器432的web客户机430。客户机适配器432可以包括用户界面管理器436和渲染引擎438,该渲染引擎438用于根据图2中示出的客户机呈现逻辑238、248将内容显示在客户机用户界面(诸如客户机用户界面)中。

[0056] 图4可表示3层应用体系结构,其中某些应用层可以分布在服务器402和客户机404之间。例如,如图2所示,客户机呈现逻辑238和/或248可以驻留在客户机404上,而应用逻辑214和数据库逻辑216可以分布在服务器402上。图4中示出的体系结构已将用户界面管理器436和渲染引擎438的功能从服务器402上的解释运行时引擎412中解耦出来,并将该功能与客户机适配器432一起放置在客户机404上。

[0057] 在一个实施例中,解释运行时引擎412可包括脚本解释器414。脚本解释器414一般可被安排成响应于诸如但不限于保存事务或公告批处理的用户事件而执行脚本代码。脚本化代码的示例可包括预脚本(pre-script)、改变脚本、和后脚本(post-script)及其他类型的脚本。

[0058] 在一个实施例中,解释运行时引擎412可包括文件管理器416。文件管理器416一般可被安排成对数据库420中存储的文件执行文件管理操作。文件管理操作的示例可包括新建文件、打开文件、复制文件、移动文件、删除文件等。

[0059] 在一个实施例中,解释运行时引擎412可包括用户界面管理器418。用户界面管理器418一般可被安排成控制各个用户界面元素(诸如,屏幕元素)在实现给定GUI系统的用户界面内的放置和外观。

[0060] 在操作中,用户可以经由web客户机430与客户机用户界面交互。web客户机430可以包括具有用于渲染基于web的内容的用户界面代码的web浏览器。web客户机430可以使用诸如HTML、XHTML和XML等各种web技术来实现。web客户机430的示例可包括但不限于由华盛顿州雷蒙德市的微软公司开发的Internet Explorer®以及其他类型的web浏览器软件。

[0061] 根据一实施例,在操作中,用户可以经由web客户机430与客户机用户界面交互,并可输入用户事件,这些用户事件可被客户机适配器432接收并处理。用户事件的示例可包括但不限于,将指针移动到某一个域、悬停在某一个域上、选择某一个域、鼠标点击某一按钮、填充文本域及类似的操作。用户事件可以使用用户事件属性集合来定义。在一个实施例中,仅需将用户事件属性的改变从web客户机430发送到服务器应用410,而非将用户事件属性的完整集合从web客户机430发送到服务器应用410。该差分技术可以节省通信带宽并减少等待时间。

[0062] 用户事件属性可以是向显示在用户界面布局中的用户界面元素分配的任何性质,诸如域、屏幕或图形对象。用户事件属性描述了相应用户界面元素的呈现样式或呈现格式的性质。用户事件属性可包括用户界面元素标识符(ID)、属性(例如,边界、字体、字体大小、

字体颜色、背景、背景色、样式、右对齐、居中对齐、左对齐、单倍行距、和双倍行距等)和属性值(例如,假、真、0、1等)、以及其他类型的信息。例如,GUI屏幕可具有标识符“Window 001”以及被设为假的、可重设大小属性,这意味着用户无法在运行时为GUI屏幕重设大小。这些仅是一些示例,任何用户界面元素和用户界面属性可以按照给定实现的需要被实现。各实施例不限于该上下文。

[0063] web客户机430可以在消息450中向服务器应用410发送已改变用户事件属性451的集合。在服务器402上操作的用户界面管理器418将消息450中的已改变用户事件属性451转发给脚本解释器414进行处理。服务器应用410可确保应用输入和应用状态在执行服务器应用410的任何应用逻辑之前是正确的。脚本解释器414可随后在与具有对数据库420的访问权的文件管理器416通信,如果执行从客户机404接收到的消息450中的已改变用户事件属性451得到的任何应用规则需要的话。在执行了合适的应用逻辑后,解释运行时引擎412可以产生GUI无关的对象452。GUI无关的对象452可以包括经更新的用户事件属性454及其他信息。由服务器402实现的用户界面管理器418可以将GUI无关的对象452连同任何经更新的用户事件属性454一起发送回客户机404。随后,客户机适配器432可通过客户机界面管理器436和渲染引擎438使用GUI无关的对象452以及由服务器应用410生成并从服务器应用410接收的经更新用户事件属性454来更新之前渲染的图形。

[0064] 上述实施例的操作可参考一个或多个逻辑流程来进一步描述。可以理解,除非另外指明,否则代表性的逻辑流程不一定要按所呈现的次序或者按任何特定次序来执行。而且,关于逻辑流程描述的各种活动可按串行或并行的方式执行。视给定一组设计和性能约束的需要,逻辑流程可使用所述实施例的一个或多个硬件元件和/或软件元件或替代元件来实现。例如,逻辑流程可被实现为供逻辑设备(例如,通用或专用计算机)执行的逻辑(例如,计算机程序指令)。

[0065] 图5示出逻辑流程500的实施例。逻辑流程500可示出根据一个或多个实施例来执行的操作。例如,逻辑流程500可示出由web客户机430和/或服务器应用410来执行的操作。

[0066] 在逻辑流程500中,在框502,用户与在客户机侧用户界面中执行的web客户机交互。例如,web客户机430可以接收从输入设备接收到的一个或多个控制指示形式的用户输入,该用户输入影响渲染引擎438所呈现的用户界面中的一个或多个用户界面元素。用户输入可以与用户界面元素交互,从而造成用户事件。例如,用户可以选择GUI屏幕中呈现的表单上的域,并修改该域的值。

[0067] 在逻辑流程500中,在框504,在其中执行的客户机适配器可以按与在服务器上执行的服务器应用兼容的方式来解释表示用户事件的控制指示。例如,由web客户机430执行的客户机适配器432可以按与服务器应用410类似的方式来解释用户事件。用户事件可以包括与在web客户机430上运行的用户界面的一个或多个用户交互,诸如但不限于,点击按钮、填充文本域等。

[0068] 在逻辑流程500中,框504处,解释操作检查新输入的用户事件属性,以在菱形506处确定用户事件属性是否已改变成服务器应用应该被通知到的所需程度。例如,客户机适配器432可以检查任何用户输入及受影响的用户界面元素的属性的相应改变,以确定用户事件属性是否已改变了超过某一阈值量。例如,系统在某一个域上以将其带入焦点可能不足以触发用户事件属性中的任何改变,而选择某一个域将足以通知服务器应用410。

[0069] 在逻辑流程500中,在框508,在需要通知的那些情况下,客户机适配器可以将任何待处理的已改变用户事件属性发送给服务器应用。例如,客户机适配器432可以经由网络250将消息450中的已改变用户事件属性451发送给服务器应用410。在一些实施例中,客户机适配器432可以将消息450中的多个用户事件的已改变用户事件属性451的多个集合发送至在服务器402上执行的服务器应用410。该“批处理”发送在许多方面是有用的,包括帮助服务器应用410进行用户事件定时。例如,脚本解释器414可以定时执行各种脚本(例如,预脚本、改变脚本、后脚本等)以确保服务器应用410的准确更新序列。批处理发送还可以通过跨网络250发送较少的消息450来减少通信开销。其他优点也存在,且各实施例不限于该上下文。

[0070] 在逻辑流程500中,在业务逻辑事件在框512处被执行之前,在服务器上执行的运行时引擎可以在框510处确保服务器应用的正确的输入/状态。例如,在服务器402上执行的解释运行时引擎412可以确保服务器应用410在执行任何应用或业务逻辑之前的正确的应用输入和应用状态。

[0071] 在逻辑流程500中,在框514,将从业务逻辑的执行得出的经更新用户事件属性连同GUI无关的对象一起传送回客户机适配器。例如,可以将应用或业务逻辑的执行得出的经更新用户事件属性454连同GUI无关的对象452一起从服务器应用410发送至web客户机430,以供传送回客户机适配器432。

[0072] 在逻辑流程500中,客户机适配器可随后在框516处使用经更新用户事件属性以及GUI无关的对象来更新客户机用户界面中的之前渲染的图像。例如,客户机适配器432接收GUI无关的对象452,渲染引擎438可以使用经更新用户事件属性454和GUI无关的对象452来更新客户机用户界面中的之前渲染的图像。

[0073] 图6A示出了可如何使用来自服务器应用410的数据来为客户机适配器432创建GUI无关的对象452的实施例。如上所述,客户机适配器432可接收具有已更新用户事件属性454的GUI无关的对象452。经更新用户事件属性454可包括GUI无关的对象元数据602及其他信息。在一个实施例中,GUI无关的对象元数据602可包括固定的或静态的元数据。经更新用户事件属性454还可包括属性/值集合604。固定/静态的GUI无关的对象元数据602可以与GUI无关的属性/值集合604组合以产生GUI无关的对象606,该GUI无关的对象606可以被客户机适配器432渲染在web客户机430中。

[0074] 图6B示出了可如何使用图6A中展示的构造来创建特定的GUI无关的对象452的实施例。经更新用户事件属性454可包括对象元数据612和属性/值集合614及其他信息。

[0075] 经更新用户事件属性454可包括具有一个或多个用户界面元素的对象元数据612。在该示例中,对象元数据612包括域形式的三个用户界面元素,标记为域A、域B以及域C。域A、B以及C中的每一个一般被示为具有分别围绕包括短语‘域A’、‘域B’以及‘域C’的默认字体文本的边框的文本框。

[0076] 经更新用户事件属性454还可包括属性/值集合614。在一个实施例中,属性/值集合614可以以诸如具有一个或多个元组(或行)的表格的数据结构来实现,其中每一元组包括以下性质(或列):用户界面元素的标识符、该用户界面元素的属性以及该属性的值。标识符、属性和值的表格可以对应于对象元数据612的各个域。

[0077] 在被组合在一起时,结果可以是GUI无关的对象616。如在GUI无关的对象616中示

出的,域A未从通用元数据模板改变,因为其属性或值中的任何一个在属性/值集合614中都没有被改变。域B被示为没有其边框,因为在属性/值集合614中属性‘边框’被设为值‘假’。域C中的文本被示为斜体,因为在属性/值集合614中属性‘斜体’被设为值‘真’。对象616现在可被客户机适配器432的渲染引擎438渲染在客户机404上的web客户机430中。

[0078] 图7示出了逻辑流程700的实施例。逻辑流程700可示出根据一个或多个实施例来执行的操作。例如,逻辑流程700可示出出于恢复已被解体的客户机适配器432的目的,由web客户机430和/或服务器应用410执行的操作。

[0079] 本文描述的各实施例的另一益处是在客户机适配器432被解体的情况下,在给定客户机404中所渲染的图像可被恢复。如果客户机适配器432被解体,则包括各GUI相关的对象的所渲染的图像也被解体。然而,服务器应用410可按GUI无关的对象452的形式来继续保持状态。如图7所示,在框702,用户可与在客户机侧用户界面中执行的web客户机430交互以创建客户机适配器432的新实例。在框704,客户机适配器432的新实例随后可重新连接到服务器应用410。在重新连接后,服务器应用410仍然可维护所有GUI无关的对象452的最后已知状态。在框706,所有GUI无关的对象452的最后已知状态被传送回客户机404并被客户机404接收。在框708,所有GUI无关的对象452的最后已知状态随后可与客户机404的web客户机430同步。结果是使用服务器应用410所存储的信息,客户机适配器432的当前状态可被有效地恢复。

[0080] 图8示出适用于实现上述各实施例的示例性计算体系结构800的实施例。计算体系结构800包括各种常见计算元件,如一个或多个处理器、协处理器、存储器单元、芯片组、控制器、外围设备、接口、振荡器、定时设备、视频卡、音频卡、多媒体输入/输出(I/O)组件,等等。然而,各实施例不限于由计算体系结构800来实现。

[0081] 如图8所示,计算体系结构800包括处理单元804、系统存储器806以及系统总线808。处理单元804可以是可购得的各种处理器中的任一种。双微处理器和其他多处理器体系结构也可用作处理单元804。系统总线808向包括但不限于系统存储器806的各系统组件提供到处理单元804的接口。系统总线808可以是若干类型总线结构中的任一种,这些总线结构还可互连到存储器总线(带有或没有存储器控制器)、外围总线、以及使用各类市场上可购买到的总线体系结构中的任一种的局部总线。

[0082] 系统存储器806可以包括各种类型的存储器单元,诸如只读存储器(ROM)、随机存取存储器(RAM)、动态RAM(DRAM)、双倍数据率DRAM(DDRAM)、同步DRAM(SDRAM)、静态RAM(SRAM)、可编程ROM(PROM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、闪存、诸如铁电聚合物存储器等聚合物存储器、奥氏存储器、相变或铁电存储器、硅-氧化物-氮化物-氧化物-硅(SONOS)存储器、磁卡或光卡、或适于存储信息的任何其他类型的介质。在图8示出的所示实施例中,系统存储器806可包括非易失性存储器810和/或易失性存储器812。基本输入/输出系统(BIOS)可以存储在非易失性存储器810中。

[0083] 计算机802可包括各种类型的计算机可读存储介质,包括内置硬盘驱动器(HDD)814、用于读写可移动磁盘818的磁软盘驱动器(FDD)816、以及用于读写可移动光盘822(例如,CD-ROM或DVD)的光盘驱动器820。HDD 814、FDD 816、以及光盘驱动器820可分别由HDD接口824、FDD接口826和光盘驱动器接口828连接到系统总线808。用于外置驱动器实现的HDD接口824可包括通用串行总线(USB)和IEEE 1394接口技术中的至少一种或两者。

[0084] 驱动器及相关联的计算机可读介质提供了对数据、数据结构、计算机可执行指令等的易失性和/或非易失性存储。例如,多个程序模块可被存储在驱动器和存储器单元810、812中,包括操作系统830、一个或多个应用程序832、其他程序模块834和程序数据836。这一个或多个应用程序832、其他程序模块834、以及程序数据836可包括,例如,客户机服务器系统200、300和400的各种软件组件。

[0085] 用户可以通过一个或多个有线/无线输入设备,例如键盘838和诸如鼠标840等定点设备将命令和信息输入到计算机802中。其他输入设备可包括话筒、红外(IR)遥控器、操纵杆、游戏垫、指示笔、触摸屏等等。这些和其他输入设备通常通过耦合到系统总线808的输入设备接口842连接到处理单元804,但也可通过诸如并行端口、IEEE 1394串行端口、游戏端口、USB端口、IR接口等其他接口连接。

[0086] 一个或多个监视器844或其它类型的显示设备也经由诸如视频适配器846等的接口连接到系统总线808。除了监视器844之外,计算机通常包括诸如扬声器、打印机等其他外围输出设备。一个或多个监视器845也可经由输入设备接口842和/或诸如USB集线器843等的集线器连接到系统总线808。监视器845可包括各种组件,如摄像头、话筒阵列、触摸传感器、动作传感器、扬声器等等。这些组件可经由USB中枢843连接到输入设备接口842。

[0087] 计算机802可使用经由有线和/或无线通信至一个或多个远程计算机(诸如远程计算机848)的逻辑连接在联网环境中操作。远程计算机848可以是工作站、服务器计算机、路由器、个人计算机、便携式计算机、基于微处理器的娱乐设备、对等设备或其他常见的网络节点,并且通常包括相对于计算机802描述的许多或所有元件,但为简明起见仅示出了存储器/存储设备850。所描绘的逻辑连接包括到局域网(LAN) 852和/或例如广域网(WAN) 854等更大网络的有线/无线连接。这种LAN和WAN联网环境常见于办公室和公司,并且方便了诸如内联网等企业范围计算机网络,所有这些都可连接到例如因特网等全球通信网络。

[0088] 当在LAN联网环境中使用时,计算机802通过有线和/或无线通信网络接口或适配器856连接到LAN 852。适配器856可以方便到LAN 852的有线和/或无线通信,并且还可包括其上设置的用于使用适配器856的无线功能进行通信的无线接入点。

[0089] 当在WAN联网环境中使用时,计算机802可包括调制解调器858,或连接到WAN 854上的通信服务器,或具有用于诸如通过因特网等通过WAN 854建立通信的其他装置。或为内置或为外置以及有线和/或无线设备的调制解调器858经由输入设备接口842连接到系统总线808。在联网环境中,相对于计算机802所描绘的程序模块或其部分可以存储在远程存储器/存储设备850中。将明白,所示网络连接是示例性的,并且可以使用在计算机之间建立通信链路的其他手段。

[0090] 计算机802可操作来使用IEEE 802标准家族来与有线和无线设备或实体进行通信,这些实体例如是在操作上安置成与例如打印机、扫描仪、台式和/或便携式计算机、个人数字助理(PDA)、通信卫星、任何一件与无线可检测标签相关联的设备或位置(例如,电话亭、报亭、休息室)以及电话进行无线通信(例如,IEEE 802.11空中调制技术)的无线设备。这至少包括Wi-Fi(即无线保真)、WiMax和蓝牙™无线技术。由此,通信可以如对于常规网络那样是预定义结构,或者仅仅是至少两个设备之间的自组织(ad hoc)通信。Wi-Fi网络使用称为IEEE 802.11x(a、b、g等等)的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi网络可用于将计算机彼此连接、连接到因特网以及连接到有线网络(使用IEEE 802.3相关的

介质和功能)。

[0091] 图9示出适用于实现上述各实施例的示例性通信体系结构900的框图。通信体系结构900包括各种常见通信元件,如发射机、接收机、收发机、无线电装置、网络接口、基带处理器、天线、放大器、滤波器,等等。然而,各实施例不限于由通信体系结构900来实现。

[0092] 如图9所示,通信体系结构900包括一个或多个客户机902和服务器904。客户机902可实现web客户机330。服务器904可实现运行时引擎312。客户机902和服务器904可操作地连接到可被用来存储相应客户机902和服务器904本地的信息(如cookie和/或相关联的上下文信息)的一个或多个相应客户机数据存储908和服务器数据存储910。

[0093] 客户机902和服务器904可以使用通信框架906在彼此之间传递信息。通信框架906可以实现任何公知通信技术,如适用于与分组交换网络(例如,诸如因特网等公共网络、诸如企业内联网等专有网络,等等)、电路交换网络(例如,公共交换电话网)、或分组交换网络和电路交换网络的组合(使用合适的网关和转换器)一起使用的技术。客户机902和服务器904可以包括被设计成可与通信框架906进行互操作的各种类型的标准通信元件,如一个或多个通信接口、网络接口、网络接口卡(NIC)、无线电装置、无线发射机/接收机(收发机)、有线和/或无线通信介质、物理连接器等。作为示例而非限制,通信介质包括有线通信介质和无线通信介质。有线通信介质的示例可以包括导线、电缆、金属线、印刷电路板(PCB)、背板、交换光纤、半导体材料、双绞线、同轴电缆、光纤、所传播的信号等。无线通信介质的示例可以包括声学、射频(RF)频谱、红外和其他无线介质。客户机902和服务器904之间的一种可能的通信可以是以适用于在两个或更多计算机进程之间传输的数据包的形式。例如,数据包可以包括cookie和/或相关联的上下文信息。

[0094] 各实施例可以使用硬件元件、软件元件或两者的组合来实现。硬件元素的示例可以包括设备、逻辑设备、组件、处理器、微处理器、电路、电路元素(例如,晶体管、电阻器、电容器、电感器等)、集成电路、专用集成电路(ASIC)、可编程逻辑器件(PLD)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、存储器单元、逻辑门、寄存器、半导体设备、芯片、微芯片、芯片组等。软件元件的示例可以包括软件组件、程序、应用软件、计算机程序、应用程序、系统程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子例程、函数、方法、过程、软件接口、应用程序接口(API)、指令集、计算代码、计算机代码、代码段、计算机代码段、文字、值、符号、或其任意组合。确定一实施例是否使用硬件元件和/或软件元件来实现可根据如给定实现所需的任何数量的因素而变化,这些因素诸如所需计算速率、功率级、耐热性、处理周期预算、输入数据速率、输出数据速率、存储器资源、数据总线速度以及其他设计或性能约束。

[0095] 一些实施例可包括制品。制品可包括被设置为存储逻辑的计算机可读的存储介质。计算机可读的存储介质的示例可包括能够存储电子数据的任何存储介质,包括易失性存储器或非易失性存储器、可移动或不可移动存储器、可擦除或不可擦除存储器、可写或可重写存储器等。逻辑的示例可包括各种软件元件,诸如软件组件、程序、应用、计算机程序、应用程序、系统程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子例程、函数、方法、过程、软件接口、应用程序接口(API)、指令集、计算代码、计算机代码、代码段、计算机代码段、文字、值、符号、或其任意组合。例如,在一个实施例中,制品可以存储可执行计算机程序指令,该指令在由计算机执行时使得该计算机执行根据所描述的各实施例的方法

和/或操作。可执行计算机程序指令可包括任何合适类型的代码,诸如源代码、已编译代码、已解释代码、可执行代码、静态代码、动态代码等。可执行计算机程序指令可根据用于指示计算机执行特定功能的预定义的计算机语言、方式或句法来实现。这些指令可使用任何合适的高级、低级、面向对象、可视、已编译和/或已解释编程语言来实现。

[0096] 一些实施例可使用表述“一个实施例”和“一实施例”及其派生词来描述。这些术语意味着结合该实施例描述的特定特征、结构、或特性包括在至少一个实施例中。出现在说明书中各个地方的短语“在一个实施例中”并不必全都指的是同一实施例。

[0097] 一些实施例可使用表述“耦合的”和“连接的”及其派生词来描述。这些术语不必旨在互为同义词。例如,一些实施例可使用术语“连接的”和/或“耦合的”来描述以指示两个或更多元件彼此有直接的物理或电接触。然而,术语“耦合的”还可以意味着两个或更多元件彼此不直接接触,而仍彼此合作或交互。

[0098] 要强调,本公开的摘要是为了允许读者快速确定本技术公开的性质而提供的。提交摘要的同时要明白,将不用它来解释或限制权利要求的范围或含义。另外,在前面的详细描述中,可以看到,出于将本公开连成一个整体的目的而将各种特征组合在一起放在单个实施例中。此公开方法将不被解释为反映所要求保护的实施例要求比每个权利要求中明确陈述的更多特征的意图。相反,如所附权利要求书所反映,发明性的主题存在于比单个已公开实施例的所有特征少的特征中。从而,据此将所附权利要求结合进详细描述中,其中每个权利要求独立地代表一个单独的实施例。在所附权利要求书中,术语“包括”和“其中”分别用作术语“包含”和“其特征存在于”的易懂的英文等价词。而且,术语“第一”、“第二”、“第三”等等只用作标记,而不旨在将数字要求强加于其对象上。

[0099] 尽管用结构特征和/或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于上述具体特征或动作。更确切而言,上述具体特征和动作是作为实现权利要求的示例形式公开的。

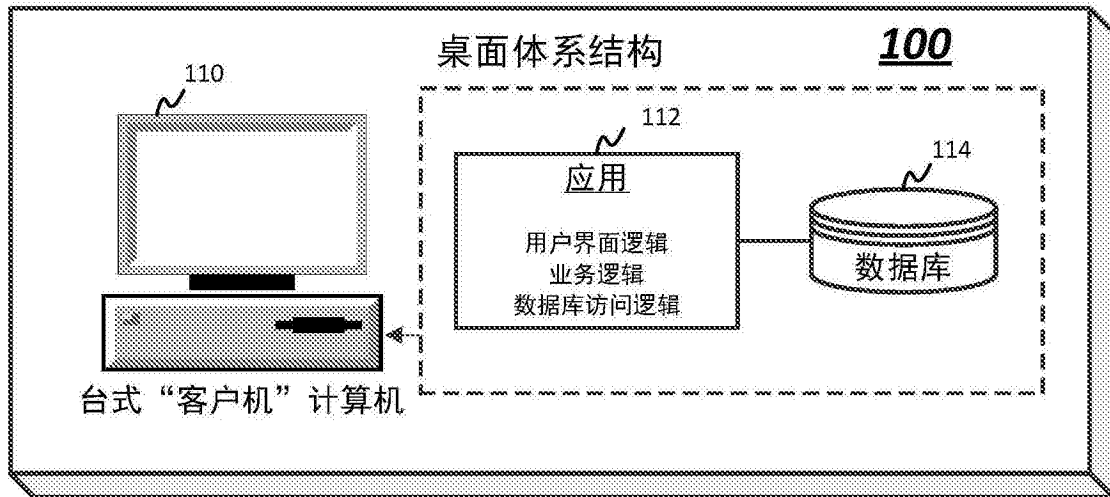


图1A

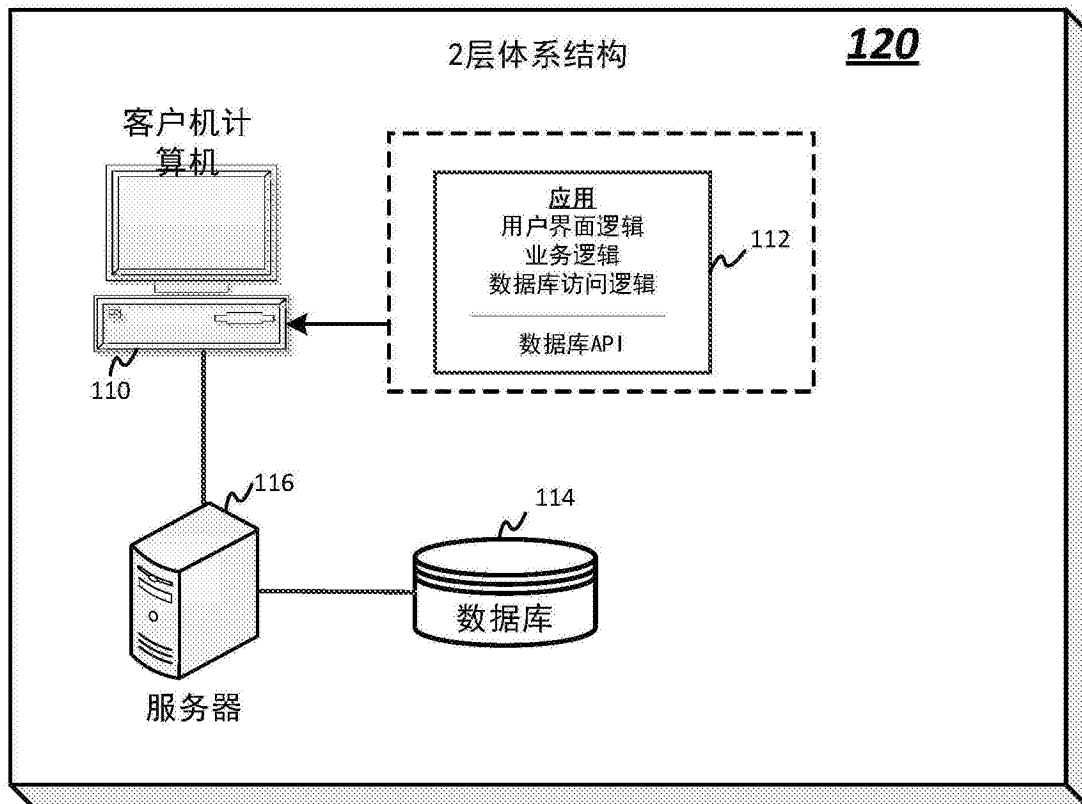


图1B

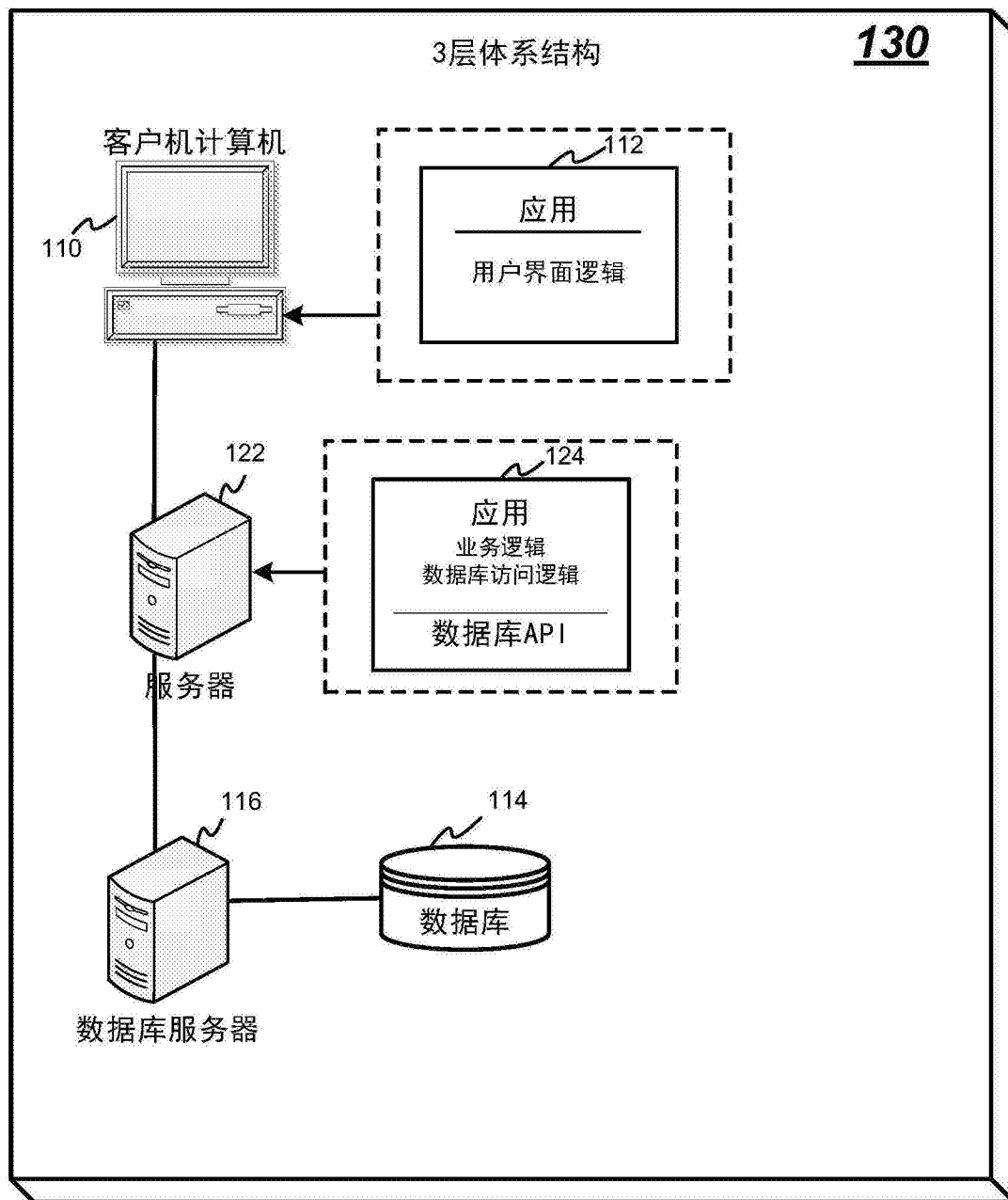


图1C

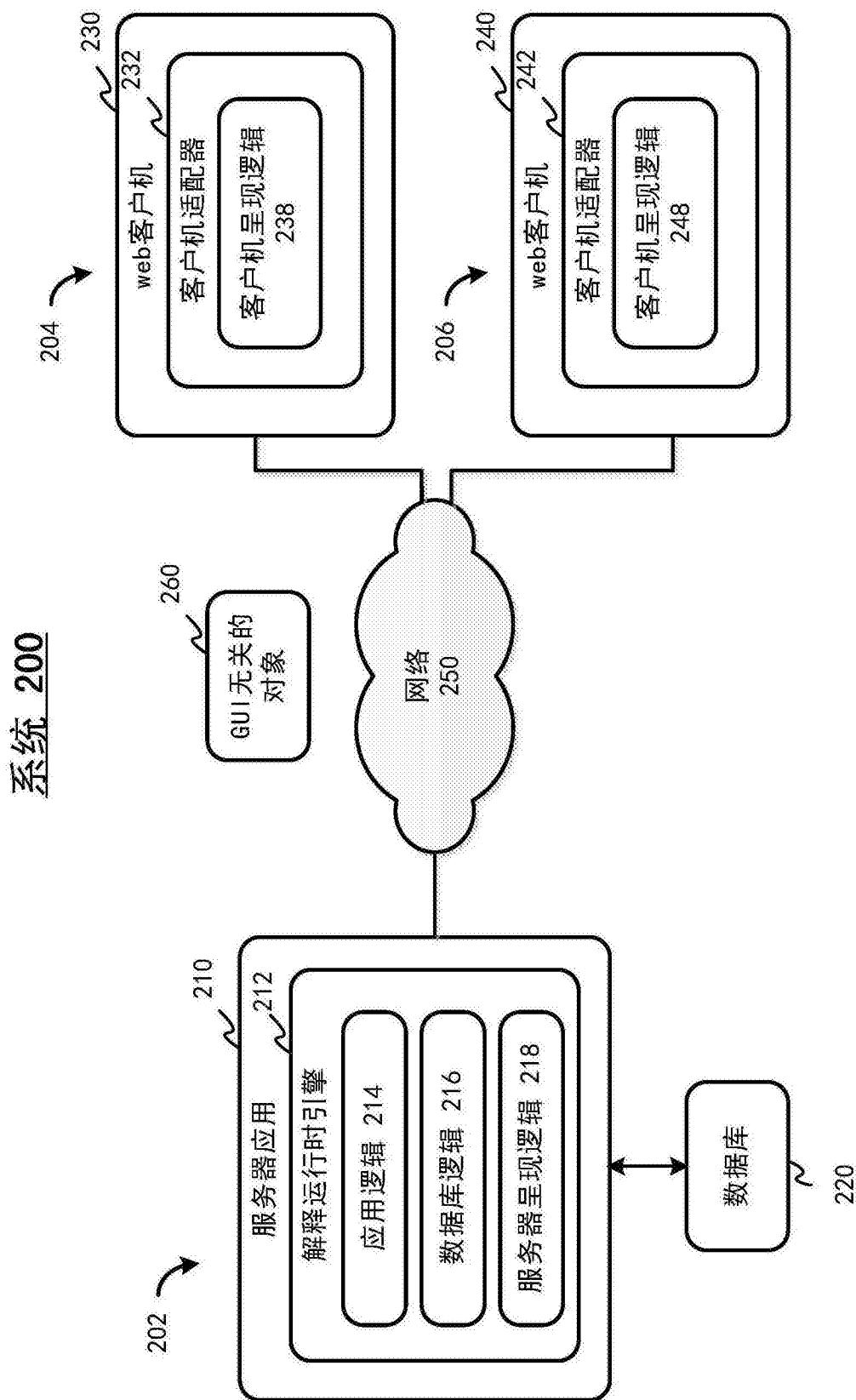


图2

系统 300

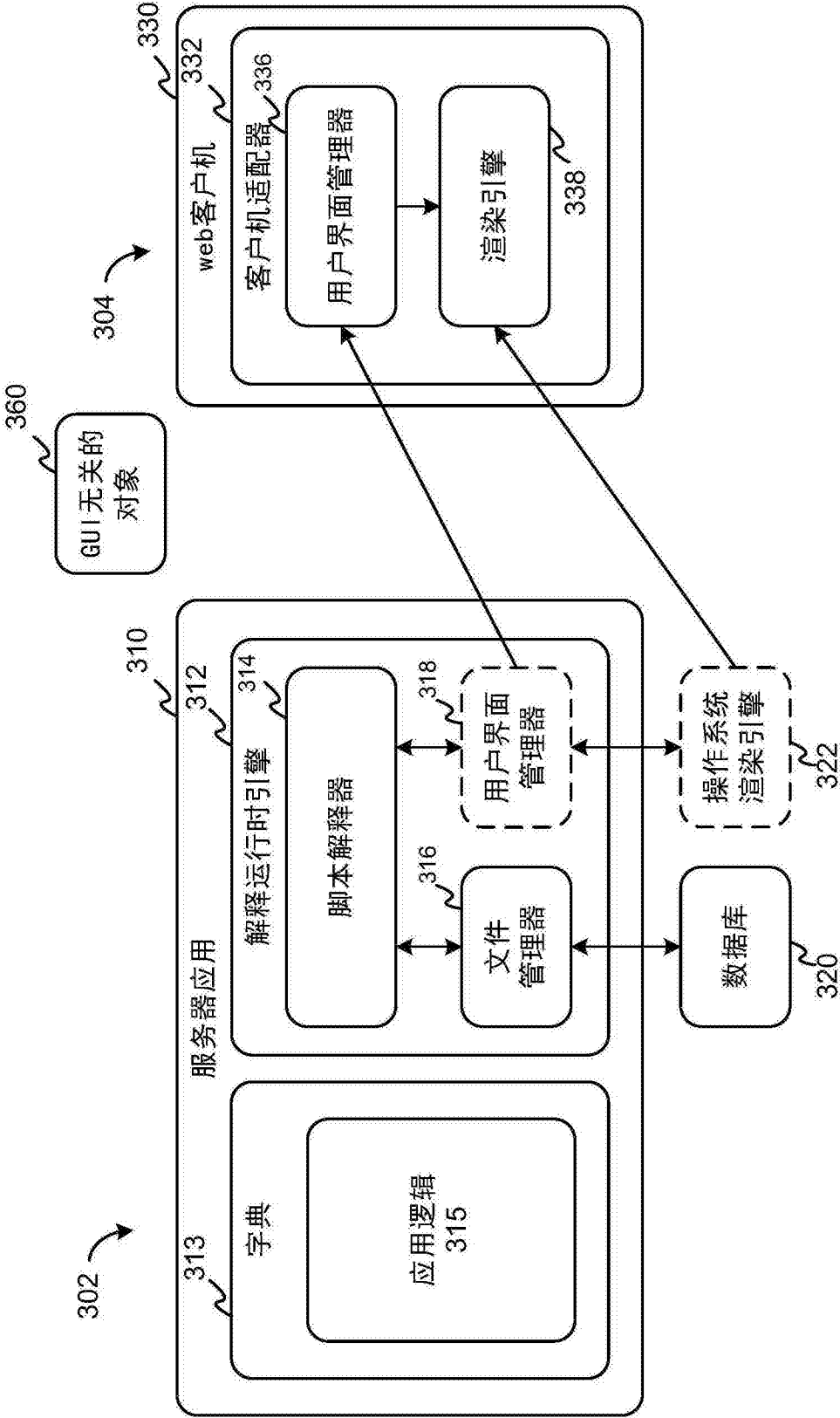


图3

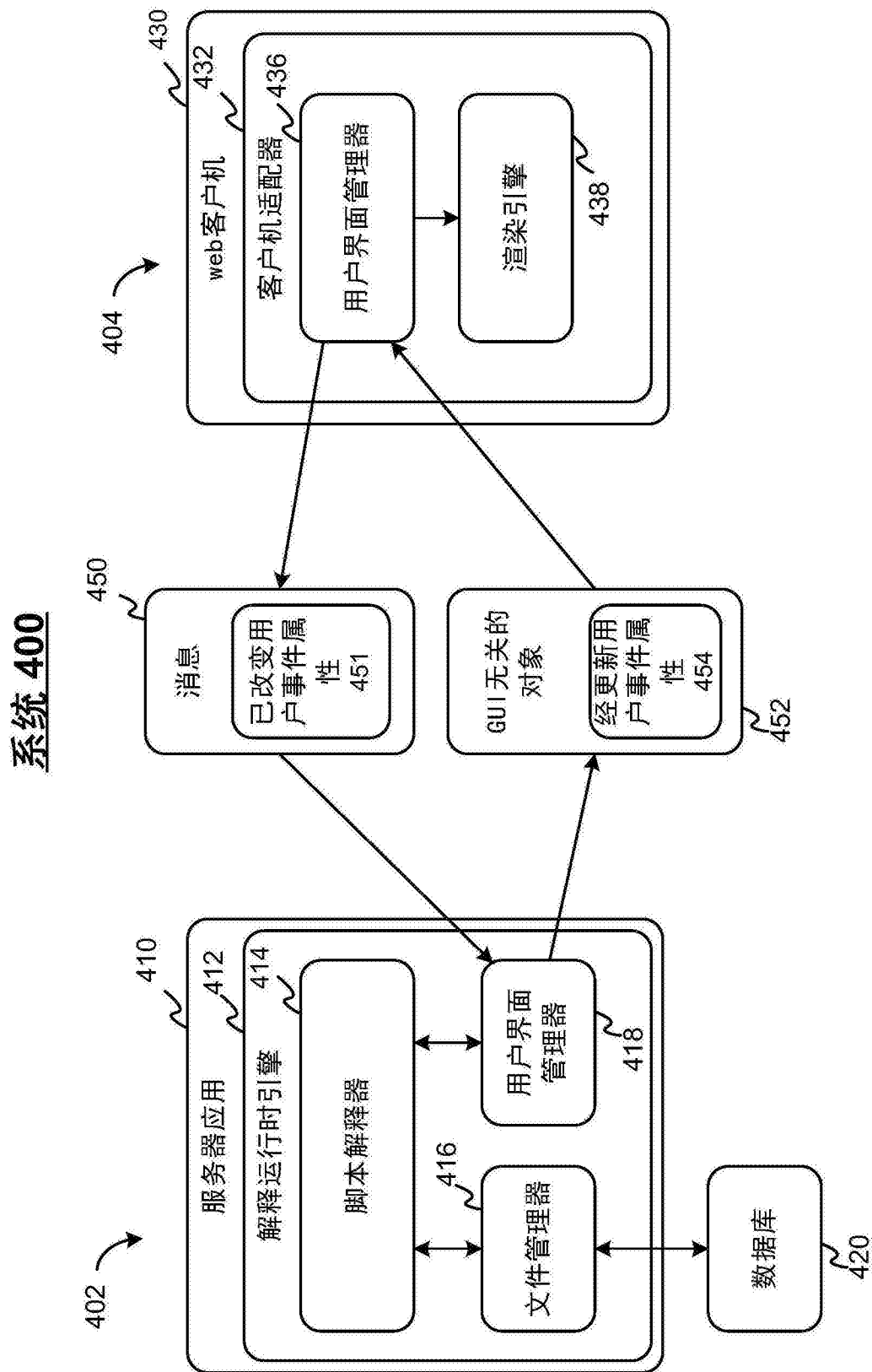


图4

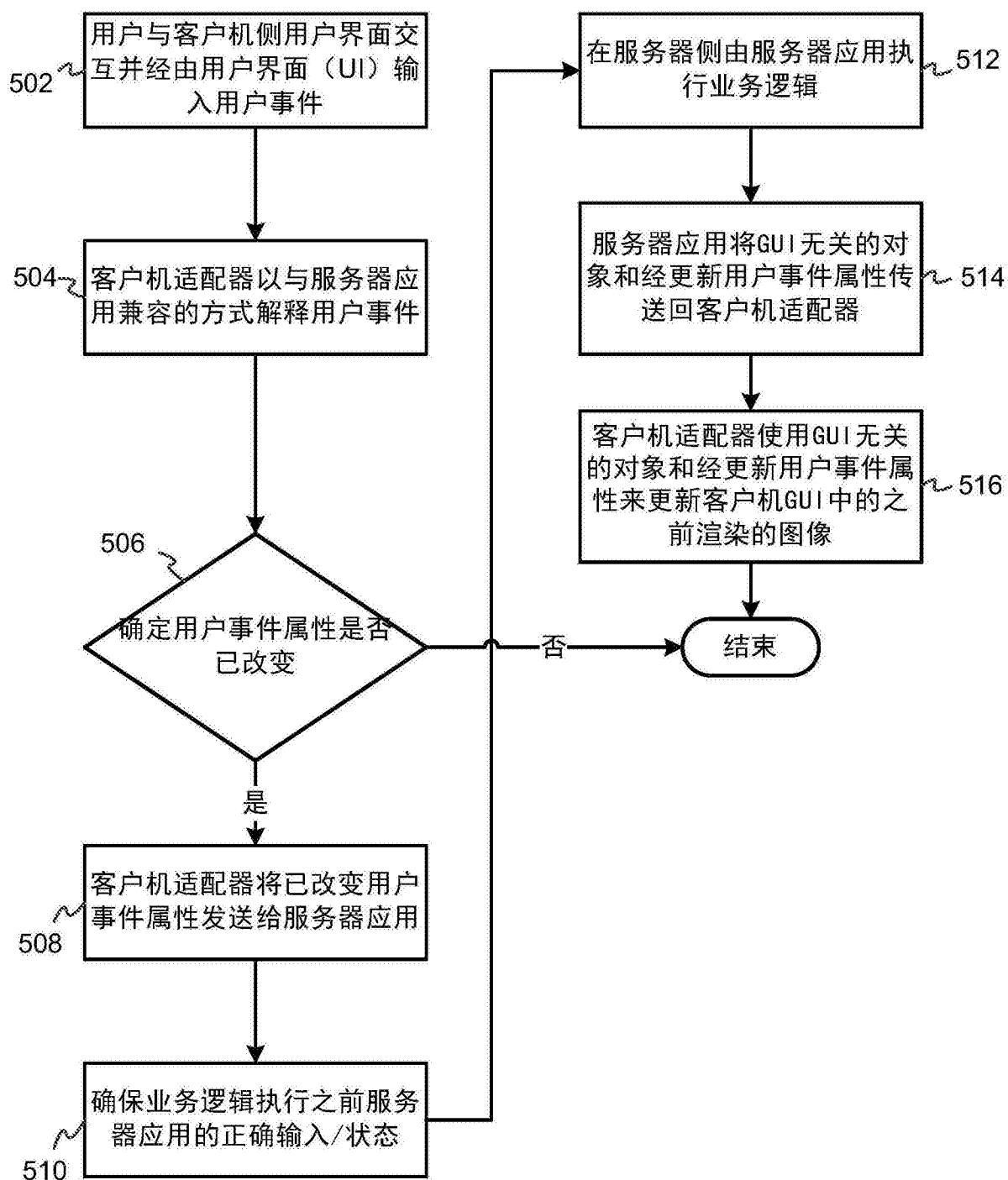
500

图5



图6A

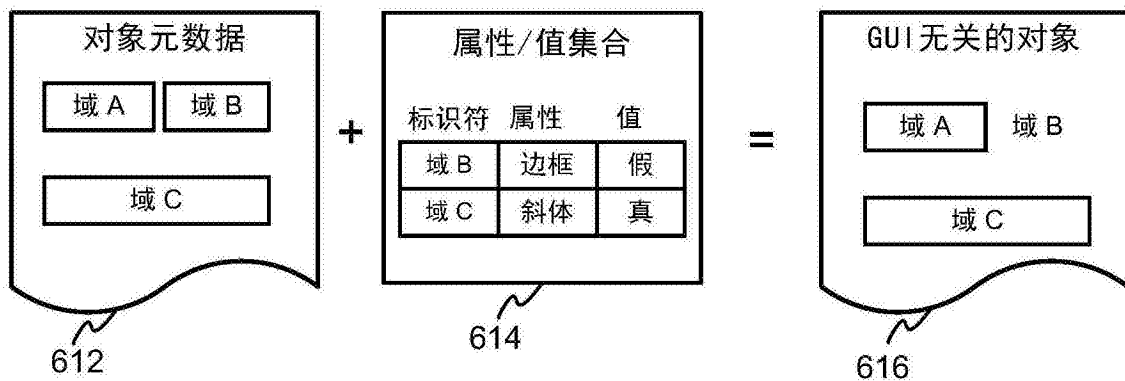


图6B

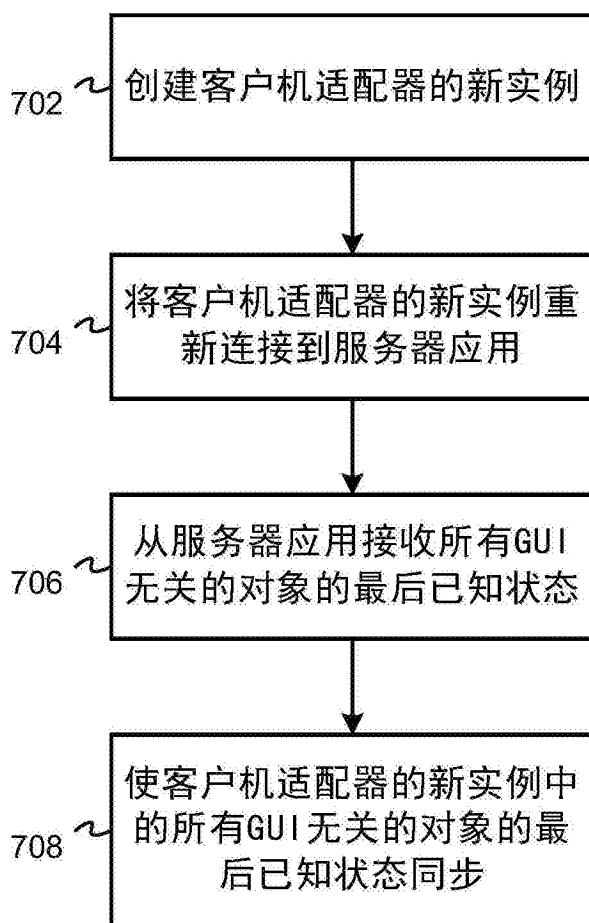
700

图7

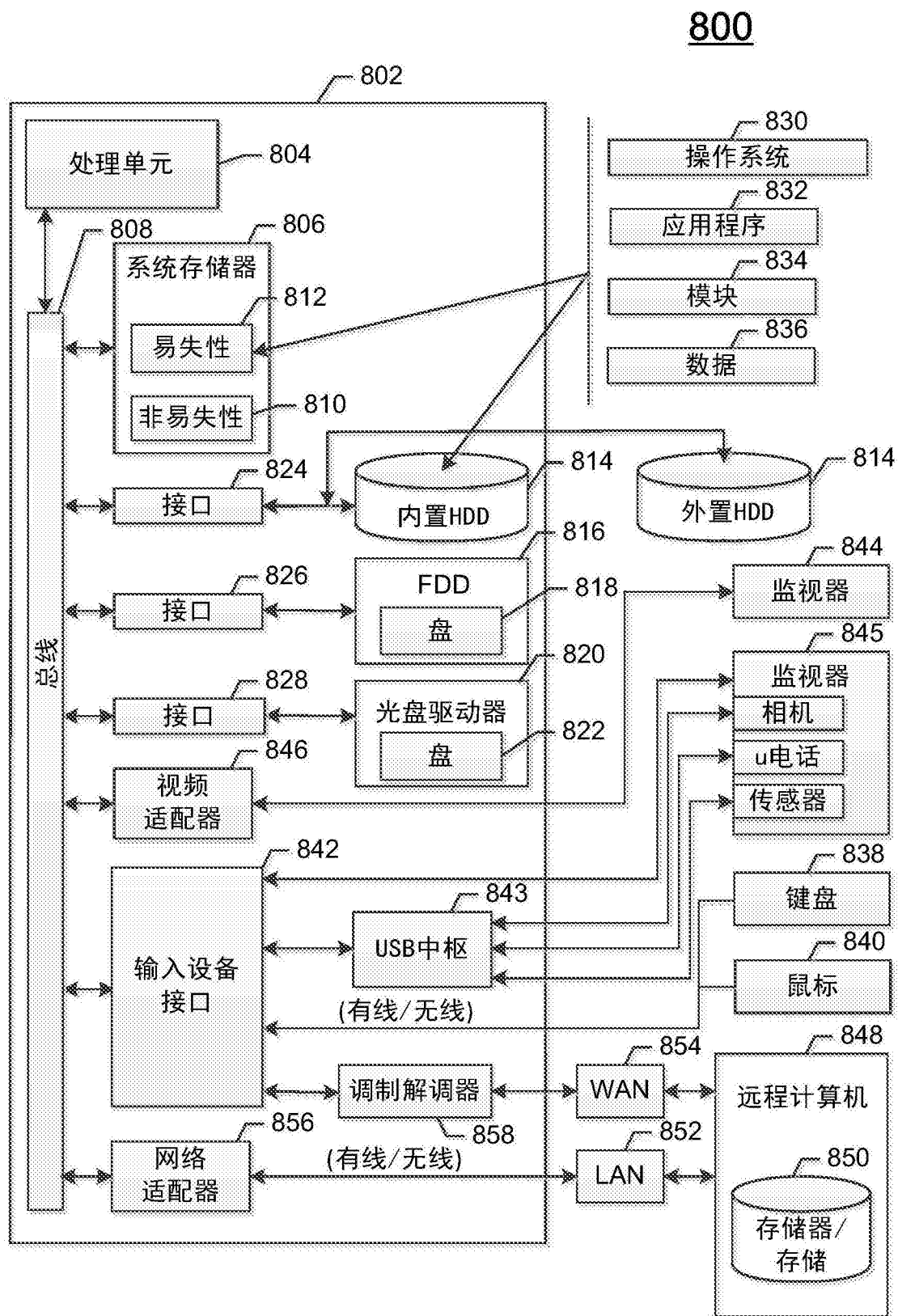


图8

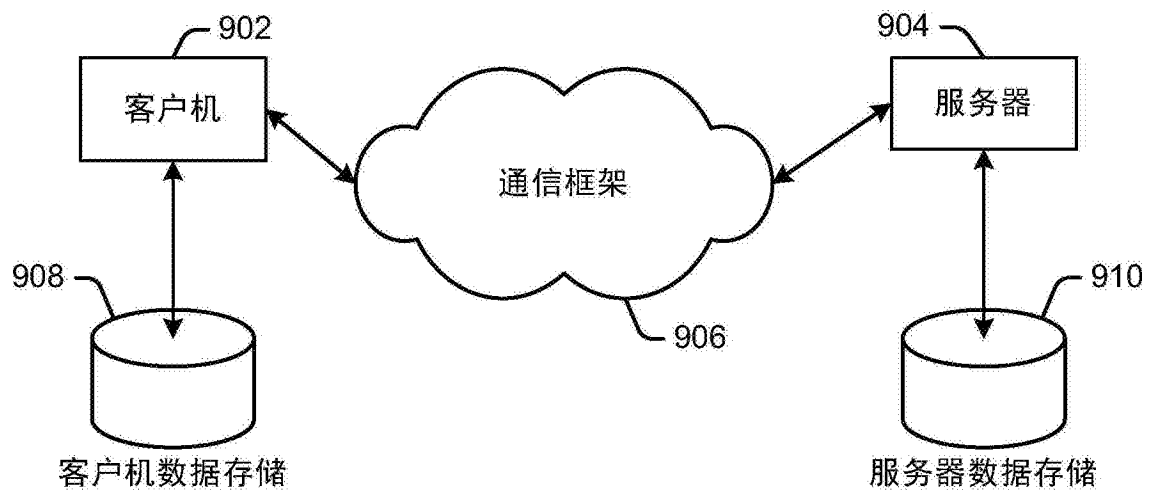
900

图9