



(12) 发明专利

(10) 授权公告号 CN 1703048 B

(45) 授权公告日 2011. 10. 12

(21) 申请号 200510075867. 4

(22) 申请日 2005. 05. 27

(30) 优先权数据

10/856, 399 2004. 05. 27 US

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 H · F · 尼尔森

G · 克里杉萨科宝罗斯

(74) 专利代理机构 上海专利商标事务所有限公
司 31100

代理人 蔡悦

(51) Int. Cl.

H04L 29/06 (2006. 01)

(56) 对比文件

US 2002029256 A1, 2002. 03. 07, 说
明书第 [0067]–[0083], [0130]–[0140],

[0171]–[0175], [0202]–[0217], [0237]–[0264]
小节, 图 3, 7 – 10, 14 – 17.

EP 1249765 A1, 2002. 10. 16, 说明书第
[0003]–[0007] 小节, 图 2.

审查员 李菁

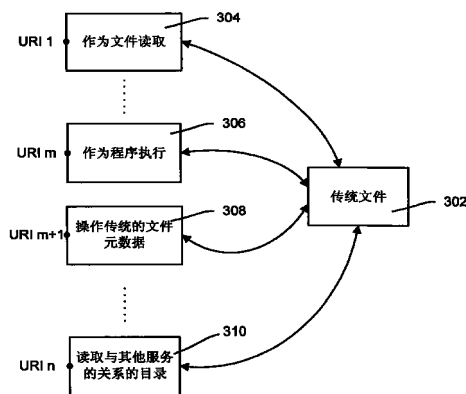
权利要求书 2 页 说明书 24 页 附图 13 页

(54) 发明名称

网络服务应用协议和 SOAP 处理模型

(57) 摘要

描述了一种网络服务应用协议 (WSAP), 它包括基于 SOAP 的协议, 该协议基于公用消息操作集, 定义网络服务应用模型。WSAP 根据它们如何彼此相互作用, 提供定义网络服务的基础。在 WSAP 中, 分别为单个端口提供服务的行为 (例如 UR), 由此可以直接交互作用服务的行为。WSAP 还定义用于操作的共享语义, 识别消息是什么、消息的处理是什么, 以及它如何影响接收者的状态 / 行为。消息的排序还提供正在发生什么的上下文。作为操作和排序的已知特性的结果, 中介体能处理所交换的消息以便将值添加到通信上。SOAP 处理模型定义成并行或顺序运行的合成多个网络服务。



1. 一种在计算环境中执行的方法,所述计算环境包括用于表示服务的状态和行为的网络服务应用协议 WSAP,所述方法包括:

确定网络服务具有不同行为;

向每个不同行为分配标识符,所述标识符允许客户机访问那个行为,其中所述标识符是 URI;

接收请求所述具有不同行为的网络服务的客户机请求;以及

响应所述客户机请求,返回为每个不同行为所分配的标识符,以允许所述客户机使用相应的标识符的来访问每个行为。

2. 如权利要求 1 所述的方法,其特征在于,还包括向所述客户机返回定义所述服务能处理的操作的订约。

3. 如权利要求 2 所述的方法,其特征在于,向所述客户机返回订约包括向客户机提供文档的标识符。

4. 如权利要求 2 所述的方法,其特征在于,还包括向所述客户机提供捆绑列表。

5. 如权利要求 2 所述的方法,其特征在于,所述订约定义来自包含 LOOKUP、CREATE、QUERY、INSERT、DELETE、GET、NOTIFY、REPLICATE、CANCELREPLICATE、SUBSCRIBE、UNSUBSCRIBE 和 DROP 操作的集合的至少一个操作。

6. 如权利要求 5 所述的方法,其特征在于,服务以预定顺序,实现所述集合的至少两个操作。

7. 如权利要求 1 所述的方法,其特征在于,还包括基于所分配的标识符,从客户机接收发送到实体的消息。

8. 如权利要求 7 所述的方法,其特征在于,所述实体包括中介体,并且所述方法还包括在所述中介体处理所述消息。

9. 如权利要求 1 所述的方法,其特征在于,所述标识符中的一个对应于预约列表以便任何预约服务具有单独的状态。

10. 如权利要求 9 所述的方法,其特征在于,还包括通知所述预约列表上的用户。

11. 如权利要求 9 所述的方法,其特征在于,还包括通过将数据复制到所述预约列表上的用户服务,与另一服务同步。

12. 如权利要求 9 所述的方法,其特征在于,还包括响应于来自包含 REPLICATE、CANCELPREPLICATE、SUBSCRIBE 和 UNSUBSCRIBE 操作的集合的至少一个操作,修改预约列表的状态。

13. 如权利要求 9 所述的方法,其特征在于,还包括通过与助手服务通信,维护所述预约列表。

14. 如权利要求 13 所述的方法,其特征在于进一步包括向所述客户机提供捆绑列表,其中,在捆绑列表中标识所述助手服务。

15. 一种在计算环境中执行的方法,所述计算环境包括用于表示服务的状态和行为的网络服务应用协议 WSAP,所述方法包括:

在所述服务处,接收来自客户机的订约请求,所述订约定义了服务能够处理的操作,所述请求请求所述具有不同行为的网络服务,所述不同行为的每一个被分配一个标识符,所述标识符允许客户机访问那个行为,其中所述标识符是 URI;

响应于所述请求,提供订约文档的标识符,和捆绑列表文档的标识符,所述订约文档包含包括消息排序和对应于所述服务的行为的交互作用点集的订约信息,并且所述捆绑列表文档包含所述服务的实例间的关系的服务关系信息。

16. 如权利要求 15 所述的方法,其特征在于,所述捆绑列表文档包含维护用于所述服务的至少一个预约列表的至少一个助手服务的标识符。

17. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 CREATE 请求。

18. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 QUERY 请求。

19. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 GET 请求。

20. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 INSERT 请求。

21. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 UPDATE 请求。

22. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 DELETE 请求。

23. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收 DROP 请求。

24. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收通知请求。

25. 如权利要求 15 所述的方法,其特征在于,还包括在交互作用点接收针对修改预约列表的请求。

26. 如权利要求 25 所述的方法,其特征在于,还包括响应来自包含 REPLICATE、CANCELPREPLICATE、SUBSCRIBE 和 UNSUBSCRIBE 操作的集合的至少一个操作,修改所述预约列表的状态。

27. 如权利要求 26 所述的方法,其特征在于,修改所述预约列表的状态包括与助手服务通信。

网络服务应用协议和 SOAP 处理模型

技术领域

[0001] 本发明通常涉及计算机系统和网络,以及更具体地说,涉及网络服务。

背景技术

[0002] 有许多类型的计算机用户和应用需要管理或者访问的计算服务、资源和数据。这些资源和数据包括本地维护的服务和数据、公司网络和其他远程可访问站点,包括内联网和因特网上维护的数据。

[0003] 网络服务的概念通常针对经实质上为交叉平台的协议和标准,向客户机提供计算服务。通常,网络服务提供用于将分布式应用的节点连接在一起的基本工具,而与请求客户机机正运行的平台的类型无关。由于存在许多不同的计算平台,便于交换网络信息的各种平台无关的机制和协议在网络服务中正变得普遍,包括 HTTP(超文本传输协议)、XML(可扩展标记语言)、XML 模式以及 SOAP(简单对象访问协议)XML。传统的网络服务,其中企业、公司和其他提供者向用户和应用提供服务,目前基于这些标准。此外,如转让给本发明的受让人的 U. S. 专利申请序列号 10/328,714,所述,网络服务能表示其他类型资源,诸如硬件设备、数据库等等。

[0004] 当前,网络服务用来与客户机机交换数据的协议以根据由网络服务提供的订约,达成一致的方式,允许客户机机向对应于该服务的服务端口(例如 URL)发送消息以及接收在客户机机处返回的消息。然而,一个潜在的问题是客户机机需要通过服务的 URL 发送其消息,即使服务本身可能由单独的实体建立,由此通过服务的 URL 的通信变成潜在的瓶颈。

[0005] 现有协议的另一问题在于除基本上注意到消息存在外,没有消息格式的内在知识,任何指定消息是不可观察的。因此,对中介体(例如另一可信服务)来说,没有方法来提供消息交换值,因为中介体不能确信消息的基本特性和/或目的。

[0006] 所需要的是客户机机和网络服务能安全和有效地与网络服务,包括组成服务的实体通信,允许中介体增加通信价值的机制。

发明内容

[0007] 简单地说,本发明提供一种方法、系统和机制,包括网络服务应用协议(WSAP),其中,形式上表示经端口(例如 URL)的服务的状态和行为,由此可以直接接触服务中的实体,以便交互作用。换句话说,与其他协议不同,WSAP 以形式化的方式,例如通过具有其自己的端口的服务的每个行为,描述网络服务的行为。WSAP 协议还通过定义有关识别消息是什么、处理的消息是什么,以及它如何影响接收机的状态/行为的动词的共享语义,便于使用中介。另外,消息和操作的排序提供正发生的事的环境。

[0008] 通常,在任何时候服务的逻辑实体的行为(或状态)不同时,WSAP 允许服务提供其自己端口的每个行为/状态(例如诸如 URL 的身份)。用这种方式,客户机机能直接与控制行为/状态的服务内的逻辑实体通信。举例来说,考虑提供访问存储设备,诸如硬盘驱动器的网络服务。传统上,在经文件系统打开文件和接收文件句柄后,为读取和将数据写入打

开的文件,客户机机将文件句柄发送到文件系统。通过 WSAP,而不是返回文件句柄,仲裁器服务将 URI 返回到提供对有关适当状态(数据)的行为(执行操作)的访问的网络服务的另一逻辑实体(另一服务)。在该例子中,通过 WSAP,客户机机将接收用于写入(例如使用适当的 XML 模式)到存储介质本身上的允许位置的端口,以及用于从存储介质读取的另一端口,而不再经过仲裁器。注意读取操作对应于存储介质的一个行为,由此,可以具有与对应于另一行为的写入操作不同的 URI,如能意识到,数据存取的这一模型比对每个操作,转到文件系统控制器更有效,特别是如果文件系统在其他方面是瓶颈的情况下。

[0009] 此外,除了展现状态和行为的端口的形式的 WSAP 的确定的交互作用点外,WSAP 消息使用描述能在交互作用点上做什么的共享语义。语义包括与(可能专用的)消息细节分离的共享消息动词,允许任何任意的中介在已知程度上观察正在发生什么,同时维持消息内容的安全。

[0010] 消息排序也能用来向观察者提供上下文。例如,每个服务可以定义它接收哪一 WSAP 操作,以及以哪种顺序。另外,与每个端口的通信通常识别正在请求的行为,因为每个行为可以分配其自己的端口。实际上可以维护为端口的列表的服务端口间的这些交互作用模式提供产生何种 WSAP 操作以及它们以何种顺序发生的易观察的目录。

[0011] 因此,中介体可以处理所交换的消息以便增加通信价值。例如,当从所请求的操作类型和/或已知 URI 了解到消息将不改变资源的状态,中介体,诸如高速缓存服务将从其自己的高速缓存安全地读取数据以便满足该请求,而不是使该高速缓存无效并传递消息(到另一服务)以便存取其他存储介质。注意如果已知是只读服务,中介体高速缓存服务仅需要从协议的语义知道操作类型,或 URI,而不是消息的内部内容,其仍然是安全的(以及的确,可以经另一中介体服务加密)。在不包含状态变化的 WSAP 中的类似这种读取的操作的例子包括称为使用 QUERY 或 GET 的共享语义的操作。这种操作每个具有安全,主要表示只读存取足以满足操作请求,以及等幂的,主要表示在不改变状态的情况下,不确定地重复相同操作。注意超高速缓存仅是这种中介体的一个例子,以及适当的话,中介体能是任何其他服务,例如负载均衡器、数据压缩器、复制服务、加密/解密机制等等。

[0012] 此外,因为行为具有它们自己的端口,提供 SOAP 处理模型,定义如何组成并行或顺序运行的多个网络服务。该模型允许实现几个服务,诸如感知 SOAP 的串行转换器,以便串行化或解串行化指定 SOAP 消息的头部和主体。只要服务所关心的头部可用,这允许根据 SOAP 包的部分来解决的服务。注意当两个通信方例如共同位于相同设备上时,由于性能原因,转发和传输服务不串行化 SOAP 包。

[0013] 从下述结合附图的详细描述,其他优点将是显而易见的,其中:

附图说明

[0014] 图 1 是通常表示可以包含本发明的计算机系统的框图;

[0015] 图 2 是通常表示根据本发明的一方面,在网络服务应用协议上,与另一方通信的服务的框图;

[0016] 图 3 是通常表示根据本发明的一方面,当传送 WSAP 消息时,单向消息交换的原理的框图;

[0017] 图 4 是通常表示根据本发明的一方面,用于 WSAP 消息的示例性请求-响应消息交

换模式的框图；

[0018] 图 5 是通常表示根据本发明的一方面，展现为具有不同行为的服务集的文件框图；

[0019] 图 6 是通常表示根据本发明的一方面，与是否包含相同服务值无关，模拟为不同服务的不同存取模型的框图；

[0020] 图 7 是通常表示根据本发明的一方面，在明确初始化后的 CREATE 操作中的消息的示例性序列的图；

[0021] 图 8 是通常表示根据本发明的一方面，用于复杂服务创建的消息的示例性序列的图；

[0022] 图 9 是通常表示根据本发明的一方面，在 LOOKUP 操作中，消息的示例性序列的图；

[0023] 图 10 是通常表示根据本发明的一方面，用于从 QUERY 操作映射到 GET 操作的序列的图；

[0024] 图 11 是表示根据本发明的一方面，为证明 INSERT 操作的结果在 INSERT 操作前后执行的 GET 操作的差异的序列图；

[0025] 图 12 是表示根据本发明的一方面，为证明 DELETE 操作的结果在 DELETE 操作前后执行的 GET 操作之间的差异的序列图；

[0026] 图 13 是表示根据本发明的一方面，为证明 UPDATE 操作的结果在 UPDATE 操作前后执行的 GET 操作之间的差异的序列图；

[0027] 图 14 是通常表示根据本发明的一方面，示例性 REPLICATION 序列的框图；

[0028] 图 15 是通常表示根据本发明的一方面，用于取消复制的示例性 CANCELREPLICATION 的序列的框图；

[0029] 图 16 是通常表示根据本发明的一方面，获得事件通知的示例性 SUBSCRIBE 序列的框图；

[0030] 图 17 是通常表示根据本发明的一方面，示例性 UNSUBSCRIBE 序列的框图；

[0031] 图 18 是通常表示根据本发明的一方面，简单服务终止的例子的框图；

[0032] 图 19 是通常表示根据本发明的一方面，通过另外的清除，简单服务终止的例子的框图；以及

[0033] 图 20 是通常表示根据本发明的一方面，复杂服务终止的例子的框图。

具体实施方式

[0034] 示例性操作环境

[0035] 图 1 示例说明可以实现本发明的合适的计算系统环境 100 的例子。计算系统环境 100 仅是合适的计算环境的一个例子以及不打算提出有关本发明的使用或功能性的范围的任何限制。也不应当将计算环境 100 解释为对于在示例性操作环境 100 示出的部件的任何一个或组合有任何依赖性要求。

[0036] 本发明用多个其他通用或专用计算系统环境或配置来操作。适合于与本发明一起使用的众知的计算系统、环境和 / 或结构的例子包括，但不限于：个人计算机，服务器计算机、手持或膝上型设备、图形输入板设备、微处理器系统、基于微处理器的系统、机顶盒、可

编程消费者电子设备、网络 PCs、小型计算机、大型计算机、包括任何上述系统或设备的分布式计算环境等等。

[0037] 在由计算机执行的计算机可执行指令,诸如程序模块的一般环境中,描述本发明。通常,程序模块包括例程、程序、对象、部件、数据结构等等,它们执行特定的任务或实现特定的抽象数据类型。本发明也可以在分布式计算环境中实现,其中,由通过通信网络链接的远程处理设备执行任务。在分布式计算环境中,程序模块加载在包括存储器件的本地和 / 或远程计算机存储介质中。

[0038] 参考图 1,用于实现本发明的示例性系统包括以计算机 110 的形式的通用计算设备。计算机 110 的部件可以包括但不限于处理单元 120、系统存储器 130 和将包括系统存储器的各种系统部件连接到处理单元 120 的系统总线 121。系统总线 121 可以是包括存储器总线或存储器控制器、外围总线和使用各种总线体系结构的任何一个的局部总线的几种总线结构的任何一个。举例来说,但不是限制,这种体系结构包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MCA) 总线、增强 ISA (EISA) 总线、视频电子技术标准协会 (VESA) 总线以及也称为 Mezzanine 总线的外围部件互连 (PCI) 总线。

[0039] 计算机 110 通常包括各种计算机可读介质。计算机可读介质能是能由计算机 110 存取的任何可用介质,包括易失和非易失介质,以及可移动和不可移动介质。举例来说,但不是限制,计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以任何方法或技术实现的易失和非易失、可移动和不可移动介质,用于存储诸如计算机可读指令、数据结构、程序模块和其他数据的信息。计算机存储介质包括但不限于 RAM、ROM、EEPROM、闪存或其他存储技术,CD-ROM、数字万用盘 (DVD) 或其他光盘存储器、磁带盒、磁带、磁盘存储器或其他磁性存储器件,或能用来存储所需信息并能由计算机 110 存取的任何其他介质。通信介质通常具体化为以调制数据信号,诸如载波或其他传输机制的计算机可读指令、数据结构、程序模块或其他数据,并包括任何信息递送介质。术语“调制数据信号”是指具有在信号中编码的方式设置或改变的一个或多个特性的信号。举例来说,但不是限制,通信介质包括有线介质,诸如有线网络或直接导线连接,以及无线介质,诸如声波、RF、红外和其他无线介质。上述的任何的组合应当包括在计算机可读介质的范围内。

[0040] 系统存储器 130 包括以易失和 / 或非易失存储器,诸如只读存储器 (ROM) 131 和随机存取存储器 (RAM) 132 形式的计算机存储介质。基本输入 / 输出系统 133 (BIOS),包含诸如在启动期间,有助于在计算机 110 内的元件间传送信息的基本例程,通常存储在 ROM 131 中。RAM 132 通常包含立即由处理单元 120 存取和 / 或当前正由处理单元 120 操作的数据和 / 或程序模块。举例来说,但不是限制,图 1 示例说明操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。

[0041] 计算机 110 也可以包括其他可移动 / 不可移动、易失 / 不可易失计算机存储介质。仅举例来说,图 1 示例说明读取或写入不可移动、非易失磁性介质的硬盘驱动器 141、读取或写入可移动、不可易失磁盘 152 的磁盘驱动器 151,以及读取或写入可移动、不可易失光盘 156,诸如 CD ROM 或其他光学介质的光盘驱动器 155。能用在示例性操作环境中的其他可移动 / 不可移动、易失 / 不可易失计算机存储介质包括但不限于磁带盒、闪存卡、数字多功能盘、数字视频带、固态 RAM、固态 ROM 等等。硬盘驱动器 141 通常通过不可移动存储器接口,诸如接口 140 连接到系统总线 121,而磁盘驱动器 151 和光盘驱动器 155 一般通过可移

动存储器接口,诸如接口 150,连接到系统总线 121。

[0042] 如上所述以及图 1 所示的驱动器和它们相关的计算机存储介质为计算机 110 提供计算机可读指令、数据结构、程序模块和其他数据的存储。在图 1 中,例如,硬盘驱动器 141 示例为存储操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147。注意这些部件能与操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137 相同或不同。操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147 在此给予不同的编号以示例说明至少,它们是不同的拷贝。用户可以通过输入设备,诸如图形输入板,或电子数字化仪 164、麦克风 163、键盘 162 和意指鼠标、跟踪球或触控板的定位设备 161,将命令和信息输入到计算机 110 中。在图 1 中未示出的其他输入设备可以包括操纵杆、游戏垫、卫星盘、扫描仪等等。这些和其他输入设备通常通过连接到系统总线的用户输入接口 160,连接到处理单元 120,但也可以由其他接口和总线结构,诸如并行端口、游戏端口或通用串行总线 (USB) 口连接。监视器 191 或其他类型的显示设备也可以经接口,诸如视频接口 190 连接到系统总线 121。监视器 191 也可以与触摸屏面板等等集成。注意监视器和 / 或触摸屏面板能物理地连接到包含计算设备 110 的外壳,诸如在图形输入板型个人计算机中。另外,计算机,诸如计算设备 110 也可以包括其他外围输出设备,诸如扬声器 195 和打印机 196,它们可以通过输出外围接口 194 等等连接。

[0043] 计算机 110 可以使用到一个或多个远程计算机,诸如远程计算机 180 的逻辑连接,在连网环境中操作。远程计算机 180 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其他公用网络节点,并通常包括相对于计算机 110 所述的许多或全部元件,尽管在图 1 中仅示例说明存储器存储设备 181。图 1 中所述的逻辑连接包括局域网 (LAN) 171 和广域网 (WAN) 173,但也可以包括其他网络。这些连网环境在办公室、企业范围计算机网络、内联网和互联网中是很普遍的。例如,在本发明中,计算机系统 110 可以包括从其迁移数据的源机器,而远程计算机 180 可以包括目的机器。然而注意,源和目的机器不需要由网络或任何其他装置连接,而相反,可以通过由源平台写入和由一个或多个目的平台读取的任何介质迁移数据。

[0044] 当用在 LAN 连网环境中时,计算机 110 通过网络接口或适配器 170,连接到 LAN 171。当用在 WAN 连网环境中时,计算机 110 通常包括调制解调器 172 或用于在诸如互联网的 WAN 173 上建立通信的其他装置。可以为内部或外部的调制解调器 172 可以经用户输入接口 160 或其他适当的机构,连接到系统总线 121。在连网环境中,相对于计算机 110 画出的程序模块或其一部分可以存储在远程存储器存储设备中。举例来说,但不是限制,图 1 示例说明驻留在存储器设备 181 上的远程应用程序 185。将意识到所示的网络连接是示例性的,可以使用能在计算机之间建立通信链路的其他装置。

[0045] 网络服务应用协议和 SOAP 处理模型

[0046] 在分布式编程模型中,基本的编程原语是轻便端口 (lightweight port),包括能用来存储状态和产生通信的无序消息集合。通过引用存储项,以及端口支持 (至少) 两个原语操作:Post() 和 Get()。登记项,诸如类范例缓存该引用并立即返回。如果没有由远程服务能登记具有该关键字的项,基于所提供的关键字检索项能阻止。尽管由服务直接使用端口,但它们通常在将特定消息类型与集合中的每个端口关联的端口集合的环境内使用。根据所需的灵活度,用固定或动态的支持消息类型集,创建端口集合,但从程序员的视点看,

端口集合简单的是打字端口。

[0047] 与本发明一致,使用严格的端口通信,服务码能实现关键部件以及复杂的 I/O 协议以及产生非常同时的处理模型。为此,本发明部分针对用于与网络服务通信的协议(网络服务应用协议或 WSAP),其中,客户机消费者请求服务提供者为该客户机执行一些服务,以及为客户机提供适当的响应。然而,如将理解到,通用网络服务模型不限于服务器为客户机运行软件,而是应用于客户机希望访问的任何资源,诸如由 WSAP 支持的统一服务模型能用来模拟文件、设备、服务和人。例如,可以某种程度上组件化硬件,以及在许多方面,实际上,将与面向软件的网络服务不可区分,因为用户可以选择硬件部件集以及将它们互连以执行计算任务。只要每个设备服从通信协议,(以及适当的带宽量可用),实际上,任何授权的设备将能与另一设备通信来使用它的资源。

[0048] 图 2 表示包括经在 SOAP(简单对象访问协议)包中传送的 WSAP 消息,与网络服务 204 通信的网络客户机码 202(例如,可以是图 1 的计算机系统 110)的体系结构 200。用命名端口(例如 URI)描述服务的身份,而通常由 XML 模式来描述服务的状态。注意,服务可以是本地(在与客户机码相同的机器内)或可以是远程资源,诸如服务器或硬件设备。通常,如下所述,WSAP 语义上定义网络服务做什么,而 SOAP 处理模型定义如何组成并行或顺序运行的多个网络服务,以及可以根据本发明使用来经基本上与外部网络服务相同的方式,例如经存储器内非 XML 串行化数据,组成对计算设备的内部服务。

[0049] 关于其他网络服务,WSAP 可能的服务通过响应订约请求,提供订约来操作,如在图 2 中所示,通过网络服务订约请求 206 和订约提供 208。注意,在 WSAP 中,为获得订约,将端口展现给对应于 LOOKUP 操作的客户机,如下所述。也如下所述,向捆绑列表信息提供响应 208,例如,作为对应于“示例”订约的关键字/值对的数组的标识符/端口,如下所述。不要求服务来实现不是 LOOKUP 操作的 WSAP 操作,,尽管当交互作用的语义与 WSAP 操作兼容时,WSAP 服务应当实现 WSAP 操作。在提供服务中,服务发起人可以使用 WSAP 动词和表示,映射任何交互作用或服务功能性。并非所有 WSAP 动词必须用于特定的服务,而相反,根据情况,可以使用子集。

[0050] 根据本发明的一方面,WSAP 是用在借助能根据服务的行为使用的身份、状态和操作集,定义网络服务的分布式计算环境中的基于 SOAP 的协议。这有效地允许服务和它们的状态以一致和可观察方式操作。在 WSAP 中,行为网络服务是具有身份和行为的过程的抽象概念。该概念在能应用于许多不同应用模型和名字空间方面是抽象的。WSAP 定义一种这样的应用模型,它已经设计成扩大到整个 Internet,但许多其他的应用模型也是可能的。

[0051] WSAP 中的身份的概念与由传统的网络模型中的 URI 识别的资源的概念类似,即具有身份的任何事。身份的目的是允许服务将其自己与可能存在于相同名字空间中的其他服务区分开来。如在此所述,WSAP 网络服务的范围可以是 Internet,表示 DNS 或 IP 是用于 URI“授权”部件的命名授权,尽管在更有限的范围内,诸如在设备或有限网络内,另一命名授权(naming authority)也是可行的。

[0052] 每个 WSAP 网络服务具有唯一身份,诸如与在图 3 中示例的 URI。关于任何其他 URI(诸如网络上),仅基于 URI,不能获得语义知识,以及类似地,不能从它们的 URIs 推导两个 WSAP 网络服务之间的关系(如牵制关系)。在 WSAP 中,URIs 可以是相对或绝对的。如果 URI 是相对的,那么由 XML 基础(XML base)确定其基础。WSAP 对 URI 的长度没有任何限

制,由此任何 WSAP 接收机需要能处理其公布的任何 URI 的长度;WSAP 实现应当能处理至少两个千字节长度的 URIs。

[0053] 服务间的交互作用(能视为协议)是服务的行为,并能形式上描述为行为类型订约是使用行为类型的服务的描述。能用各种方式执行服务,只要符合它要求实现的订约。注意在 WSAP 中,订约是象基本上任何其他事的服务,由此由端口(例如 URIs)识别订约以及能根据其订约,如任何其他服务那样操作。

[0054] 行为的概念基于根据它生成和接收的消息类型的序列,服务与其环境如何交互作用。行为允许服务将其特定的行为与可能存在于相同名字空间中的其他服务的行为区分开来,以及行为相对于为定义服务的特定名字空间定义的可能行为集。换句话说,行为是由该服务展现的分布式状态机。因此,通过包括那些资源或服务与其他服务的交互作用,行为增加用于识别资源的当前网络模型。两个服务需要具有兼容的行为以便通信,即,它们必须达成和解。如果服务的行为不由另一服务已知,这暗指两个服务不能交互作用,其类似于资源的网络模型,其中因为用于执行该操作的机制对那个 URI 的支持器是未知的,因此,URI 不能被解除参考。

[0055] 从 WSAP 服务视点看,任何数据,不管是元数据还是数据,均表示为服务。即,不是将相同的数据集合(诸如文件)展现为能读取或执行的单一实体,将文件展现为具有不同订约的不同服务。这在图 3 中示出,其通过展现为具有不同行为的服务集的传统的数据 302,示例说明身份和行为的唯一性的概念,以及每个服务具有其自己的 URI。也如图 3 所示,在对应于文件的行为的服务中,包括读取服务 304、执行服务 306、元数据操作服务 308 和用于读取该文件与其他服务 310 的关系的目录的服务。实质上,关于文件的任何其他行为(例如写入)也能是服务,以及在图 3 中隐含表示。

[0056] 将文件展现为服务集,每个具有其自己的 URI,与传统模型比这提供显著的好处,在传统模型中文件通过打开后返回的句柄,经文件系统访问。例如,通过本发明,只要到介质服务的 URI 是已知的,经适当的服务,执行对文件的操作,消除继续通过文件系统用文件数据执行一些操作的需要。这对分布式计算环境特别有利,其中例如,文件系统可以在一些遥远的服务器上和/或可以是经本发明能避免的瓶颈。

[0057] WSAP 的主要目的是提供应用层协议框架,它解决 WSAP 网络服务是什么以及如何识别它;WSAP 网络服务的行为是什么和它如何与其他 WSAP 网络服务通信;以及 WSAP 网络服务一般如何与其他 WSAP 服务以及其他资源关联。由 WSAP 提供的框架,经消息操作和交换模式集,意图在身份和行为方面约束行为网络服务的概念,但它不在结构或复杂性方面约束它。WSAP 网络服务能用在许多不同的应用环境中。例如,除如图 3 的例子中所示,作为服务的文件的概念外,身份和行为的唯一性的其他简明例子是可行的,诸如通过向设备的不同行为,或向简单聊天服务的不同行为分配端口。注意为了简单,在此的例子使用相对低级的实体和服务,然而,应理解到,该模型同样适合于更复杂的实体,诸如商业实体和它们的关系。

[0058] 除服务身份和服务行为外,行为的网络服务具有服务状态,有时称为“服务价值”的概念。更具体地说,WSAP 消息操作可以包含表示服务的部分或子部分的串行化状态的交换(而不是 WSAP 服务本身)。服务的串行化状态称为服务价值,与表示服务类型的行为相反。服务能具有由服务的行为允许的任何值,以及那一价值可以随时间改变。注意 WSAP 在

服务上操作,以及采用表示服务的串行化状态的服务表示。服务可以具有零个或多个可能的表示,即,对服务来说,可以不具有任何表示。WSAP 操作可以包含服务表示的交换,而不是服务本身的交换。服务表示可以随时间改变以及可以作为导致将生成的表示的条件的函数改变。应用通过表示的交换,而不是通过服务的交换交互作用。

[0059] 图 4 示出服务价值 403,其中,服务价值能是任何数据(例如文件的状态数据),诸如数据库数据、字母数字数据、房屋的图像或花的图像。从服务视点看,相同的服务值可以表示为具有不同交互方式的不同服务。如上所述,因为将不同行为模型化为分别的服务,影响诸如访问权限以及通信属性的行为的属性能产生不同服务。

[0060] 在图 4 中,对相同服务价值的不同访问模型模型化为不同服务 411-414,分别包括只读访问、读写访问、加密访问和谈判访问。注意相关服务间的同步能使用 WSAP 复制模型来执行,如下所述。

[0061] 考虑图 3 和 4,可以看出,文件系统是包含多个数据的“视图”,即文件的实体的一个例子,包括元数据(诸如文件名、长度、所有者和存取许可)和数据(诸如文件内容)。另外,文件的视图由打开文件的实体的存取许可而定,以及实际上,只读文件的行为不同于读写文件的行为。同样地,文件系统易于模型化为行为服务集。

[0062] 只要经兼容订约连接服务,服务能通过生成和接受特定类型的消息序列,彼此交互。为此,WSAP 用相关语义定义一起消息操作。每个消息操作包括能在两个通信方向间交换的一组特定类型的消息。

[0063] 根据本发明的另一方面,WSAP 构建有关行为网络服务的定义,其中,根据消息操作和消息交换方的共享集,定义用于确定身份和行为的框架。由通常为应用程序的客户机代码 202(图 2),定义 WSAP 消息操作,其中操作包含使用由 WSAP 定义的一个消息交换模式的,交换特定类型的消息,如下面参考图 5 和 6 所述。也如下所述,端口通过通信提供同步,以及 SOAP 提供用于在服务间交换的消息的处理模型。

[0064] WSAP 的目标是提供用于定义网络服务和它们的交互作用的公共应用层基础。注意将 SOAP 设计成支持多个应用模型,以及应用免于使用 SOAP 可扩展性模型的全部型谱。然而,WSAP 的一个意图是定义通常用于各种分散应用的应用模型。WSAP 的另一目标是允许高度互操作性,导致潜在增加整个系统的有效性。

[0065] 为实现这些目的和其他,WSAP 基于一组公共消息操作,定义网络服务应用模型。注意,迄今还未知应用于网络服务的公共词汇的概念 WSAP 通过听起来熟悉的名称,诸如 LOOKUP、QUERY、INSERT、UPDATE、DELETE、GET 和 DROP,定义操作的整个结构和语义,由此各个服务能由那些操作继承而来,同时保存对稍后捆绑的支持。通常,大写字母常常在此用于 WSAP 操作,以便将 WSAP 操作与传统的动词区分开来,但不要求操作使用大写字母,以及另外,其他或不同命名的 WSAP 操作集也是可行的,并且服务能根据需要定义它们自己的附加操作。

[0066] 操作允许应用与其他服务通信。例如,LOOKUP 以订约的形式,提供有关服务的信息,能使用 GET 和 QUERY 来检索服务的当前状态。CREATE 和 DROP 操作能创建和删除服务等,如下所述。大部分操作在特定类型的消息上是多形的,以致各个服务能使消息的结构适合于服务的形状。例如,由目录服务定义的 INSERT 请求消息的 SOAP Body 元素与由载入程序服务定义的 INSERT 请求消息不同,但它们遵循相同的基本语义。

[0067] WSAP 消息操作能视为能组成为更复杂订约的小型订约。这允许应用将消息操作组合成适合于它们的特定需要的序列,因此,构建更复杂的服务。换句话说,能组成和 / 或排序 WSAP 操作本身以便定义用于服务的更复杂的行为。例如,服务可以定义为接受在 UPDATE 操作前的 QUERY 操作 (QUERY 和 UPDATE 操作描述如下)。注意,由此 URI 能表示 WSAP 操作序列,表示 URI 本身能表示单一行为或复杂的行为,尽管 WSAP 不要求,能使用订约来更有利于实现所需兼容性。

[0068] 除网络的外围中潜在的增值外,使用可能允许整个系统的更高透明度的共享语义。因此,在整个系统中,能配置诸如高速缓存和其他中介体处理服务的性能优化以便实现 Internet 广泛系统所需的可缩放性。

[0069] 更具体地说,WSAP 操作具有允许中介体,诸如高速缓存服务、负载均衡器服务、数据压缩服务、复制服务、加密 / 解密服务等等处理消息的服务。限定 WSAP 操作的两个这种属性包括操作是否“安全”和 / 或“等幂”。当成功执行的结果与执行操作的次数无关时,操作是等幂的。如果其语义局限于只读访问施加到其上的资源,操作是安全的。注意,按定义,安全操作是等幂的,然而,逆过程不为真,由此只读操作是既安全且等幂。另外,这些限定公司应用于操作,而不是该操作的特定实现。

[0070] 这些限定公司提供启动操作方和目标服务间的服务级期望。注意特定的安全或等幂操作实现可以具有与执行该操作有关的第二阶副作用。这种副作用的典型例子是登录、在动态内容的情况下,与生成只读数据有关的处理、状态管理更新等等。这种第二阶副作用不影响安全和等幂操作的概念。

[0071] 因此,中介体服务能处理所交换的消息来增加通信价值。举例来说,考虑中介体,诸如高速缓存。在该例子中,客户机连接到提供数据存取的服务,并在响应中,客户机接收 URI 到高速缓存。为响应 GET 和 / 或 QUERY 请求,返回数据,超高速缓存不需要知道有关数据的内容的任何事,例如内部如何格式化、是否加密等等,但从请求操作类型知道处理消息将不改变高速缓存下,基础资源的状态。因此,高速缓存服务能安全地从其自己的高速缓冲存储器读取数据以满足任何 GET 和 / 或 QUERY 请求。如果接收到将改变状态的请求,例如 WSAPINSERT、DELETEU 或 PDATE 操作,高速缓存服务能将消息传递到适当的 URI,或者,客户机能使用不同的 URI 来修改状态,在那样事件中,高速缓存服务将被通知其某个部分无效。

[0072] 根据本发明的另一方面,WSAP 操作基于两个消息交换模式,即单向和请求 - 响应,以及在最终响应到来之前,支持中介体响应。尽管 WSAP 消息潜在的能在允许多播的体系结构上传播,可能具有 WSAP 消息时,WSAP 消息操作的语义是点对点通信。注意,WSAP 以形式化方式,描述网络服务的行为,这是 WSAP 和其他协议的主要区别。

[0073] 图 5 和 6 分别表示使用单向和请求 - 响应消息交换模式的 WSAP 操作。在任一情况下,发送者和接收者均命名 WSAP 服务 (即,经 URL)。注意这不同于客户机 / 服务器协议,诸如 HTTP,其中,仅命名请求的目标资源。

[0074] 在图 5 中表示单向消息交换模式,并包括通过零个或多个中介体,从初始发送者发送到最终接收者的消息。注意如果单向消息导致生成 SOAP 故障 (如下所述),不将 SOAP 故障消息返回到初始发送者。

[0075] 请求 - 响应消息交换模式如图 6 所示,并包括在通过零个或多个中介体,从初始发送者向最终接收者发送请求消息,然后是也通过零个或多个中介体从最终接收者发送到初

始发送者的一个或多个中间响应。注意,没有对该消息交换模式强加时间约束,以及支持该消息交换模式的操作能用在同步和异步通信模型中。另外,注意如果请求消息导致生成 SOAP 故障,如下所述,将 SOAP Fault 消息返回到初始发送者,然而,如果响应消息导致生成 SOAP 故障,生成 SOAP 故障,但不将 SOAP 故障发送到响应者。

[0076] 根据本发明的一方面,WSAP 操作包括具有如 SOAP Body 内容的消息类型、这些消息类型的行为以及通过调用操作隐含的语义的集合。对 SOAP 消息,WSAP 消息类型能由可扩展数目的 SOAP 特征,(通常表示为 SOAP 首部块)组成。这些特性的例子包括但不限于安全性、路由和可靠性。

[0077] 下述 C# 类(用在如 IRP 数据结构的一些情形中)表示一个实现中的 SOAP 包:

[0078] msgSoapEnvelope

[0079] {

[0080] IPortCollection portHeader;

[0081] IPort portBody;

[0082] }

[0083] 处理 SOAP 包包括消息生成、消息交换和消息解析。由 SOAP 处理组中的中介体使用 PortHeader 端口来交换能串行化或严格的内部首部结构。第二端口 PortBody 存储消息体。使用该容器内的端口允许 SOAP 包的同时和线程安全的操作。不要求严格的流水线模型或基于堆栈的模型(通常具有处理 SOAP 首部块和/或 SOAP 体间的交互作用的问题)。

[0084] SOAP 处理模型的基于 WSAP 实现的好处在于它匹配用于在同时处理一个 SOAP 消息的 SOAP 中的内在支持。经对应于行为的端口,实现便于通过几个服务,同时处理消息。例如,感知 SOAP 的串行化器可以与其他服务操作同时地串行化或并行化指定 SOAP 消息的首部和主体。这允许依赖于 SOAP 包的部份的服务只要服务关心的头部可用,解开块。由于该并行解析设计,SOAP 并行化本身不是原子的。为更好性能,转发和传输服务永远串行化 SOAP 包,只要两个通信方定位一起。

[0085] 由于使用 SOAP,使用 SOAP 故障结构,使用请求-响应 MPE 报告在执行 WSAP 操作的任何一个中的故障,以及由 WSAP 应用生成的 SOAP 故障需要包括其值反映带有 S:Envelope/S:Body/S:Fault/S:Code/S:Value 的值的 SOAP 故障的 SOAPAction 元素信息项。通常,每个故障有单独的动作。

[0086] 根据本发明的一方面,能识别 WSAP 消息类型。不关心 SOAP Body 的特定类型,或不能看见 SOAP Body 的内容(例如如果已经加密过)的中介体和其他 SOAP 节点能使用 SOAP Action 元素信息项,例如,由 WS-Addressing 所具有的。WSAP 操作定义唯一 Action 值。

[0087] 不关心 SOAP Body 的特定类型的 SOAP 节点能使用 SOAP Body 的最外元素信息项的 XML 扩展名的类型。WSAP 定义需要替换由应用定义的消息类型的抽象元素信息项集。

[0088] 对 SOAP 故障,模型是类似的,但使用 SOAP 故障码提供增加的检查能力级,因为不关心 SOAP 故障的细节的中介体和其他 SOAP 能检查 SOAP 故障码。

[0089] 除如上所述的操作外,展现状态允许能一致地应用于所有服务的更少传统操作,诸如高速缓存和复制以及统一公开/预约(公开/预约)模型。能使用此来构建能测试、监视和管理在分布式操作系统节点和节点之间的服务的一般应用。

[0090] 也能将会话模拟为 WSAP 中的服务,其中会话管理是指在根据呼叫者的值状态之

间作出区分。作为会话管理的一个例子,HTTP Cookie 通常用来根据独立 HTTP 操作上的用户,个性化网页。然而,Cookie 的问题在于服务的状态以及可能的行为由 Cookie 的值而定,它与行为网络服务模型不一致。然而,因为 WSAP 将会话模拟为另一服务,而不是使用会话标识符或 Cookie 来识别会话,会话是具有状态和行为的的服务,正如创建、操作和终止的任何其他服务。

[0091] 除如上所述的服务交互作用外,服务能以不包含 WSAP 消息的实际交换的方式,彼此关联。这种关系的例子包括(但不限于)将服务集描述为属于相同安全组的安全性政策,以及描述特定方如何管理服务集的管理语句。在这种示例性的情况中,描述指一组服务,但描述不包含所涉及的服务间的任何实际的通信。然而,捕捉关系仍然很重要,即使未交换显式 WSAP 消息。

[0092] 可以将这些关系模拟为行为类型以及将它们描述为订约的一部分。这些订约称为“图订约”,以便将这些关系订约与 WSAP 消息订约区分开来。使用与其他订约相同的描述语言,描述图订约。

[0093] 转到共享语义的说明,通常,由其他服务创建服务。更具体地说,能使用 CREATE 操作创建服务,它通过拷贝现有的服务创建新服务。为此,新创建的服务具有创建服务的服务行为和服务价值,即接受 CREATE 操作的服务充当用于新创建的服务的价值的模板。CREATE 请求可以包含用来初始化新创建的服务的另外的状态。这允许通过其他不变状态初始化服务。CREATE 请求可以包含根据服务的形状,修改新创建的服务的拷贝状态的状态。将所提出的终点命名为 CREATE 请求的部分或使创建服务提供名称是可行的。注意继承服务可以展现为网络服务,而不直接创建为网络服务。

[0094] 在 WSAP 中,支持 CREATE 操作的服务能创建特定行为的服务。任何服务可以支持 CREATE 操作,由此可以支持根据需要,创建许多行为。因为 WSAP 操作与显式服务关联,CREATE 服务是用于创建新服务的目标服务的请求。CREATE 本身不创建目标服务。

[0095] 图 7 表示 CREATE 操作如何能导致创建一个或多个相关服务。在简单的服务创建中,CREATE 操作在显式初始化之前的创建序列,创建单个服务。在图 7 中,客户机 702 通过处理这些操作的服务 710,启动 CREATE 操作。一般而言,关于网络服务,CREATE 请求包含用于将创建的服务类型的订约。创建服务包括为服务分配 URI 以及将该服务与指定行为关联。在已经创建服务后,另外的消息操作可以包含根据服务的结构改变服务值,和 / 或通过关系、存取控制或其他机制,将新服务与其他服务关联。

[0096] 响应包含即时订约,它包含新创建的服务 720 的 URI。在已经创建服务后,根据其行为和如果适当的话,与其他服务的关系操作。例如,服务能通过操作处理这些政策的服务,与安全性政策关联。

[0097] 下述表阐明用于 WSAP CREATE 操作的属性:

[0098]

用于 CREATE 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/createrequest
响应动作	http://abc/def/ghij/10/wsap/createrequest
安全	否
等幂	否

[0099] 下述数据结构包含样本 CREAT 操作的 SOAP Body 内容。该请求由服务定义,响应使用为空响应而定义的方便的类型:

[0100]

```
<M:MyServiceCreateRequest xmlns:M="http://www.example.org/myservice">
  <M:MyServiceInfo>
    <M:Status>ok</M:Status>
    <M:MaxEntries>100</M:MaxEntries>
  </M:MyServiceInfo>
</M:MyServiceCreateRequest>
<W:DefaultCreateResponse xmlns:W="http://abc/def/ghij/10/wsap.xsd"/>
```

[0101] 在更复杂的服务创建中, CREATE 操作创建一组相关服务。例如,如果创建服务知道以它们总是一起存在的方式关联服务集,会出现复杂的服务创建。图 8 是示例说明两个服务的创建的序列图。在图 8 中,如在简单的服务创建情况下,客户机 802 通过处理这些操作的服务 810(创建服务 A)启动 CREATE 操作。操作请求包含用于将创建的服务类型的订约。创建服务 A 服务 810 相对于创建服务 B(在图 8 中标记 812),启动另一 CREATE 操作以便创建另一新服务 824,服务 B。

[0102] 在已经创建服务 A 和服务 B(822 和 824)后,创建服务 B 服务 812 向创建服务 A 服务 810 发送创建响应,810 又向客户机 802 发送创建响应。由创建服务 A 服务 810 发送的创建响应包含即时订约,它包含两个新创建的服务 822 和 824 的 URIs。注意在复杂的服务创建中,WSAP 不提供固有保证,即在事务环境内创建服务。这些保证能通过增加具有事务支持的服务的行为来提供。

[0103] 如上所述,WSAP 服务的行为描述为订约。订约通常被串行化为 XML 文档,尽管不要求。订约本身是服务,同样地,订约由 URI 识别以及能根据其订约操作,正如任何其他服务那样。

[0104] 服务可以具有另一服务的 URI,但不是其订约,这是两个服务通信所需要的。为获得订约,WSAP 协议要求每个服务支持允许服务向另一服务请求订约,从而确定服务的行为的 LOOKUP 操作:

[0105]

用于 LOOKUP 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/lookuprequest
响应动作	http://abc/def/ghij/10/wsap/lookuprequest
安全	否
等幂	否

[0106] 下文表示样本 LOOKUP 操作的 SOAP Body 内容。由 WSAP 定义请求和响应：

```
<W:LookupRequest xmlns:W=" http://abc/def/ghij/10/wsap.xsd" />
<W:LookupResponse xmlns:M=" http://abc/def/ghij/10/wsap.xsd" >
  <W:Contract>http://www.example.org/myservice</W:Contract>
  <W:BindingList>http://www.example.org/mybl</W:BindingList>
</W:LookupResponse>
```

[0107] 图 9 表示示例说明客户机 902 和服务 920 间的 LOOKUP 操作的样本序列。查找响应包含订约的 URI,如上所述,它也是服务,尽管返回订约本身也是可行的。

[0108] 因此,在一个实现中,LOOKUP 将 URI 返回到包含订约 / 协议的串行化的文档。订约不仅定义消息的顺序,而且定义由在其上交换消息的具有 URIS 的端口所表示的交互作用点。另外,使用那个文档加上为关键字 / 值对的数组的 BindingList 来“例示”订约文档。这允许在运行时,由真值替换一般名称,以及允许在运行时,确定服务的实例间的关系,提供服务的实例间的反映能力。订约 / 捆绑列表从而允许管理分布式设定值中的依赖性,同时,在启动时,指定所需服务。

[0109] 对服务状态检索,WSAP 协议定义两个操作,名为 GET 和 QUERY。尽管两个操作是安全和等幂的,GET 和 QUERY 操作间的差别在于如下所述,QUERY 请求包含结构化查询,而 GET 请求是用于服务的当前状态的快照的与应用无关的请求。因此,QUERY 请求要求有关正查询的服务的特定的模式知识,而 GET 请求不需要。这种区别的结果在于 QUERY 请求包含不为服务的名称(结构化查询)的一部分的参数,而 GET 请求提供将服务识别为服务的名称的一部分的参数。

[0110] 通过支持 QUERY 和 GET,WSAP 支持结构化标识符以及隐性标识符,以及允许这些解决方案间的映象。这是因为根据它们的用途的环境,对任何一个都有利。更具体地说,在命名服务中,在使用结构化名称,诸如查询和语义隐性名称,诸如 URIs 间存在紧张关系。例如,考虑识别为能提供人名的结构化表示的“http://example.com/people”的示例性服务。至少用三种不同方式,引用特定人,即作为结构化查询、作为在 URI 查询构件中编码的非结构化查询,或作为 URI 路径构件的一部分的隐性标识符。针对服务发出的结构化查询需要有关服务的结构的知识以便引用任何特定人,如在相对于示例性服务“http://example.com/people”发出的结构化查询的下述例子中所示。

[0111]

```
<Query>
  <Name>
    <First>Santa</First>
    <Last>Claus</Last>
  </Name>
</Query>
```

[0112] 相反,作为 URI 查询构件的非隐性部分编码的非结构化查询通常使用 URIHTML 形式编码。该模型要求有关名称-值对的集合的知识,但不要求有关特定服务的结构化知识,以便组成编码这些名称-值对的 URI。下述例子表示在 URI 查询构件中,编码为名称-值对的非结构化查询:

[0113]

```
http://example.com/people? first = santa&last = claus
```

[0114] 作为 URI 路径构件的一部分的隐性标识符不透露,实际解可以不包含查询。客户机需要了解的唯一的事情是如何解析标识符,在这种情况下,通过使用 HTTP。下述例子表示那种隐性标识符,它不表示查询的任何部分。

[0115]

```
http://example.com/people/12345678
```

[0116] 查询解决方案的好处在于能完全分类型查询参数。然而,结果是结构化查询变成用于该人的标识符的一部分。这表示想打听那个人的任何实体需要使用和理解特定的结构化名称。识别其他事物的结构化名称将具有也要求特定知识以便使用的不同的形状。在一些情况下,这工作良好,特别是当用户期望在该数据上操作时。然而,因为需要特定的知识,可缩放性会快速变成问题,因为难以配置在具有有关名称的那个特定形状的特定知识的组之外的名称的概念。

[0117] 标识符间的另一重要差别是存在的数据隐藏的级。更具体地说,查询解决方案要求查询的结构的知识,而非结构化编码查询和隐性解决方案逐渐提供更少信息。如果存在能变成公开的某些关键字,特别是如果有助于降低必须映射的 URIs 的数量,能使用非结构化编码查询。这些应用的典型例子是公众搜索引擎,诸如 Google(tm) (“http://www.google.com/search? hl = en&ie = UTF-8&oe = UTF-8&q = msft”) 以及典型的股票指数服务 (“http://moneycentral.msn.com/scripts/webquote.dll? iPage = qd&Symbol = msft”)。

[0118] 非结构化编码查询解决方案是今天 Web 上最通用类型的标识符,但非常难以编码 URI 中的结构化数据。此外,编码成 URI 的一部分的数据对体系结构来说本来是公开的,使得它更难以处理用于作为标识符的一部分的任何数据的与安全性有关的问题。隐性标识符解决方案不支持查询界面,除非存在用于将查询映射到那个标识符的模型。如果查询的结构仅对服务的一部分潜在用户知道(例如,由于隐私或其他原因),这能通过使用非结构化编码查询方案或隐性标识符方案,不展现或展现结构化数据的受限部分来完成。

[0119] 因此,能使用 GET 操作来检索服务的串行化值状态,并具有下述属性:

[0120]

用于 GET 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/getrequest
响应动作	http://abc/def/ghij/10/wsap/getrequest
安全	是
等幂	是

[0121] 下述表包含样本 GET 操作的 SOAP Body 内容。由 WSAP 定义请求,由服务定义响应：

[0122]

```
<W:GetRequest xmlns:M=" http://abc/def/ghii/10/wsap.xsd" />
<M:MyServiceGetResponse xmlns:M=" http://www.example.org/myservice" >
  <M:MyServiceEntries>
    <M:Entry>http://www.example.com/first/endpoint</M:Entry>
    <M:Entry>http://www.example.com/second/endpoint</M:Entry>
    <M:Entry>http://www.example.com/third/endpoint</M:Entry>
    <M:Entry>http://www.example.com/fourth/endpoint</M:Entry>
  </M:MyS erviceEntries>
</M:MyServiceGetResponse>
```

[0123] 在一些情况下, QUERY 请求能映射到 GET 操作,例如,这种映射有效地为特定查询提供标识符。例如,当在许多不同环境中,出现相同查询时,或当需要 URI 引用以便引入高速缓存时,这提供有利方面。图 10 表示示例说明从 QUERY 映射到 GET 的样本序列。QUERY 响应包含用于能针对 A 提供与 QUERY 相同的结果的服务 B 的 URI。注意能通过任何服务进行映射,并不必与针对以 QUERY 为目标的服务有关。

[0124] 因此,也能使用 QUERY 操作来查询服务的串行化值状态,并具有下述属性：

[0125]

用于 QUERY 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/queryrequest
响应动作	http://abc/def/ghij/10/wsap/queryrequest
安全	是
等幂	是

[0126] 下述表包含样本 QUERY 操作的 SOAP Body 内容。请求和响应均由服务定义：

[0127]

```
<M:MyServiceQueryRequest xmlns:M=" http://www.example.org/myservice" >
  <M:Entry>http://www.example.com/eight/endpoint</M:Entry>
</M:MyServiceQueryRequest>
<M:MyServiceQueryResponse xmlns:M=" http://www.example.org/myservice" >
  <M:MyServiceEntries/>
</M:MyServiceQueryResponse>
```

[0128] 注意,因为 GET 和 QUERY 操作是安全且等幂的,并且已知具有通过共享语义的可观察属性,各种中介体能提供消息交换的值。例如,高速缓存服务知道 QUERY 或 GET 请求不改变状态。因此,高速缓存能响应这些请求,无限地继续提供状态数据,除非以及直到检测到改变状态的操作为止。

[0129] 除状态检索外,WSAP 允许使用 INSERT、UPDATE 和 DELETE 操作更新服务的状态。这些操作仅适用于服务的值,它们不创建或终止服务。

[0130] 由服务将 WSAP 服务的数据结构定义为其行为的函数。因此,INSERT、UPDATE 和 DELETE 操作要求请求者具有有关其结构的知识以便影响服务的值,与 QUERY 类似。因此,抽象地定义 INSERT、UPDATE 和 DELETE 操作以便它们能适合于各个服务。

[0131] 当使用 INSERT 操作修改服务值时,包括在 INSERT 请求中的值增加到服务值上。假定隔离观察服务,通过查看在 INSERT 操作前发出的 GET 操作和在 INSERT 操作后发出的 GET 操作间的差别,检测 INSERT 操作的结果,通常如图 11 的消息序列图中所示。在图 11 中,在插入操作前,客户机 1102 从服务 1120 接收状态数据,以及在插入操作后的下一请求之后,接收更新的状态数据。

[0132] 能使用 INSERT 操作来将串行化值状态增加到服务上,如通过下述属性所述：

[0133]

用于 INSERT 的属性表	
属性	值
MEP	请求-响应
响应动作	http://abc/def/ghij/10/wsap/insertrequest
请求动作	http://abc/def/ghij/10/wsap/insertrequest
安全	否
等幂	否

[0134] 下述表包含样本 INSERT 操作的 SOAP Body 内容。由 WSAP 定义请求,由服务定义响应。例子不表示将插入数据的任何排序,然而,应用可以定义它们的模式来适应排序约束:

[0135]

```
<M:MyServiceInsertRequest xmlns:M="http://www.example.org/myservice">
  <M:MyServiceEntries>
    <M:Entry>http://www.example.com/fifth/endpoint</M:Entry>
    <M:Entry>http://www.example.com/sixth/endpoint</M:Entry>
  </M:MyServiceEntries>
</M:MyServiceInsertRequest>
<W:DefaultInsertResponse xmlns:M="http://abc/def/ghij/10/wsap.xsd" />
```

[0136] 通过那一服务定义服务的结构,而 WSAP 不要求以任何方式插入该值。作为其结构的一部分,服务可以允许 INSERT、UPDATE 和 DELETE 操作来识别将执行操作的特定位置,例如,使用 XPath 表示。

[0137] 关于 INSERT,当使用 DELETE 操作修改服务值时,应用相同的一般观察特性,其中,删除在 DELETE 请求中识别的服务值的部分。假定正隔离观察服务,通过查看在 DELETE 操作前发出的 GET 操作和在 DELETE 操作后发出的 GET 操作间的差别,能检测 DELETE 操作的结果,如一般在图 12 中所示。

[0138] 下述表表示用于 DELETE 操作的属性:

[0139]

用于 DELETE 的属性表	
属性	值
MEP	请求 - 响应
请求动作	http://abc/def/ghij/10/wsap/deleterequst
响应动作	http://abc/def/ghij/10/wsap/deleterequst

安全	否
等幂	否

[0140] 通过服务定义 DELETE 请求,响应使用为空响应而定义的方便的类型:

[0141]

```
<M:MyServiceDeleteRequest xmlns:M="http://www.example.org/myservice">
  <M:MyServiceEntries>
    <M:Entry>http://www.example.com/some/endpoint</M:Entry>
    <M:Entry>http://www.example.com/another/endpoint</M:Entry>
  </M:MyServiceEntries>
</M:MyServiceDeleteRequest>
<W:DefaultDeleteResponse xmlns:W="http://abc/def/ghij/10/wsap.xsd"/>
```

[0142] 当使用 UPDATE 操作,修改服务值时,用包括 UPDATE 请求中的值,替换在 UPDATE 请求中识别的服务值部分。注意如果在事务上下文中看 UPDATE 操作与在 INSERT 前的 DELETE 类似。即,仅当如通过 UPDATE 操作所述那样,成功地修改现有的服务,UPDATE 才成功。关于 INSERT 和 DELETE,通过查看在 UPDATE 操作后发出的 GET 操作和在 UPDATE 操作后发出的 GET 操作间的差别,能隔离地检测 UPDATE 操作的结果,如通常在图 13 中所示。

[0143] 使用 UPDATE 操作来更新现有的值状态,以及具有下述属性:

[0144]

用于 UPDATE 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/updaterequest
响应动作	http://abc/def/ghij/10/wsap/updaterequest
安全	否

[0145] 下述表包含样本 UPDATE 操作的 SOAP Body 内容。通过 WSAP 定义请求和响应动词,关于请求,SOAP 包的主体包含服务定义的响应元素。

[0146] 在该例子中,响应是为空响应而定义的方便的类型。WSAP 不规定那样的机制,通过该机制,识别现有状态,相反,留下直到该应用。下述表描述 UPDATE 请求和响应的示例性内容:

[0147]

```
<M:MyServiceUpdateRequest xmlns:M="http://www.example.org/myservice">  
  <M:Replace>http://www.example.com/first/endpoint</M:Replace>  
  <M:With>http://www.example.com/eight/endpoint</M:With>  
</M:MyServiceUpdateRequest>  
<W:DefaultUpdateResponse xmlns:W="http://abc/def/ghij/10/wsap.xsd"/>
```

[0148] 方便 WSAP 的另一操作包含复制,它提供保持在分布式服务上同步的状态的清楚、透明方式。为此,WSAP 提供允许服务与另一服务同步的复制模型。为使两个服务的状态保持同步,复制服务与在被复制服务上执行的复制服务上相同的一个或多个操作。

[0149] 通常,REPLICATE、SUBSCRIBE、CANCELREPLICATE 和 UNSUBSCRIBE 操作用户列表的状态。由于 WSAP 展现状态,用户列表可以保持在另一 WSAP 服务(例如 SubscribeHelper)中。在接收 SUBSCRIBE、REPLICATE、CANCELREPLICATE 和 UNSUBSCRIBE 请求之后,如果适当的话,主 WSAP 服务发送 Insert/Delete 请求。在主服务上的 LOOKUP 响应中,在捆绑中,识别允许第三方捕捉服务的整个状态的助手。如果用 URI 表示,包括用户列表的该整个状态,是可见的。注意通过分开有关它们自己的状态的预约,第三方可以通知服务的用户走开以及用不同 URI 再现。

[0150] 通常如图 14 所示,如果在复制服务 1430 上执行 INSERT 操作,基于来自用户 1450 的之前的 REPLICATE 请求,复制服务 1430 将在复制服务 1440 上执行 INSERT 操作。注意 WSAP 复制模型的语义不同 WSAP 通知模型(如下所述),它不要服务在接收 NOTIFY 消息上执行任何操作。

[0151] 能使用 REPLICATE 操作来复制服务或部分服务。复制的行为由包括在请求中的订约来描述。下述表阐述用于 REPLICATE 操作的属性:

[0152]

用于 REPLICATE 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/replicaterequest
响应动作	http://abc/def/ghij/10/wsap/replicaterequest
安全	否
等幂	是

[0153] 所阐述的下述表包含样本 REPLICATE 操作的 SOAP Body 内容。由 WSAP 定义请求,由服务定义响应。在该例子中,响应是为空响应而定义的方便类型:

[0154]

```
<W:ReplicateRequest xmlns:W="http://abc/def/ghij/10/wsap.xsd">  
  <W:Contact>http://www.example.com/contracts/fullreplication</W:Contract>  
</M:ReplicateRequest>  
<W:DefaultReplicateResponse xmlns:W="http://abc/def/ghij/10/wsap.xsd"/>
```

[0155] 通过发出 REPLICATE 操作启动复制,该操作包含描述拟使用的复制模型的订约。例如,复制服务可以通过 INSERT 操作,支持仅复制状态添加的复制模型。如果如在该例子中,在 INSERT 操作中,仅感兴趣复制服务,这可以通过引用 REPLICATE 操作中的复制订约来表示。如果接受 REPLICATE 请求,那么复制服务将其当前的状态作为针对复制服务的一组 INSERT 操作传送。

[0156] 注意如在图 14 中,预订服务 1450 可以不同于复制服务 1440。另外,复制服务可以已经存在以及具有某一状态。根据复制订约,当复制开始时,该状态可以或可以不被覆盖。在复制订约仅允许 INSERT 操作的上述例子中,因此,复制状态将添加到复制服务的现有状态。

[0157] 复制订约在某些情况下,可以通过本身终止,复制可以受截止日期限制,或它可以继续无限地复制该状态。在一种实现中,届满可以视为订约的一部分,然而,它可以另外被提供为值参数。

[0158] 使用 CANCELREPLICATION 操作,也可以明白地终止复制。用于 CANCELREPLICATION 操作的样本序列图在图 5 示出。如能看出,在复制服务 1530 处接收来自客户机 1502 的初始插入请求并复制到复制服务 1540。如果非用户 (unsubscribe) 服务 1552 请求 CANCELREPLICATION 操作,明确地终止所建立的复制。在图 15 的例子中,复制服务 1530 不将来自客户机 1502 的另外的插入请求复制到复制服务 1540。

[0159]

用于 CANCELREPLICATION 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/cancelreplicaterequest
响应动作	http://abc/def/ghij/10/wsap/cancelreplicaterequest
安全	否
等幂	否

[0160] 下述表包含样本 CANCELREPLICATION 操作的 SOAP Body 内容。响应使用为空响应而定义的方便类型。

[0161]

```
<W:CancelReplicateRequest xmlns:W="http://abc/def/ghij/10/wsap.xsd" />
<W:DefaultCancelReplicateResponse
xmlns:W="http://abc/def/ghij/10/wsap.xsd" />
```

[0162] 除复制模型外,WSAP 定义基于 NOTIFY 操作的简单的事件通知模型,它报告状态改变已经发生,但不揭示状态改变发生的环境。注意事件通知模型的语义不同于复制模型,其中,复制服务作为从复制服务发送的消息的结果,改变状态。在通知模型中,在接收 NOTIFY 消息后,不要求用户执行任何操作。因此,用户所执行的任何动作仅基于其自己的语义,而

不是事件通知源的语义。注意预订 NOTIFY 操作,因此可以是过期的,其可以是经值参数设置的订约的一部分。

[0163] NOTIFY 操作是 WSAP 通知模型的一部分,能用来发出事件通知,以及具有下述属性:

[0164]

用于 NOTIFY 的属性表	
属性	值
MEP	单向
请求动作	http://abc/def/ghij/10/wsap/notify
安全	是
等幂	是

[0165] 下述表包含样本 NOTIFY 操作的 SOAP Body 内容:

```
<M:MyNotification xmlns:M="http://www.example.org/myservice">  
  <M:CurrentEntries>6</M:CurrentEntries>  
</M:MyNotification>
```

[0166] 图 16 表示与 NOTIFY 有关的 SUBSCRIBE 操作,其中,用户 1650 向事件源 1670 发出请求。请求包括识别在查看状态变化中用户感兴趣的事件通知源的部分的查询。如果接受 SUBSCRIBE 操作,事件通知源 1670 启动表示源的当前状态的 NOTIFY 操作。在该初始化 NOTIFY 操作后,只要事件通知源的状态适当地改变事件通知源发出新 NOTIFY 操作。因为描述事件通知源的行为的订约与应用语义无关,其完全由 WSAP 定义。

[0167] 由此使用 SUBSCRIBE 操作来订约服务的状态的变化,以及具有下述属性:

[0168]

用于 SUBSCRIBE 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/subscribe request
响应动作	http://abc/def/ghij/10/wsap/subscribe request
安全	否
等幂	否

[0169] 下述表包含样本 SUBSCRIBE 操作的 SOAP Body 内容。由 WSAP 定义请求,以及由服

务定义响应。在该例子中，响应是为空响应而定义的方便类型：

[0170]

```
<M:MyServiceCurrentEntriesSubscribeRequest
xmlns:M=" http://www.example.org/myservice" />
<W:DefaultSubscribeResponse
xmlns:W=" http://magnet/contracts/2003/10/wsap.xsd" />
```

[0171] 通过截止时间可以限制预定，这与复制模型类似。类似地，使用 UNSUBSCRIBE 操作，可以在任何时间终止预约，如图 17 所示，其中非用户 1752 将 UNSUBSCRIBE 操作发送到事件源以便取消另外的通知。

[0172] 取消预约的 UNSUBSCRIBE 操作具有下述属性：

[0173]

用于 UNSUBSCRIBE 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/unsubscriberequest
响应动作	http://abc/def/ghij/10/wsap/unsubscriberequest
安全	否
等幂	否

[0174] 下述表包含样本 UNSUBSCRIBE 操作的 SOAP Body 内容。由 WSAP 定义请求，由服务定义响应，在该例子中，响应是为空响应而定义的方便类型。

[0175]

```
<W:UnsubscribeRequest xmlns:W=" http://magnet/contracts/2003/10/wsap.xsd" />
<W:DefaultUnsubscribeResponse
xmlns:W=" http://magnet/contracts/2003/10/wsap.xsd" />
```

[0176] 注意，WSAP 事件通知模型是传统的进栈模型，其中事件通知从源压入收汇处。然而，通过在消息路径中使用缓冲中介体，可以将进栈模型转换成出栈模型，其提供用于事件通知收信方的离线支持。

[0177] 转到服务终止的描述，当终止 WSAP 服务时，不再能发送或接收任何类型的消息。服务可以通过将服务的行为描述成服务能发送或接收的有限消息集来终止。当服务履行其订约时，它终止，并能恢复它所消耗的任何资源。

[0178] 另外，服务的行为可以定义迫使服务终止的显式交互作用。为此，WSAP 定义 DROP 操作，由此，可以显式地终止将 DROP 接受为其订约的一部分的服务。如图 18 所示，在从服务，诸如客户机 1802 接收 DROP 操作后，目标的服务 1820 简单地停止发送或接收任何未来的消息。与 CREATE 操作类似，DROP 能简单的，如图 18；或复杂的，如参考图 20 所示。

[0179] DROP 操作具有下述属性：

[0180]

用于 DROP 的属性表	
属性	值
MEP	请求-响应
请求动作	http://abc/def/ghij/10/wsap/droprequest
响应动作	http://abc/def/ghij/10/wsap/droprequest
安全	否
等幂	否

[0181] 下述表包含样本 DROP 操作的 SOAP Body 内容。由 WSAP 定义请求,响应使用为空响应而定义的方便类型：

[0182]

```
<W:DropRequest xmlns:M="http://magnet/contracts/2003/10/wsap.xsd" />
<W:DefaultDropResponse xmlns:W="http://magnet/contracts/2003/10/wsap.xsd" />
```

[0183] 注意,服务的行为允许发送或接收无限量的消息,以及不允许能终止它的显式操作是可行的,在任一事件中,不能以适度的方式终止服务。另外,注意在所述的 WSAP 操作数和已经过时钟时间的服务的寿命间,没有内在关系。建议(但不要求)在终止服务后,不再循环使用服务标识符,以避免过时引用被应用于相同名称的新服务的旧服务。

[0184] 在一些情况下,目标服务可以执行一些另外的清除,诸如未从目标服务登记或发送出最终事件通知。这些清除在结束 DROP 操作前执行,如由图 19 中的 DELETE 请求和 NOTIFY 操作所示。

[0185] 在复杂的服务终止中,如图 20 所示,根据设定值,终止多个相关服务。如果服务间的关系是使得它们不能单独地存在,会发生这种情况。在复杂的服务终止的情况下,WSAP 不提供固有地保证,在事务环境中终止所有服务。这种保证能通过增加具有事务支持的服务的行为来提供。

[0186] 从上述命令能看出,WSAP 允许服务重启和动态地负载平衡,因为它允许捕捉和重新创建不同站点的状态。这能用来实现事务、负载平衡和 / 或当前版本更新以保持可用时间最大。举例来说,考虑具有 URI A 类型 T 的服务。经 WSAP 命令,另一服务能从 A(例如经 GET 请求)获得数据,查看 A(例如经 LOOKUP 请求)以获得其依赖性,以及 Drop A。然后,通过创建具有 URI B 的类型 T 的服务,以及在 B 上执行替换(例如 DELETE 和 INSERT),结果是从丢弃 A 复制的完全相同的状态行为。

[0187] 如能从上述详细描述看出,提供一种方法和系统以及机制,通过这些客户机(服务)和其他网络服务能以有效和安全,而且可观察和重使用的方式与网络服务通信,包括组成服务的实体,以便中介体可以将值增加到通信上。由此,方法和系统提供当前计算所需

的显著的优点以及好处。

[0188] 尽管本发明能进行各种改进和替代的结构,某些所示的实施例在图中示出并且如上详细地描述。然而,应理解到不打算将本发明限制到所公开的特定的形式,而相反,本发明意图覆盖落在本发明的精神和范围内的所有改进、替代结构和等效。

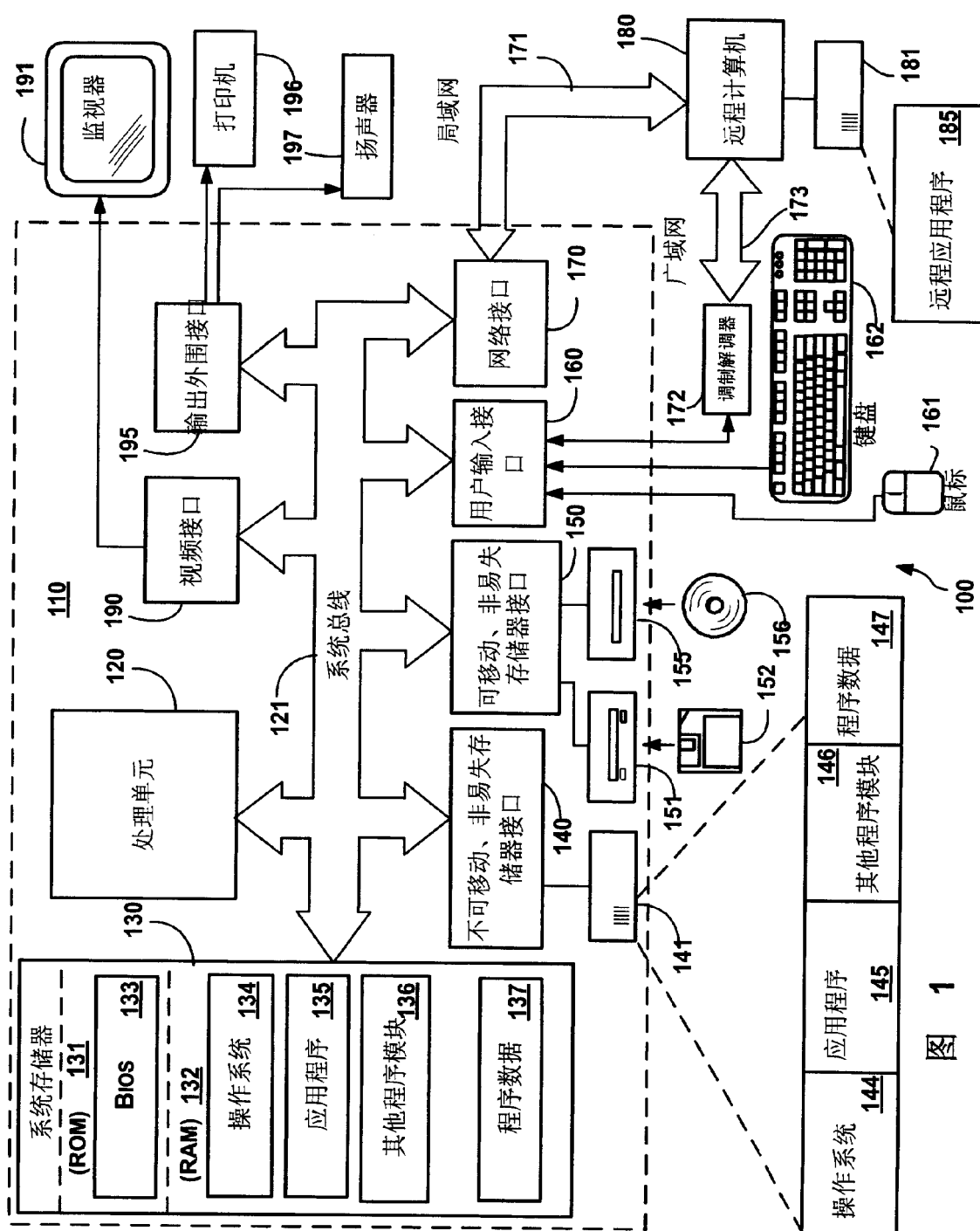


图 1

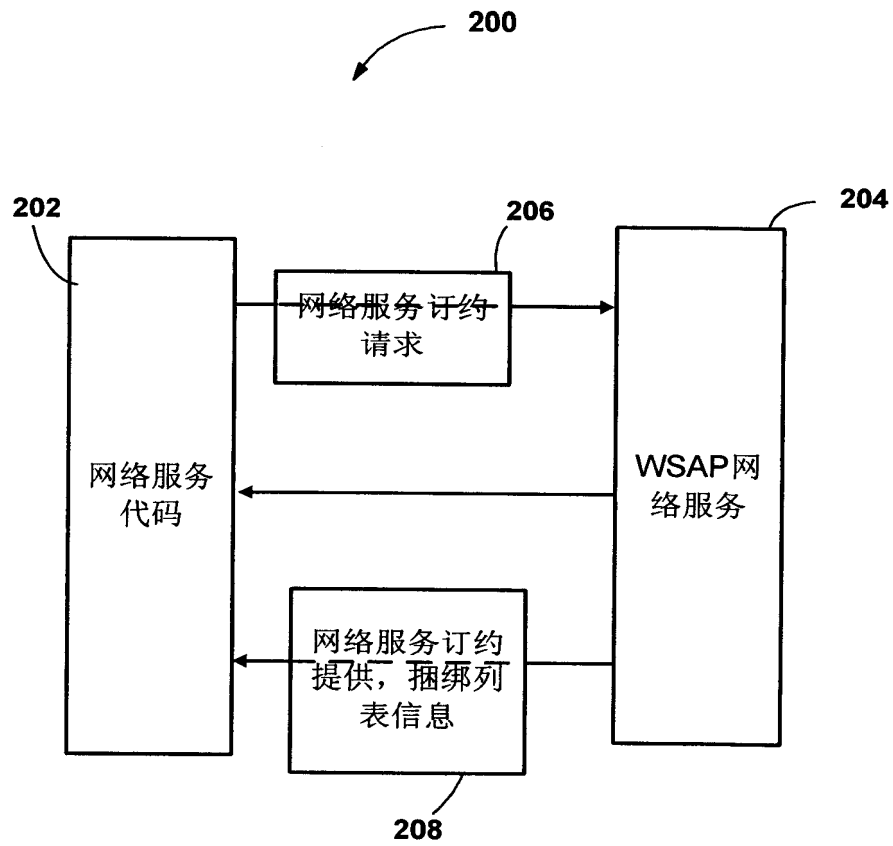


图 2

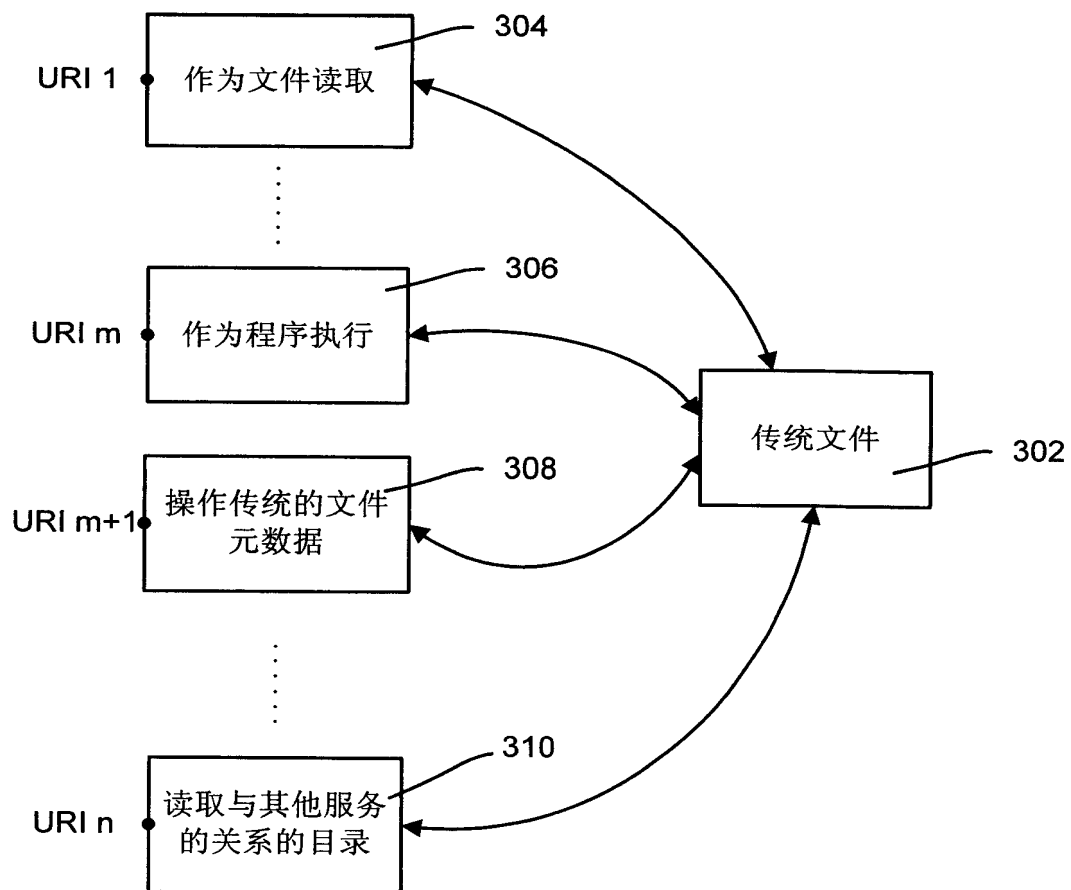


图 3

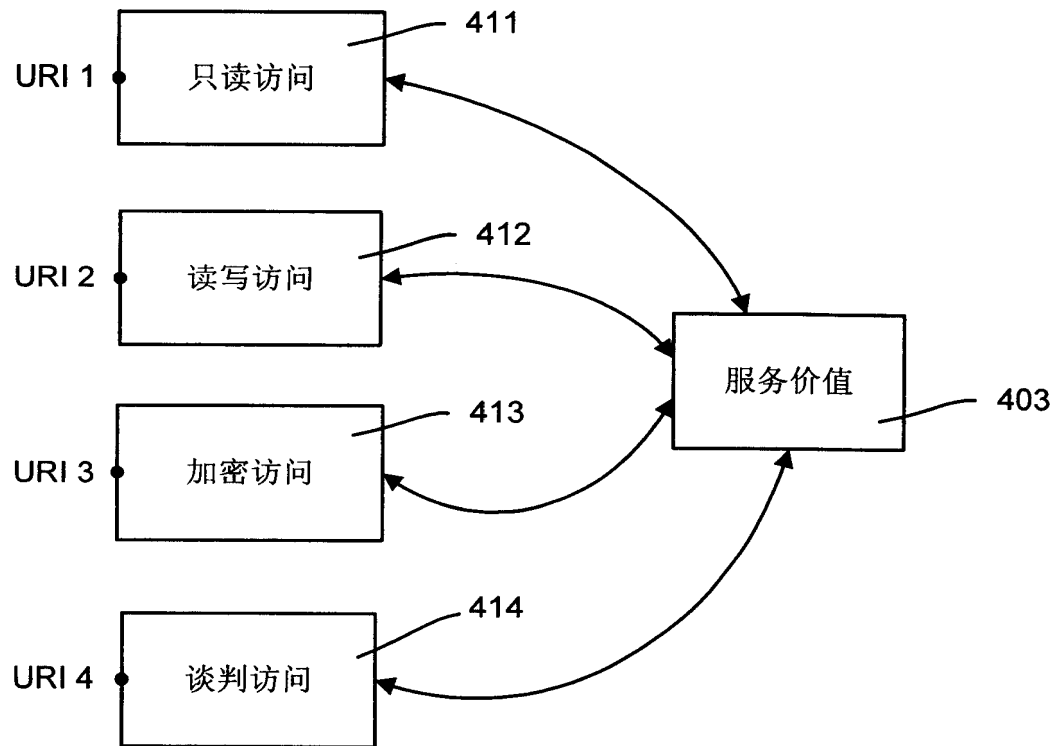


图 4

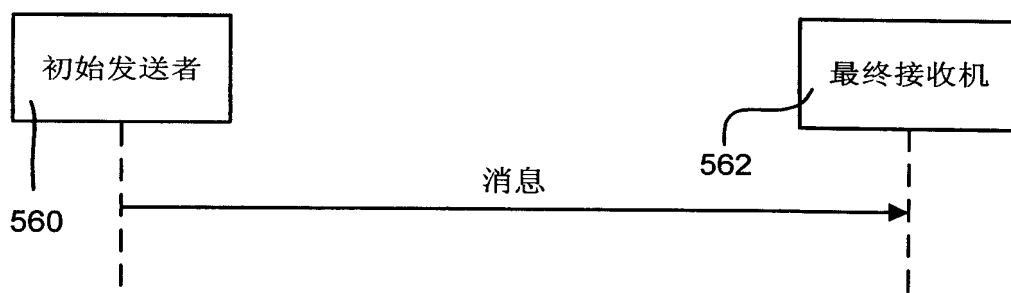


图 5

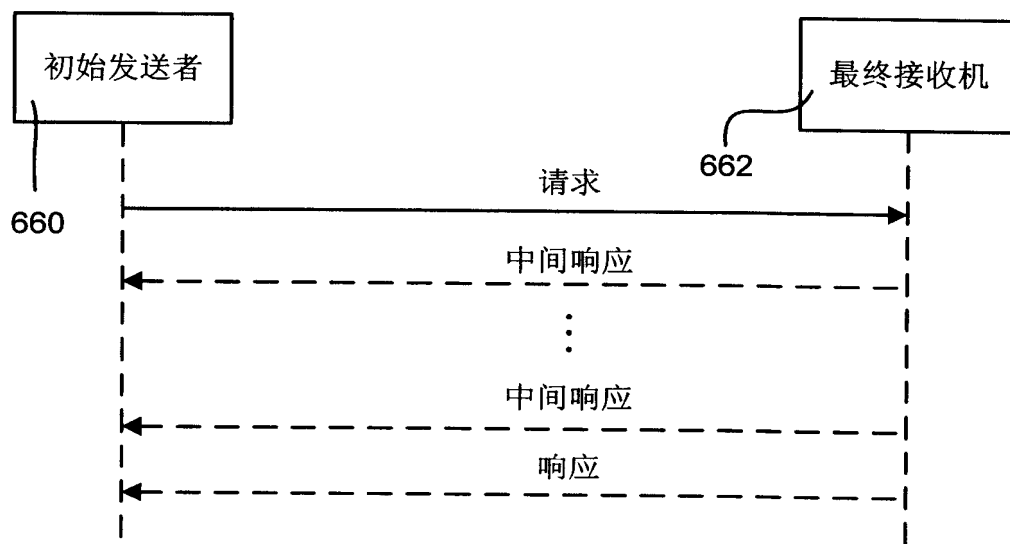
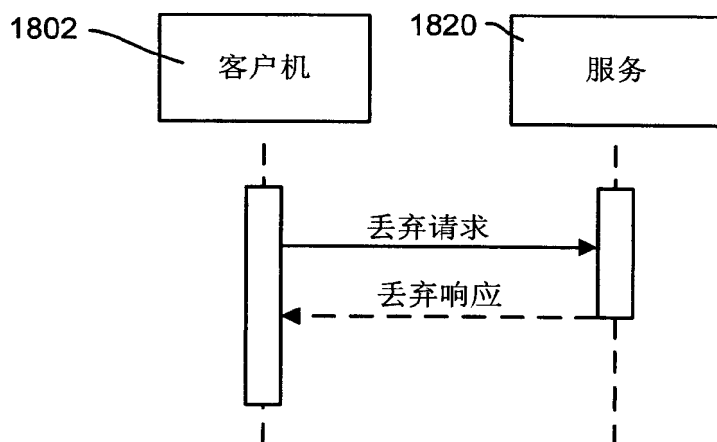
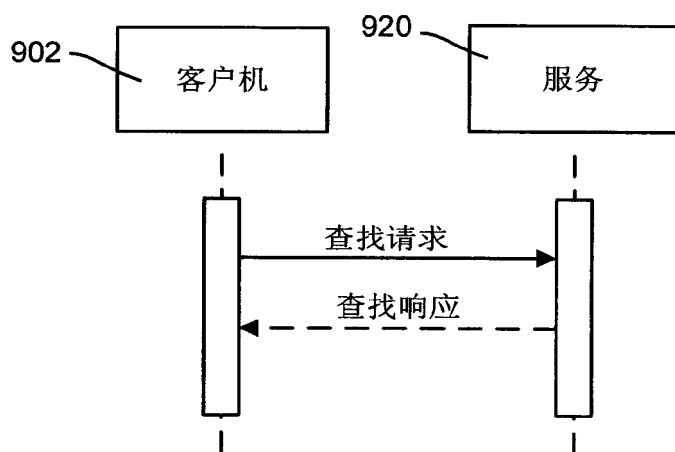
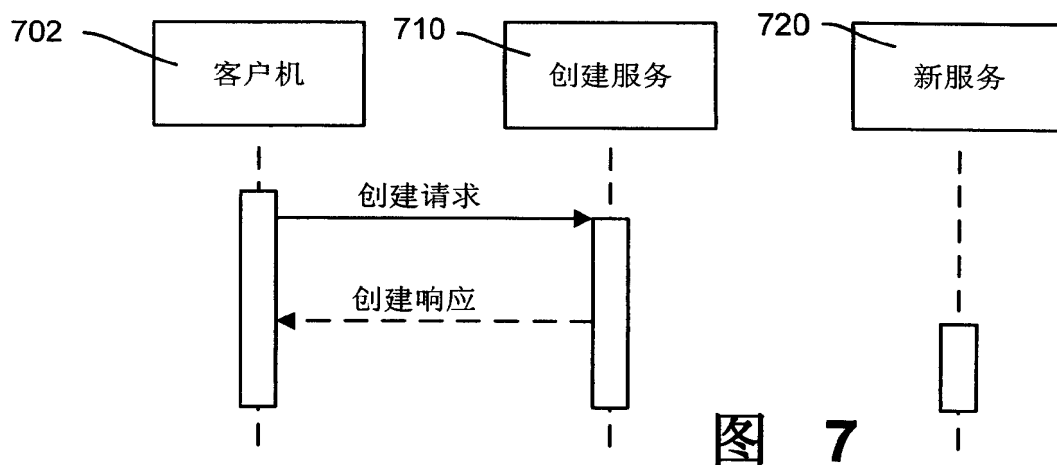


图 6



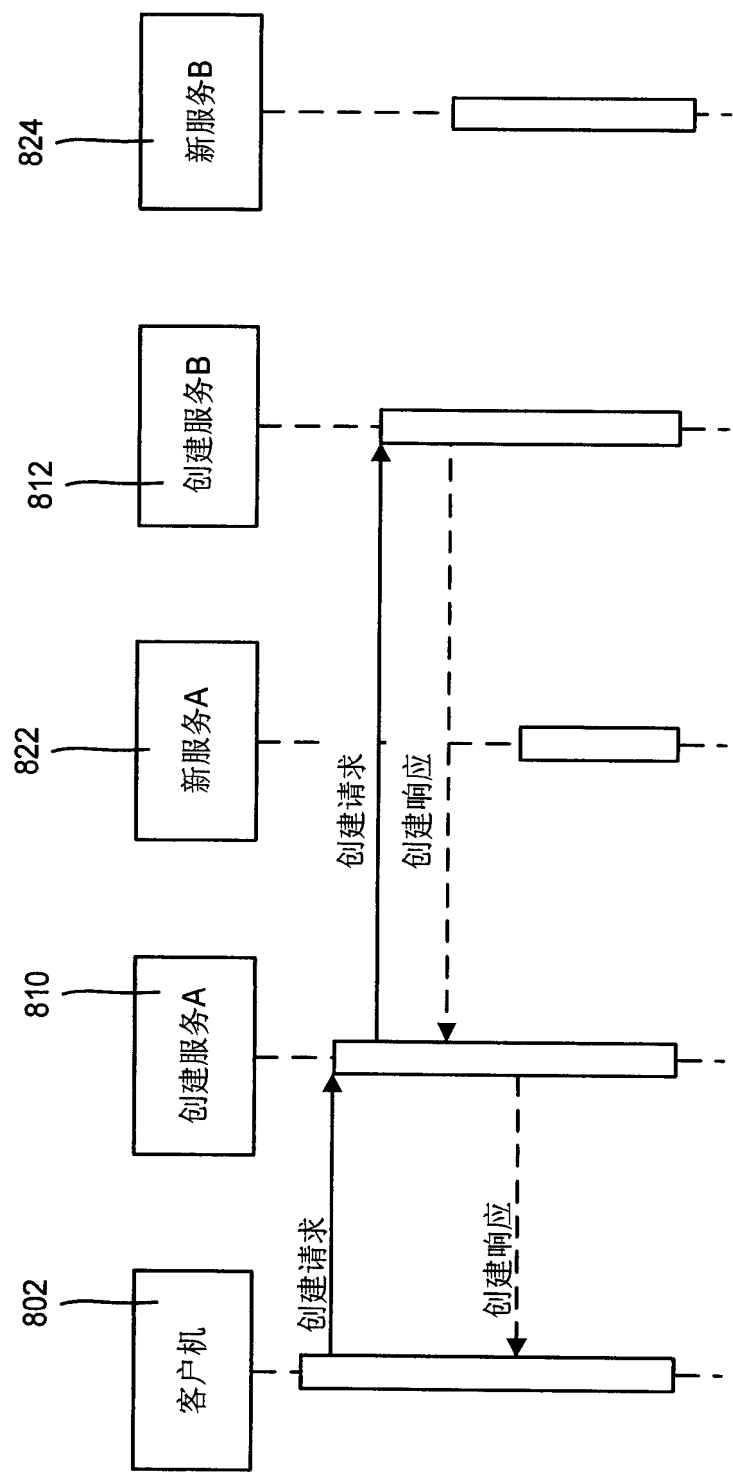


图 8

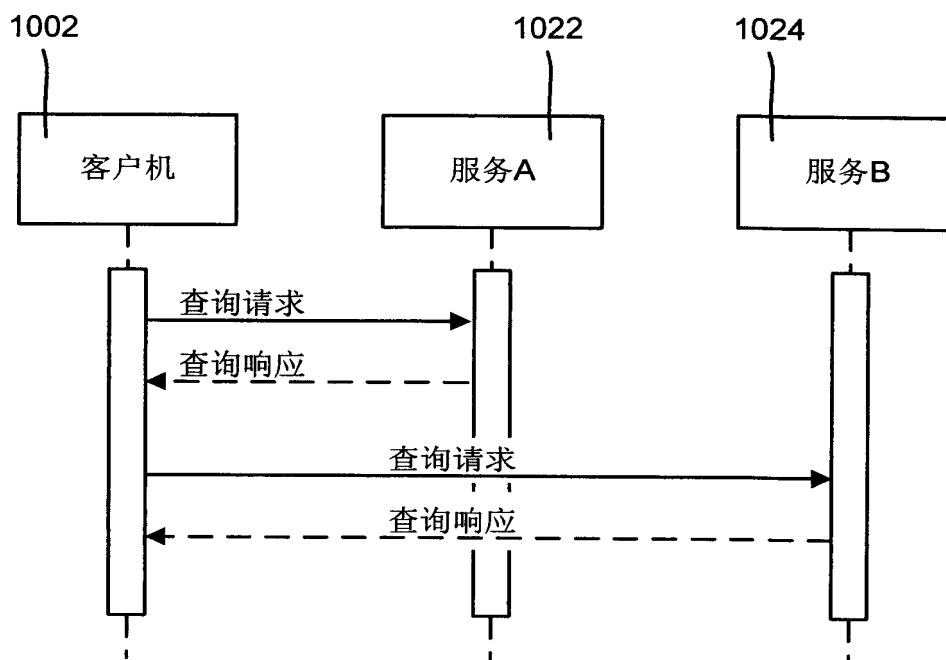


图 10

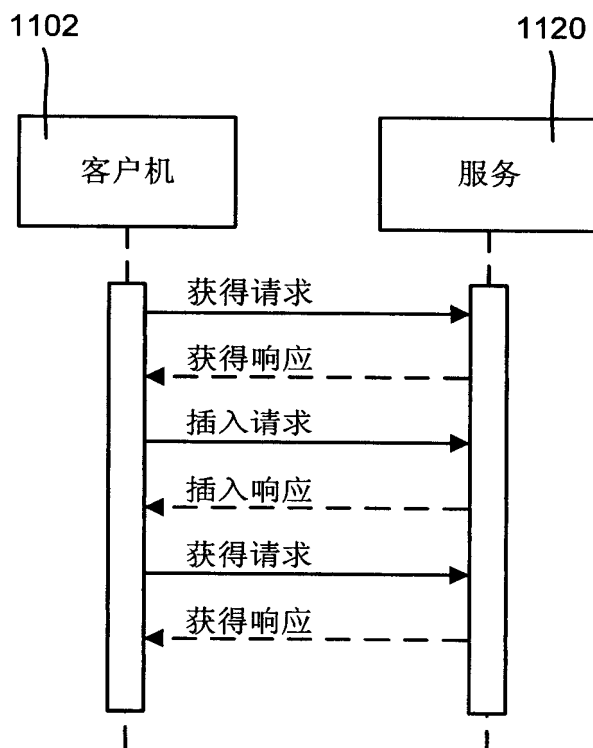


图 11

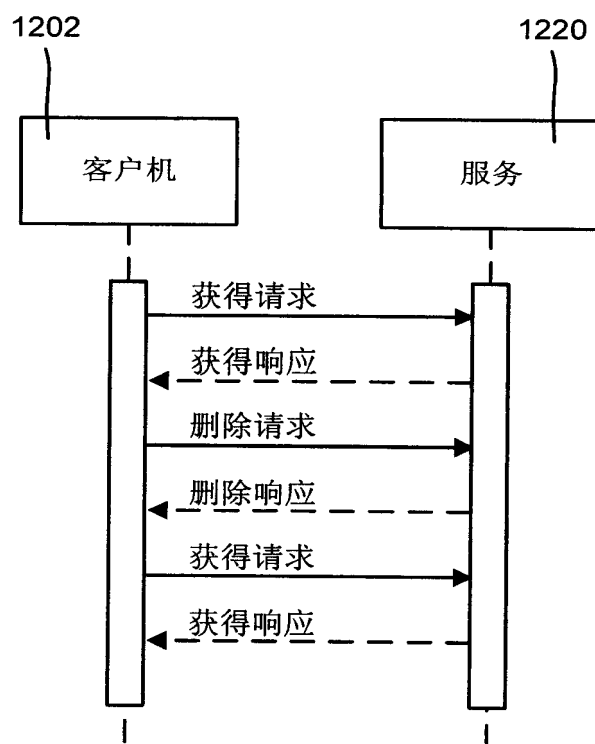


图 12

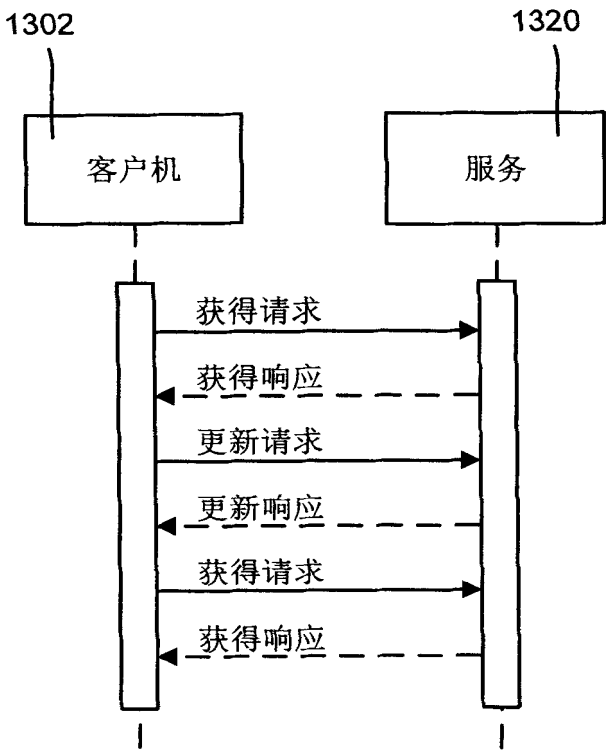


图 13

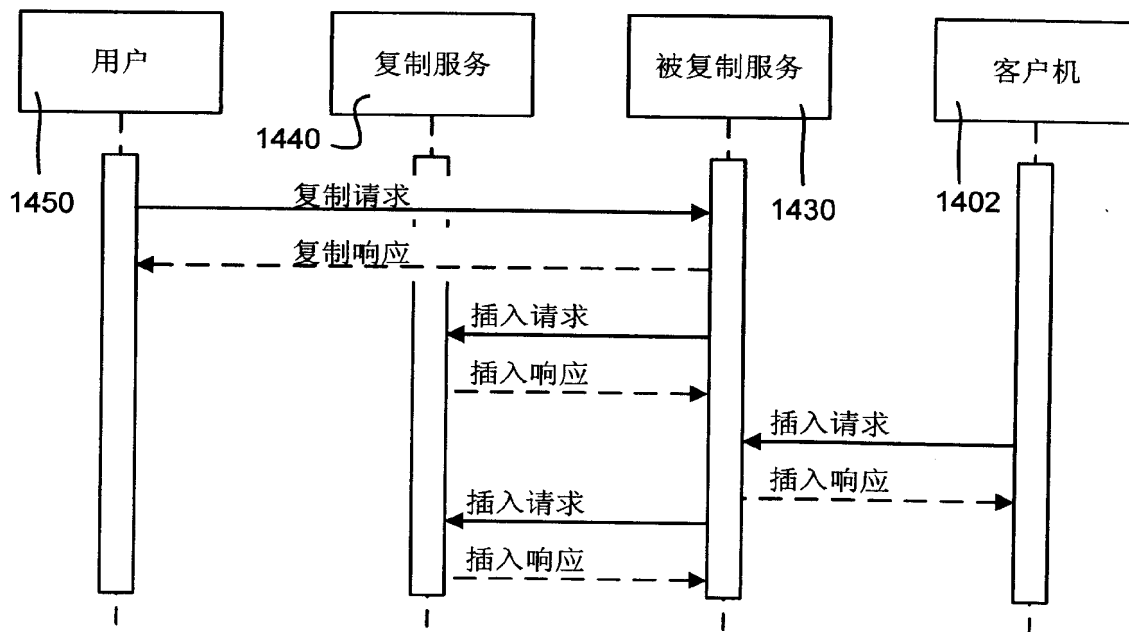


图 14

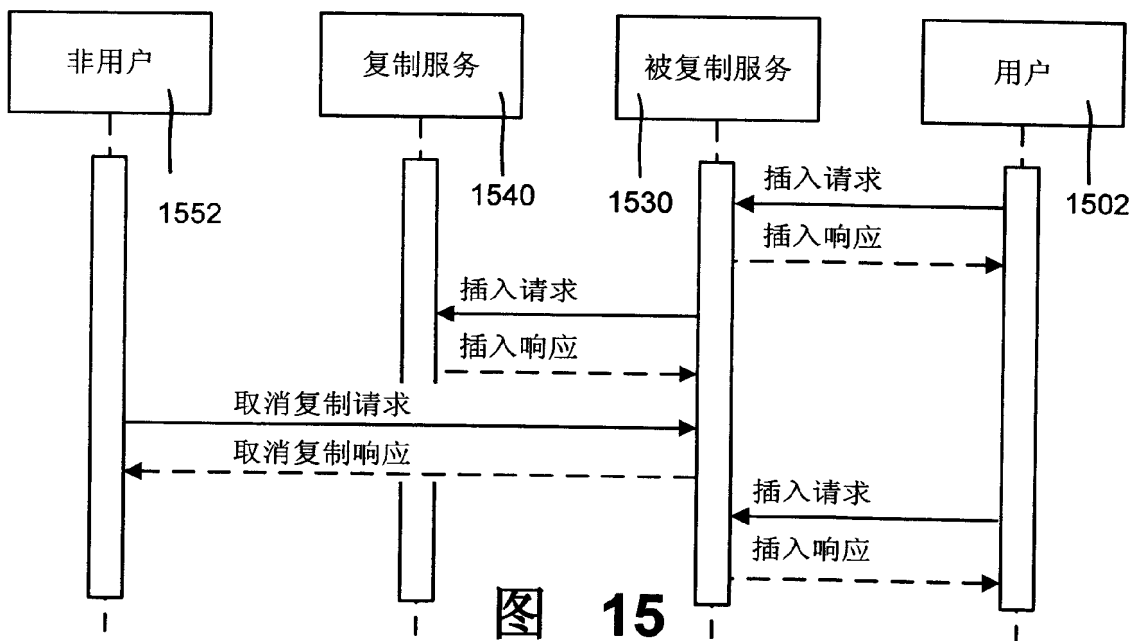


图 15

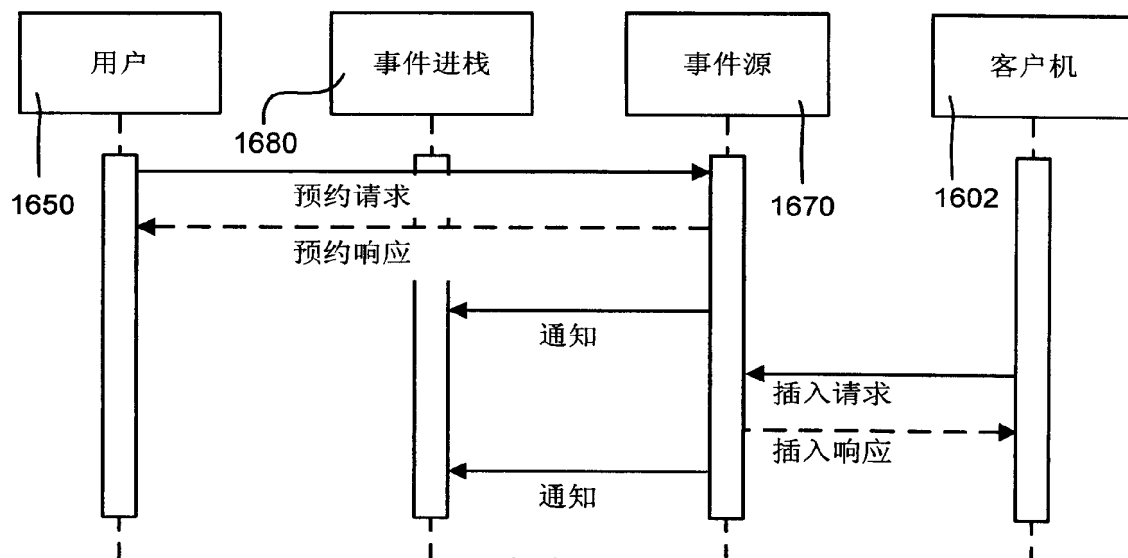


图 16

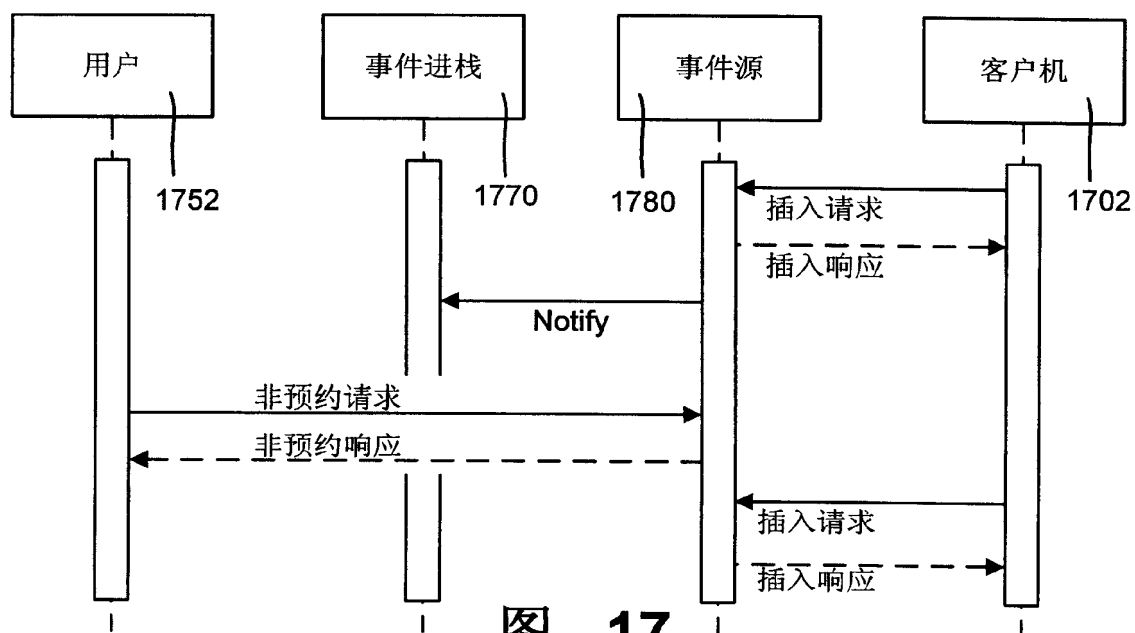


图 17

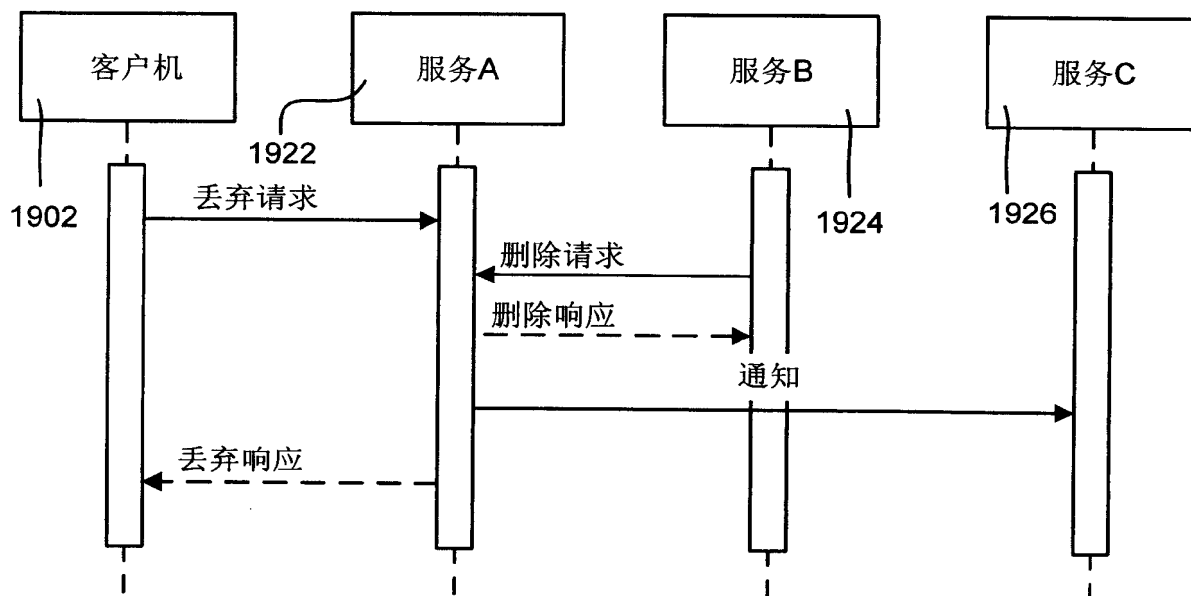


图 19

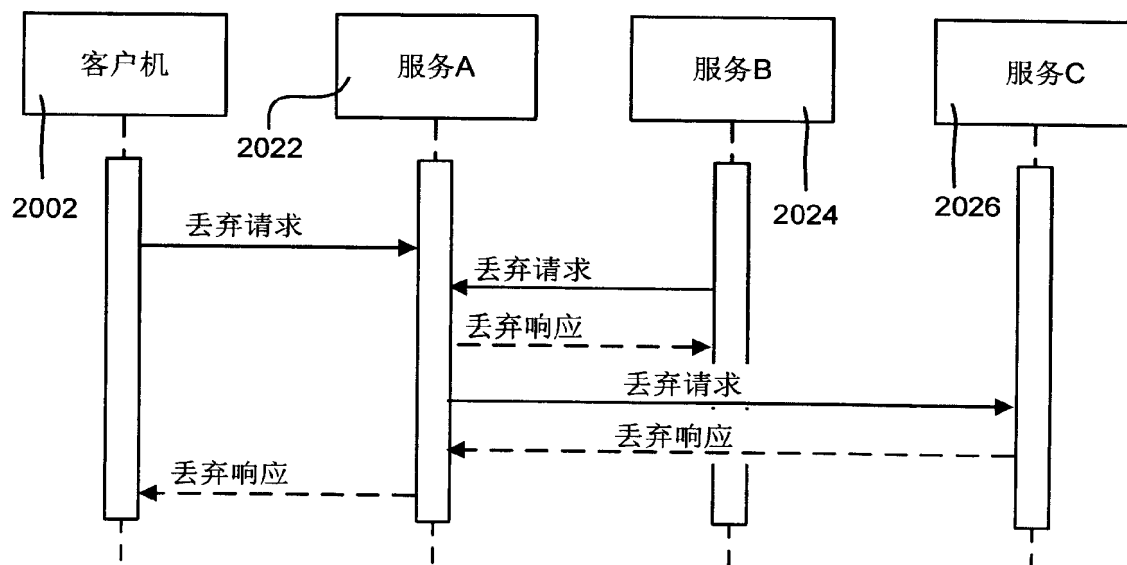


图 20