



(12) 发明专利

(10) 授权公告号 CN 102362261 B

(45) 授权公告日 2014. 01. 08

(21) 申请号 201080014045. 4

(22) 申请日 2010. 03. 19

(30) 优先权数据

12/410, 470 2009. 03. 25 US

(85) PCT国际申请进入国家阶段日

2011. 09. 23

(86) PCT国际申请的申请数据

PCT/US2010/028028 2010. 03. 19

(87) PCT国际申请的公布数据

W02010/111147 EN 2010. 09. 30

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 K·吉田

(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.

G06F 9/44 (2006. 01)

(56) 对比文件

US 6119166 A, 2000. 09. 12,

US 2007245353 A1, 2007. 10. 18,

JP 2007080063 A, 2007. 03. 29,

审查员 张俊伟

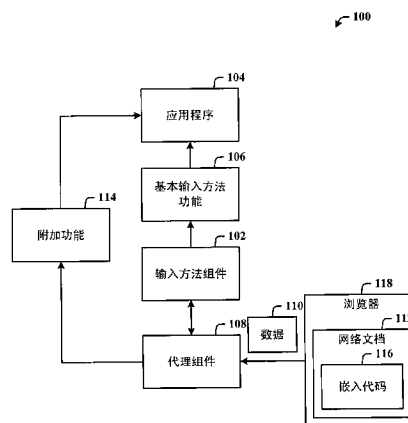
权利要求书2页 说明书9页 附图9页

(54) 发明名称

通过网络浏览器输入内容至应用程序

(57) 摘要

所公开的体系结构允许标记语言(例如HTML)页面便于将内容(例如文本)插入到输入方法所附属的应用程序中。采用这种方式,插件开发者可以使用网页语言(例如HTML)来实现任何UI外观和行为,正如规则的HTML页面一样与后端数据源通信,以及使用任意合适的技术来执行这一方式。由于网页被存储在网络服务器中并且在运行中被下载到客户端,因此安装是最小的并且用户可以自动地使用最新的版本。代理组件接收由程序设计语言编写并且嵌入标记语言的代码所产生的数据,并且提供另外的任意数据给应用程序。



1. 一种计算机实现的内容系统(100),包括:

附属于应用程序的输入方法组件(102),所述输入方法向应用程序提供基本输入方法功能;以及

配置成接收与网络文档相关联的数据的代理组件(108),所述数据添加附加功能给输入方法组件相关的应用程序,向所述代理组件注册 URL (统一资源定址器)前缀,所述 URL 前缀位于所述网络文档内,所述 URL 前缀是所述代理组件在其中运行的操作系统的本地 URL 前缀集的扩展,所述操作系统包括通过注册所述 URL 前缀来扩展所述本地 URL 前缀集的能力,注册所述 URL 前缀包括将所述 URL 前缀与操作系统的第二应用程序相关联。

2. 如权利要求 1 所述的系统,其特征在于,所述网络文档是包括在执行时生成数据的嵌入代码的网页,所述嵌入代码用程序设计语言编写并且被嵌入在标记语言中。

3. 如权利要求 1 所述的系统,其特征在于,所述网络文档被存储在网络服务器上,在网络服务器上更新,并且根据需要下载到客户端浏览器以供应用程序更新所用。

4. 如权利要求 1 所述的系统,其特征在于,所述网络文档便于经由网络文档控制将数据受控地输入应用程序中。

5. 如权利要求 1 所述的系统,其特征在于,所述代理组件可以使用 URL (统一资源定址器)协议处理器从网络文档中调用。

6. 如权利要求 1 所述的系统,其特征在于,所述输入方法组件发送通知中的标识符给代理组件,为代理组件指示数据将被路由到的相关联的输入方法组件或者应用程序。

7. 如权利要求 1 所述的系统,其特征在于,所述代理组件接收与相应的输入方法组件相关联的多个标识符,所述代理组件注册该标识符并且将数据路由至与最后注册的标识符的输入方法组件相关联的应用程序。

8. 如权利要求 1 所述的系统,其特征在于,所述附加输入方法功能包括文本或复杂对象中的至少一个。

9. 如权利要求 1 所述的系统,其特征在于,所述附加功能在浏览器运行调用代理组件的网页中的代码时从所述数据中生成。

10. 一种计算机实现的内容方法,包括:

向应用程序提供基本输入方法功能(500);

访问包括与附加应用程序功能相关的嵌入代码的网页(502);

运行网页中的嵌入代码以获取附加应用程序功能(504),所述网页包括 URL (统一资源定址器)前缀,所述 URL 前缀是所述应用程序在其中运行的操作系统的本地 URL 前缀集的扩展,向所述 URL 前缀注册代理,所述操作系统包括通过注册所述 URL 前缀来扩展所述本地 URL 前缀集的能力,注册所述 URL 前缀包括将所述 URL 前缀与操作系统的第二应用程序相关联;以及

将附加应用程序功能添加至应用程序(506)。

11. 如权利要求 10 所述的方法,其特征在于,还包括用脚本语言描述嵌入代码以及将代码嵌入网页中的标记语言内。

12. 如权利要求 10 所述的方法,其特征在于,还包括为后续执行更新网页的嵌入代码,以及基于更新的嵌入代码将更新的附加应用程序功能应用于应用程序。

13. 如权利要求 10 所述的方法,其特征在于,还包括向代理注册提供基本输入方法功

能的输入方法,并且基于该注册经由代理将附加应用程序功能路由至应用程序。

14. 如权利要求 10 所述的方法,其特征在于,还包括响应于与网页的交互,自动地添加附加应用程序功能至应用程序。

15. 如权利要求 10 所述的方法,其特征在于,还包括:

使用跟踪信息在多个应用程序中跟踪与应用程序的交互;

在通知中发送跟踪信息给代理;以及

基于跟踪信息,添加附加应用程序功能至对应的应用程序。

通过网络浏览器输入内容至应用程序

背景技术

[0001] 第三方程序要求输入方法具有插件功能以便输入方法能够将第三方组件添加到输入系统中以扩展在特定场景中的输入体验。这进一步促进了在运送完输入方法核心二进制之后用于添加更多的输入特征为服务的可扩展性。这一插件组件需要具有用于与后端数据源通信的用户界面 (UI) 外观和行为, 该后端数据源可以在本地计算机中或者位于远程服务器。此外, 这些插件组件需要被下载并且必须是可简单安装的, 并且在新版本可用时需要适当地被更新。

发明内容

[0002] 下面提供了简化的概述, 以便提供对此处所描述的一些新颖实施例的基本理解。本概述不是详尽的概述, 并且它不旨在标识关键 / 重要元素或描绘本发明的范围。其唯一目的是以简化形式呈现一些概念, 作为稍后呈现的更详细描述的前言。

[0003] 所公开的体系结构允许标记语言 (例如 HTML) 页面便于将内容 (例如文本) 插入到输入方法附属的应用程序中。采用这种方式, 插件开发者可以使用网页语言 (例如 HTML) 来实现任何用户界面 (UI) 外观和行为, 并且正如规则的网页使用合适的技术能做的, 与后端数据源通信。由于网页被存储在网络服务器中并且在运行中被下载到客户端浏览器, 因此安装是最小的并且用户可以自动地使用最新的版本。

[0004] 在一个实施例中, 实现了从嵌入在网页的 HTML 中的代码处接收指令的代理组件。客户端浏览器 (或者用于处理网页的其它合适的应用程序) 接收网页, 运行嵌入在 HTML 中的代码 (例如诸如 javascript 之类的脚本)。因此, 代理组件可以由 URL 协议处理器机制从 HTML 页面的代码中调用。此外, 代理组件可以被自身安装为客户端系统的一部分。

[0005] 例如, 考虑插件开发者想要实现将软件键盘特征添加到输入方法和 / 或相关应用程序的输入方法插件组件。开发者可以用代码编写 HTML 页面以示出范围从 'a' 到 'z' 的按键。当用户打开 HTML 页面并且选择页面上的按键时, 对应的字母被插入到客户端计算机中运行的应用程序中。由于这是 HTML 页面, 因此开发者可以用所需的装饰任意地布局按键。随后, 当选择添加更多的特征至插件 (例如添加更多插入单词而不是字符的按键) 时, 开发者简单地更新位于网络服务器处的 HTML 页面。例如, 用户通过客户端浏览器将自动地接收新版本 (更新)。

[0006] 为了为实现上述及相关目的, 本文结合下面的描述和附图来描述某些说明性方面。这些方面指示了可以实践本文所公开的原理的各种方式, 并且所有方面及其等效方面旨在落入所要求保护的主题的范围内。结合附图阅读下面的详细描述, 其他优点和新颖特征将变得显而易见。

[0007] 附图简述

[0008] 图 1 示出依照所公开的体系结构的计算机实现的内容系统。

[0009] 图 2 示出在便于经由网络文档的内容输入的组件和实体之间通信的流程图。

[0010] 图 3 示出用于添加附加的功能至应用程序的第一客户端配置。

- [0011] 图 4 示出用于添加附加的功能至应用程序的第二客户端配置。
- [0012] 图 5 示出经由浏览器输入内容到应用程序中的内容方法。
- [0013] 图 6 示出图 5 的方法的其它方面。
- [0014] 图 7 示出图 5 的方法的其它方面。
- [0015] 图 8 示出依照所公开的体系结构便于经由浏览器输入内容至应用程序的计算系统的框图。
- [0016] 图 9 示出便于通过浏览器输入内容至应用程序的计算环境的示意框图。

具体实施方式

[0017] 所公开的体系结构允许标记语言（例如 HTML）页面便于将内容（例如文本串、对象、音频、视频等）插入到输入方法附属的应用程序中。采用这种方式，插件开发者可以使用网页语言（例如 HTML）来实现任何 UI（用户界面）外观和行为，如规则的 HTML 页面所能的一样与后端数据源通信，以及使用合适的技术来实现这一方式。由于网页被存储在网络服务器中并且在运行中被下载到客户端，因此安装是最小的并且用户可以容易和自动地获取最新的版本。

[0018] 代理组件被实现在客户端上，并且接收由嵌入在网页的标记语言中的代码的执行所产生的数据。通过添加额外的代理组件至客户端系统，可以为对应的客户端应用程序在客户端系统上调节更多的数据类型。

[0019] 现在将参考附图，全部附图中相同的标号用于表示相同的元素。在下面的描述中，为了进行说明，阐述了很多具体细节以便提供对本发明的全面理解。然而，显而易见，可以在没有这些具体细节的情况下实施各新颖实施例。在其它情况下，以框图形式示出了公知的结构和设备以便于描述它们。本发明将涵盖落入所要求保护的的主题的精神和范围内的所有修改、等效方案和替换方案。

[0020] 图 1 示出依照所公开的体系结构的计算机实现的内容系统 100。系统 100 包括附属应用程序 104 的输入方法组件 102 以为应用程序 104 提供基本输入方法功能 106，以及用于接收导致添加由数据 110 提供的附加功能 114（例如输入方法功能）的网络文档 112 的数据 110 的代理组件 108，该附件功能 114 通过处理网络文档 112 中的嵌入代码 116 被间接地提供给应用程序 104。

[0021] 网络文档 112 可以是包括嵌入代码 116 的网页，该嵌入代码 116 在执行时生成数据 110。嵌入代码 116 可以用任意合适的程序设计语言（例如脚本语言）编写并且被嵌入在诸如 HTML 之类的标记语言中。网络文档 112 可以被存储在网络服务器上并且被下载到客户端浏览器 118，该客户端浏览器 118 处理或导致处理嵌入代码 116 以生成数据 110 以及因此的附加输入方法功能 114。

[0022] 一般而言，网络文档 112 基于用户与网络文档 112 的交互来便于自动补充应用程序 104 的附加功能 114（如数据 110）。附加功能 114 的产物（例如文本、图像等）然后被提供给应用程序 104。可以使用 URL（统一资源定址器）协议处理器机制从网络文档 112 中调用代理组件 108。例如，附加功能 114 包括文本以及诸如图像、音频和 / 或视频等对象中的至少一个。

[0023] 换句话说，计算机实现的内容系统 100 包括与应用程序 104 相关的用于向应用程

序 104 提供基本输入方法功能 106 的输入方法组件 102, 以及与输入方法组件 102 相关的用于从输入方法组件 102 处接收通知的代理组件 108, 该代理组件 108 以由网页 (例如网络文档 112) 中的代码 116 生成的数据的形式向应用程序 104 提供附加功能 114。

[0024] 代理组件 108 从处理网页中的代码的浏览器处接收数据 110。当浏览器 118 运行调用代理组件 108 的网页中的代码 116 时, 通过数据 110 生成附加功能 114。代理组件接收与输入方法组件 102 相关的标识符。代理组件 108 注册该标识符并且基于注册的标识符来确定数据 110 的路由。代理组件 108 便于利用特定数据类型的数据 110。此外, 代理组件 108 便于以字符、单词、短语或复杂对象中的至少一个的形式将数据 110 输入到应用程序 104 中, 其中应用程序 104 是语言应用程序。

[0025] 图 2 示出在便于内容通过网络文档输入的组件和实体之间通信的流程图 200。应用程序 104 可以是在客户端计算机上运行的程序, 并且输入方法组件 102 附属于应用程序 104。代理组件 108 可以是在客户端计算机上安装和运行的本地代码。流程图 200 示出, 在这一示例中, 网络文档是可以由插件开发者创建的 HTML 页面 202。例如, HTML 页面 202 包括用脚本语言编写的图 1 中的嵌入代码 116, 以便当浏览器运行代码 116 时, 产生的数据形成由输入方法组件 102 提供给应用程序 104 的附加输入方法功能。注意到可以使用其它技术来编写嵌入在 HTML 中的程序代码。

[0026] 插件开发者可以开发代理组件 108 并且使用户在本地客户端计算机上安装代理组件 108。当特定事件在输入方法组件 102 中发生时, 代理组件 108 从输入方法组件 102 处接收通知 204。此外, 代理组件可以由 URL 协议处理器 206 从 HTML 页面 202 中调用。

[0027] 数据流可以是单向的, 从网络浏览器中的网络文档到代理组件到输入方法组件并且最终到达应用程序。代理组件使用标识符 (也被称为票或令牌) 来标识哪一个输入方法组件实例以发送数据。代理组件预先知道输入方法组件的标识符。标识符可以从输入方法组件被推送至代理组件。

[0028] 在一个实现中, 无论何时应用程序获取用户交互的焦点 (或者应用程序窗口移动至前景), 应用程序的相关输入方法组件将其标识符 (也被称为事件通知) 推送至代理组件。附属于任何应用程序的每个输入方法组件可以在任何时候将其标识符推送至代理组件。代理组件记住最后注册的标识符。

[0029] 在任何时候 (当没有与从输入方法组件推送至代理组件的事件同步时), 可以从网络文档中调用代理组件。当被调用时, 代理组件使用最后注册的标识符以定位相关的输入方法组件实例。

[0030] 在网络浏览器中以任何可能的方式预先打开网络文档, 诸如响应于从喜爱的 URL 列表中的选择、输入方法组件用打开的特定网络文档等来开启网络浏览器。一旦被打开, 网络文档可以为所有输入方法组件实例服务。因此, 输入方法组件实例和网络文档并不必须是 1 比 1 的比率。例如, 桌面上呈现的软件键盘的单个实例可以用于与所需的任何应用程序或多个应用程序一起工作。

[0031] 在操作中, 当特定事件发生时, 输入方法组件 102 发送通知 204 (标识符) 至代理组件 108。代理组件 108 可以或者不可以用这些事件作为触发来调用 HTML 页面 202。事件包含“票”信息。票信息可以是应用程序 104、输入方法 102 的组件对象模型 (COM) 接口指针、或者代理组件 108 能够稍后使用的其它机制的抓手, 用于找到适当应用程序 (应用程序

104) 和 / 或输入方法。换句话说,客户端计算机可以使用多个不同的输入方法组件以及用于特定用途的可兼容代理组件。此处的实施例还描述了具有多个代理组件或者适合于每个客户端应用程序的不同代理的单个输入方法组件的场景

[0032] HTML 页面 202 可以在网络浏览器中打开并且由浏览器提供。开发者可以编写任何所需的 HTML 页面从具有适当参数的 HTML 页面中调用注册的 URL 协议处理器 206。作为响应,代理组件 108 被开启,并且从 HTML 页面 202 处接收参数。由于代理组件 108 知道连同通知 204 所发送的票信息,因此代理组件基于通知 206 中所接收的票数据来通信至应用程序 104 和输入方法组件 102 以添加由 HTML 页面 202 的数据产生的附加输入方法功能 208。

[0033] 一旦代理组件 108 被安装在客户端计算机上,开发者可以通过对存储在网络服务器中的 HTML 页面 202 中嵌入的代码做出改变或改进来改变或改进输入方法组件功能,也就是说不需要应用于客户端计算机的任何更新。这提供了用于自动地更新和部署附加功能的方便和有效的体系结构。

[0034] 注意到提供给应用程序 104 的内容不限于文本内容。以最简单的形式,用最简单的代理,内容可以是文本串。在更复杂的实现中,代理组件 108 可以被设计为当从 HTML 页面 202 中调用代理组件 108 时发送元数据到特定的应用程序,并且响应于接收到元数据,应用程序 104 可以插入复杂的对象(例如图像)至其本身。本质上,这基于在应用程序 104 和代理组件 108 之间的“合约”。如上文所述,多个不同的代理组件可以用于特定的用途。因此,众多不同的数据类型可以被添加至用户系统。

[0035] HTML 和网络浏览技术的使用已经存在于服务器端。当在网络服务器中的 HTML 中使用脚本并且用户浏览网页时,HTML 页面从服务器处被下载到客户端浏览器。客户端浏览器运行嵌入的脚本代码以生成数据。因此,用户不需要安装客户端,而仅需要简单地打开服务器中的 HTML 页面。代理组件可以在客户端上被安装为本地代码。每当客户端从服务器处下载对应的更新网页时,网页文档中以前下载至客户端浏览器的程序代码可以被自动地更新。

[0036] 代理组件定义数据类型。一旦数据类型被定义并且相关的代理被安装,代理一般将不需要频繁地更新。但是,数据类型的数据可以改变,并且该数据类型的数据生成器(网页中编码的)可以动态地改变。因此,由于通过例如服务器端存储的标记语言(例如 HTML)和脚本语言(例如 java script)的方式描述了数据的生成器,该生成器可以被容易地改变。如上文所述,标记语言不限于 HTML,而是可以是浏览器中可提供的合适技术。

[0037] 可以被采用的操作系统的一个特征是 URL 协议处理器。这一技术可以将 URL 映射到可执行文件(“exe”)。例如,URL 可以有以“http:”、“mailto:”或“ftp:”等开始的格式串。操作系统可以包括传递这一串、检查串的前缀、以及传递控制至注册的可执行应用程序的机制。例如,如果 URL 以“http:”开始,则操作系统将开启可执行的浏览器。在另一个实例中,如果 URL 以“mailto:”开始,则操作系统开启消息程序(例如电子邮件)。

[0038] 操作系统还可以包括允许代理组件扩展这一能力的特征,以便可以结合任何可执行文件注册和定义任何前缀串。这一机制用于注册所需的前缀串到代理组件。

[0039] 在 HTML 页面中,当用户开启具有前缀的这一 URL 时,操作系统开启注册的可执行代理。当代理组件被调用时,使用命令行参数,代理组件接收以前缀开始的完整 URL 串。例如,如果 HTML 页面包括一些“<前缀>:xyz”,其中<前缀>可以是“http”、“ftp”和“mailto”

等,则代理组件接收命令行参数中的“<前缀>:xyz”的串。如果每个 HTML 页面想要传递串“xyz”至代理组件,则 HTML 页面包括“<前缀>:xyz”。然后代理组件被开启并且接收串“<前缀>:xyz”,以便代理组件能够通过删除已知的“<前缀>:”部分来提取串“xyz”。

[0040] 当代理组件接收串“xyz”时,代理组件和应用程序使用发送和接收串的协议。能够被使用的非常基本的协议是从一个窗口发送字符到另一个窗口的 WM_CHAR(用于 Windows™ 操作系统)。如果代理组件发送 WM_CHAR 消息至应用程序(例如具有参数 x 的 WM_CHAR,紧接着为具有参数 y 的 WM_CHAR 以及具有参数 z 的 WM_CHAR),则应用程序接收“x”紧接着为“y”紧接着为“z”的键盘输入。因此,所有应用程序在接收后可以处理这一消息,并且输入“xyz”至应用程序正在处理的文档。那就是基本的代理模型。因此,如果需要,可以用不同的协议实现和注册众多不同的代理组件。

[0041] 在更高级的代理模型中,照片、视频、图像、声音等可以被推送至应用程序。在这一例子中,用户应用程序和代理可以使用用于最优操作的预定义的私人协议。在更具体的实现中,可以为具体的客户端应用程序开发定制代理。

[0042] 在客户端机器上可以有众多应用程序,并且每个应用程序与附属的输入方法组件操作,但是代理组件是单个实例。输入方法组件用信号通知代理组件代理组件将发送消息至哪一个应用程序。当用户在众多可能的应用程序中按下按键时,输入方法组件记住用户正在交互的应用程序,并且将该应用程序的操作系统处理发送给代理组件。之后,代理组件使用该处理以发送消息。这仅是一个示例,因为其它方法可以用于标识消息被发送至的应用程序。

[0043] 所公开的体系结构发现对于例如使用合适的键盘上通常不呈现的字符的语言(例如日语、中文、阿拉伯语、拉丁语等)的特别适用性。在另一个实现中,所公开的体系结构可以用于通过由输入设备(例如鼠标、键盘等)帮助的单个选择来插入单词和/或短语,这通过键盘是不能达到的。

[0044] 图 3 示出用于添加附加的功能至应用程序的第一客户端配置 300。例如,在第一客户端配置 302 中,第一输入方法组件 304 正在被第一应用程序 306 使用以利用由第一输入方法组件 304 所提供的基本输入方法功能。在任何时间点,第一输入方法组件 304 通过对第一代理组件 308 的通知来注册标识符到第一代理组件 308。当启动时,第一网页 310 调用第一代理组件 308。第一网页 310 包括用程序设计语言(例如 javascript)编写的以及嵌入在诸如 HTML 之类的标记语言中的嵌入代码 312。替代地,可以使用其它合适的网络文档语言。

[0045] 客户端浏览器从存储用于多种不同用途的网页的网络服务器(未示出)处下载第一网页 310。第一网页 310 在客户端浏览器中打开,并且客户端浏览器运行嵌入代码 312。代码 312 的执行输出通过协议处理器传递给第一代理组件 308 的数据。第一代理组件 308 然后将数据路由至第一应用程序 306 以添加用于在第一应用程序 306 中插入和使用的第一附加输入方法功能 310。

[0046] 图 4 示出用于添加附加的功能至应用程序的第二客户端配置 400。这里,单个代理组件 402 支持用于客户端机器上的两个应用程序和两个对应的输入方法组件的数据类型。第二输入方法组件 404 与第二应用程序 406 相关联(例如附属于),并且第三输入方法组件 408 与第三应用程序 410 相关联(例如附属于)。第二输入方法组件 404 和第三输入方法

组件 408 向代理组件 402 注册（发送通知），最后的注册指示活跃的应用程序（例如在用户交互的前景中）。

[0047] 示出两个网页 412：第二网页 414 和第三网页 416。当第二网页 414 由客户端浏览器下载并且处理时，浏览器运行导致调用代理组件 402 的嵌入代码 418。客户端浏览器的代码 418 的执行生成传递给代理组件 402 的数据，并且代理组件 402 基于最后注册的输入方法组件（例如第二输入方法组件 404）将数据路由为附加应用程序功能。例如，从第二网页 414 传递来的数据可以作为附加应用程序功能 422 被发送至第二应用程序 406。

[0048] 类似地，当第三网页 416 由客户端浏览器下载并且处理时，浏览器运行导致调用代理组件 402 的嵌入代码 420。客户端浏览器的代码 420 的执行生成传递给代理组件 402 的数据，并且代理组件 402 基于最后注册的输入方法组件（例如第三输入方法组件 408）将数据路由为附加应用程序功能。例如，从第三网页 416 传递来的数据可以作为附加应用程序功能 424 被发送至第三应用程序 410。

[0049] 在主体体系结构中可构想，代理组件可以被开发为提供对单个输入方法组件（例如第二输入方法组件 404）的支持，以及随后对两个客户端应用程序的支持。应用程序（406 和 410）以及相关的输入方法组件（404 和 408）是基于客户端的应用程序以及代理组件 402。网页（414 和 416）是基于服务器的以用于方便和有效的更新部署。

[0050] 此处所包括的是一组代表用于执行所公开的体系结构的新颖方面的示例性方法的流程图。尽管出于解释简明的目的，此处例如以流图或流程图形式示出的一个或多个方法被示出并描述为一系列动作，但是可以理解 and 明白，各方法不受动作的次序的限制，因为根据本发明，某些动作可以按与此处所示并描述的不同的次序和 / 或与其它动作同时发生。例如，本领域技术人员将会明白并理解，方法可被替换地表示为一系列相互关联的状态或事件，诸如以状态图的形式。此外，并非在方法中示出的所有动作都是新颖实现所必需的。

[0051] 图 5 示出通过浏览器输入内容到应用程序中的内容方法。在 500 处，基本输入方法功能被提供给应用程序。在 502 处，访问包括有关附加应用程序功能的嵌入代码的网页。在 504 处，运行网页中的嵌入代码以获取附加应用程序功能。在 506 处，添加附加应用程序功能到应用程序。

[0052] 图 6 示出图 5 的方法的其它方面。在 600 处，嵌入代码用脚本语言来描述并且嵌入在网页中的标记语言内。在 602 处，为后续执行更新网页的嵌入代码，并且基于更新的嵌入代码将更新的附加应用程序功能应用于应用程序。在 604 处，向代理注册提供基本输入方法功能的输入方法，并且基于该注册经由代理将附加应用程序功能路由至应用程序。

[0053] 图 7 示出图 5 的方法的其它方面。在 700 处，响应于与网页的交互，附加应用程序功能被自动地添加至应用程序。在 702 处，使用跟踪信息在多个应用程序中跟踪与应用程序的交互。在 704 处，用通知发送跟踪信息给代理组件。在 706 处，基于跟踪信息，附加应用程序功能被添加至对应的应用程序。

[0054] 如在本申请中所使用的，术语“组件”和“系统”旨在表示计算机相关的实体，其可以是硬件、硬件和软件的组合、软件、或者执行中的软件。例如，组件可以是，但不限于，在处理器上运行的进程、处理器、硬盘驱动器、多个存储驱动器（光学的、固态和 / 或磁存储介质）、对象、可执行件、运行的线程、程序，和 / 或计算机。作为说明，在服务器上运行的应用

程序和该服务器两者都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程内,并且组件可以位于一个计算机上和 / 或分布在两个或更多的计算机之间。词语“示例性”此处可用于表示用作示例、实例或说明。在此被描述为“示例性”的任何方面或设计并不一定要被解释为相比其它方面或设计更优选或有利。

[0055] 现在参考图 8,示出依照所公开的体系结构便于通过浏览器输入内容至应用程序的计算系统 700 的框图。为了提供用于其各方面的附加上下文,图 8 及以下讨论旨在提供对其中可实现各方面的合适的计算系统 800 的简要概括描述。尽管以上描述是在可在一个或多个计算机上运行的计算机可执行指令的一般上下文中进行的,但是本领域的技术人员将认识到,新颖实施例也可结合其它程序模块和 / 或作为硬件和软件的组合来实现。

[0056] 用于实现各方面的计算系统 800 包括计算机 802,其具有处理单元 804、系统存储器 806、以及系统总线 808。处理单元 804 可以是各种市场上可买到的处理器中的任一种,诸如单处理器、多处理器、单核单元以及多核单元等。此外,本领域的技术人员将明白,各新颖方法可用其它计算机系统配置来实施,包括小型计算机、大型计算机、以及个人计算机(例如、台式、膝上型等)、手持式计算设备、基于微处理器的或可编程消费电子产品等,其每一个都可在操作上耦合到一个或多个相关联的设备。

[0057] 系统存储器 806 可以包括易失性 (VOL) 存储器 810(例如,随机存取存储器 (RAM)) 和非易失性存储器 (NON-VOL) 812(例如 ROM、EPROM、EEPROM 等)。基本输入 / 输出系统 (BIOS) 可以被存储在非易失性存储器 812 中,并且包括诸如在启动期间便于在计算机 802 内的组件之间传递数据和信号的基本例程。易失性存储器 810 还可以包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0058] 系统总线 808 提供了用于包括,但不限于存储器子系统 806 的系统组件到处理单元 804 的接口。系统总线 808 可以是若干种总线结构中的任一种,这些总线结构还可使用各类可购买到的总线体系结构中的任一种互连到存储器总线(带有或没有存储器控制器)以及外围总线。

[0059] 计算机 802 还包括存储子系统 814 以及用于将存储子系统 814 接口到系统总线 808 和其它所需计算机组件的存储接口 816。存储子系统 814 可以包括例如硬盘驱动器 (HDD)、磁软盘驱动器 (FDD) 和 / 或光盘存储驱动器(例如,CD-ROM 驱动器、DVD 驱动器)中的一种或多种。存储接口 816 可以包括诸如,例如 EIDE、ATA、SATA 和 IEEE 1394 等接口技术。

[0060] 一个或多个程序和数据可以被存储在存储器子系统 806、可移动存储器子系统 818(例如,闪存驱动器形状因子技术)和 / 或存储子系统 814(例如,光、磁、固态)中,包括操作系统 820、一个或多个应用程序 822、其它程序模块 824 以及程序数据 826。

[0061] 例如,一个或多个应用程序 822、其它程序模块 824 以及程序数据 826 可以包括图 1 中的输入方法组件 102、应用程序 104、底线输入方法功能 106、代理组件 108、附加功能 114、数据 110、浏览器 118、网络文档 112 以及嵌入代码 116,图 2 中的组件和实体,图 4 中的组件和实体,以及图 5-7 的方法中描述的一个或多个动作。

[0062] 一般而言,程序模块包括执行特定任务或实现特定抽象数据类型的例程、方法、数据结构、其它软件组件等。操作系统 820、应用程序 822、模块 824 和 / 或数据 826 的全部或部分也可以被高速缓存在诸如易失性存储器 810 等存储器中。应当明白,所公开的体系结

构可以用各种市场上可购得的操作系统或操作系统的组合（例如，作为虚拟机）来实现。

[0063] 存储子系统 814 和存储子系统 (806 和 818) 用于数据、数据结构、计算机可执行指令等的易失性和非易失性存储的计算机可读介质。计算机可读介质可以是可由计算机 802 访问的任何可用介质，且包括易失性和非易失性介质、可移动和不可移动介质。对于计算机 802，介质容纳适当的数字格式的数据的存储。本领域的技术人员应当明白，可以使用其它类型的计算机可读介质，诸如 zip 驱动器、磁带、闪存卡、磁带盒等，来存储用于执行所公开的体系结构的新颖方法的计算机可执行指令。

[0064] 用户可以使用诸如键盘和鼠标等外部用户输入设备 828 来与计算机 802、程序和数据交互。其它外部用户输入设备 828 可以包括话筒、IR（红外）遥控器、操纵杆、游戏手柄、照相机识别系统、指示笔、触摸屏、姿势系统（例如，眼移动、头移动等）和 / 或类似物。在计算机 802 是例如便携式计算机的情况下，用户可以使用诸如触摸垫、话筒、键盘等板载用户输入设备 830 来与计算机 802、程序和数据交互。这些和其它输入设备通过输入 / 输出 (I/O) 设备接口 832 经由系统总线 808 连接到处理单元 804，但也可通过其它接口连接，诸如并行端口、IEEE 1394 串行端口、游戏端口、USB 端口、IR 接口等。I/O 设备接口 832 还便于使用输出外围设备 834，诸如打印机、音频设备、照相机设备等，诸如声卡和 / 或板载音频处理能力。

[0065] 一个或多个图形接口 836（通常也称为图形处理单元 (GPU)）提供计算机 802 和外部显示器 838（例如，LCD、等离子）和 / 或板载显示器 840（例如，对于便携式计算机）之间的图形和视频信号。图形接口 836 也可作为计算机系统板的一部分来制造。

[0066] 计算机 802 可以使用经由有线 / 无线通信子系统 842 到一个或多个网络 and / 或其它计算机的逻辑连接在联网环境（例如，IP）中操作。其它计算机可以包括工作站、服务器、路由器、个人计算机、基于微处理器的娱乐设备、对等设备或其它常见的网络节点，并且通常包括以上相对于计算机 802 描述的许多或所有元素。逻辑连接可以包括到局域网 (LAN)、广域网 (WAN)、热点等的有线 / 无线连接。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，所有这些都连接到诸如因特网等全球通信网络。

[0067] 当在联网环境中使用时，计算机 802 经由有线 / 无线通信子系统 842（例如，网络接口适配器、板载收发机子系统等）连接到网络来与有线 / 无线网络、有线 / 无线打印机、有线 / 无线输入设备 844 等通信。计算机 802 可以包括用于通过网络建立通信的调制解调器或其它装置。在联网环境中，相对于计算机 802 的程序和数据可以被存储在远程存储器 / 存储设备中，如与分布式系统相关联。应当明白，所示网络连接是说明性的，并且可以使用在计算机之间建立通信链路的其它手段。

[0068] 计算机 802 可以用于使用诸如 IEEE 802. xx 标准家族等无线电技术来与有线 / 无线设备或实体通信，例如在操作上安置在与例如打印机、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）以及电话的无线通信（例如，IEEE 802. 11 空中调制技术）中的无线设备。这至少包括用于热点的 Wi-Fi（即无线保真）、WiMax 和蓝牙™ 无线技术。由此，通信可以是如对于常规网络那样的预定义结构，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 网络可以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 相关介质和功能）。

[0069] 现在参考图 9，示出便于通过浏览器输入内容至应用程序的计算环境的示意框图。环境 900 包括一个或多个客户端 902。客户端 902 可以是硬件和 / 或软件（例如，线程、进程、计算设备）。例如，客户端 902 可以容纳 cookie 和 / 或相关联的上下文信息。

[0070] 环境 900 还包括一个或多个服务器 904。服务器 904 也可以是硬件和 / 或软件（例如，线程、进程、计算设备）。服务器 904 可以例如通过使用本体系结构来容纳线程以执行变换。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。环境 900 包括可以用来使客户端 902 和服务器 904 之间通信更容易的通信框架 906（例如，诸如因特网等全球通信网络）。

[0071] 通信可以经由有线（包括光纤）和 / 或无线技术来促进。客户端 902 可以在操作上连接到一个或多个客户端数据存储 908，可以使用这些客户端数据存储来存储客户端 902 本地的信息（例如，cookie 和 / 或相关联的上下文信息）。同样地，服务器 904 可以在操作上连接到可以用来存储服务器 904 本地的信息的一个或多个服务器数据存储 910。

[0072] 以上描述的包括所公开的体系结构的各示例。当然，描述每一个可以想到的组件和 / 或方法的组合是不可能的，但本领域内的普通技术人员可以认识到，许多其它组合和排列都是可能的。因此，该新颖体系结构旨在涵盖所有这些落入所附权利要求书的精神和范围内的更改、修改和变化。此外，就在详细描述或权利要求书中使用术语“包括”而言，这一术语旨在以与术语“包含”在被用作权利要求书中的过渡词时所解释的相似的方式为包含性的。

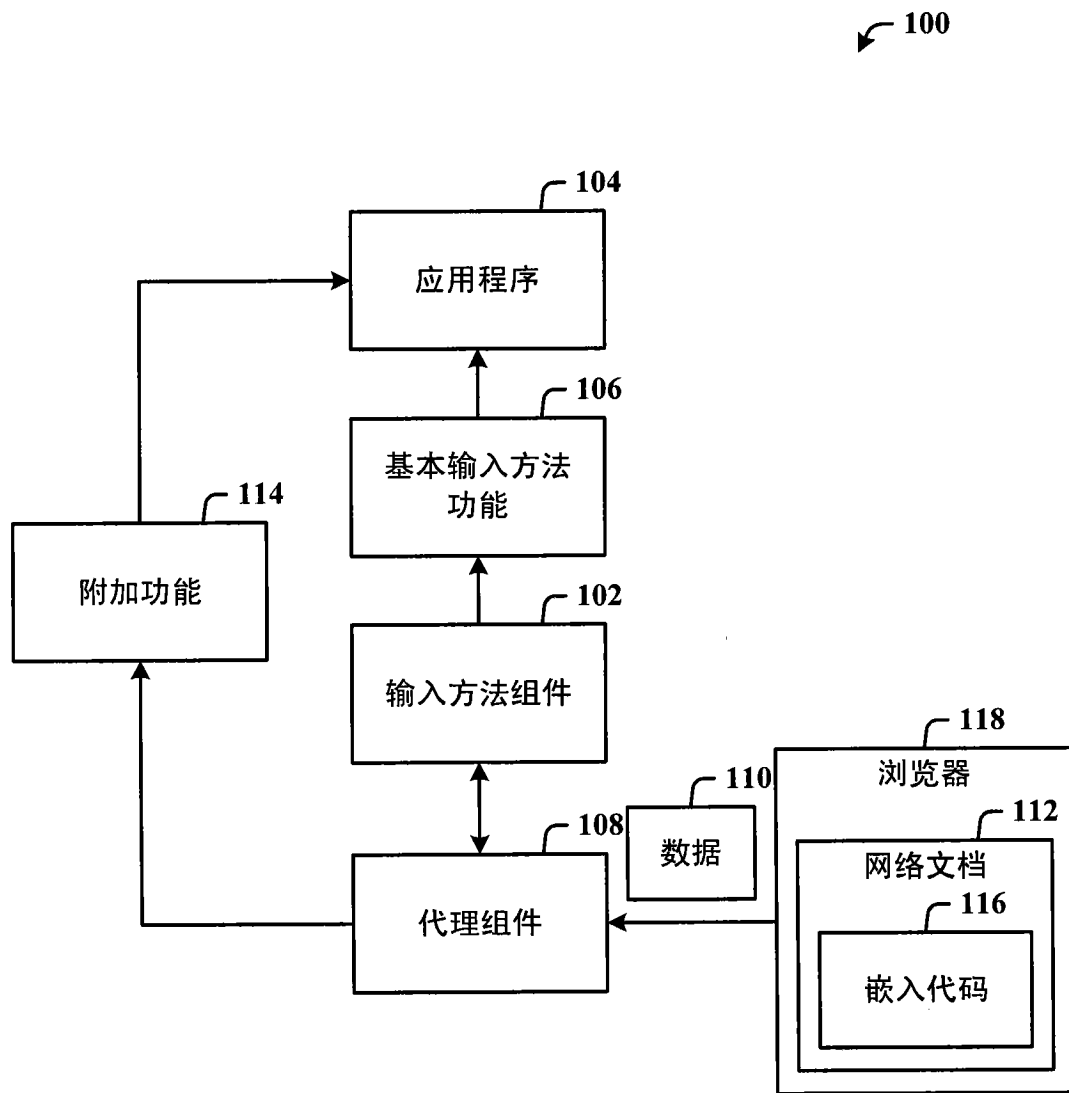


图 1

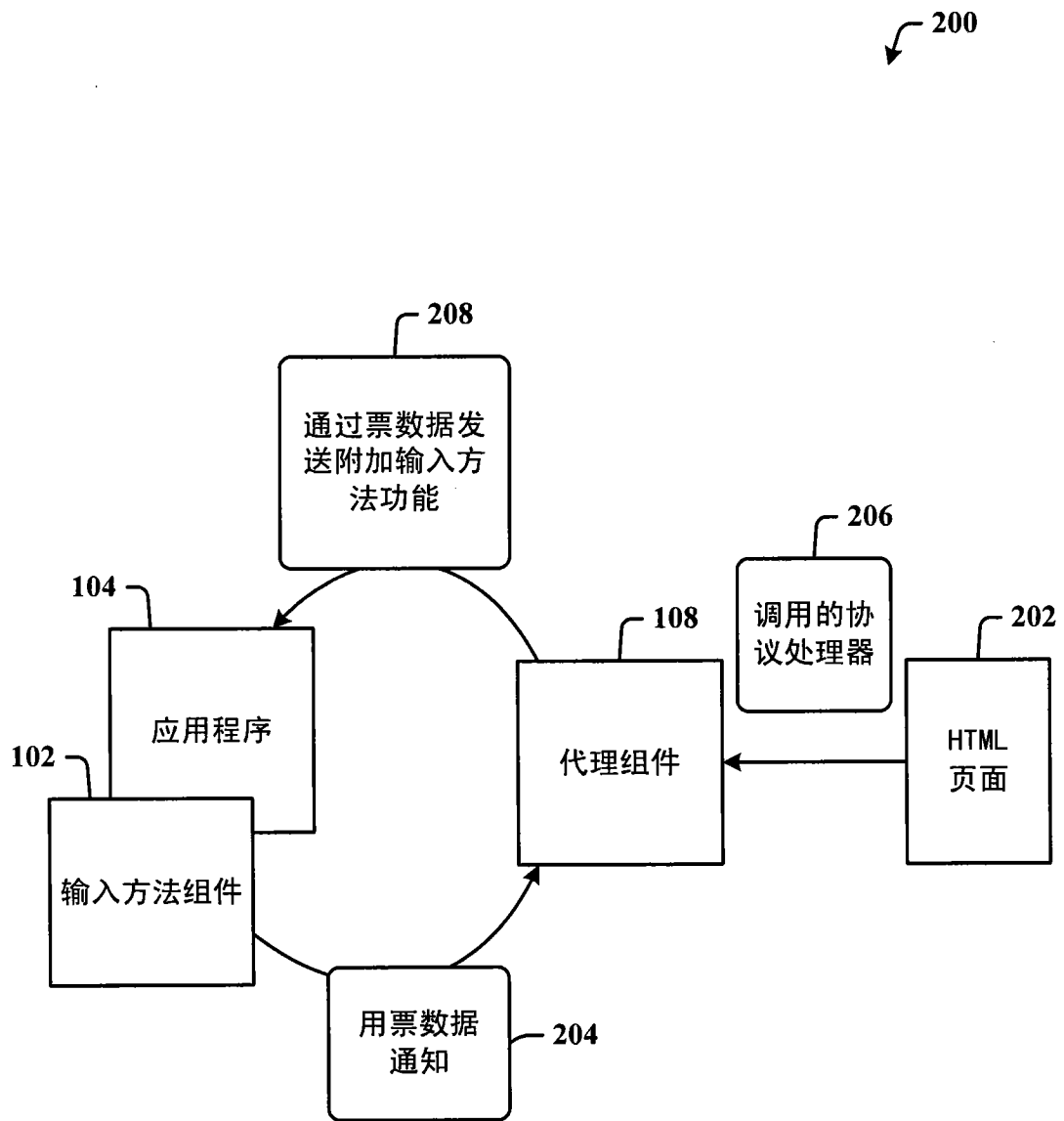


图 2

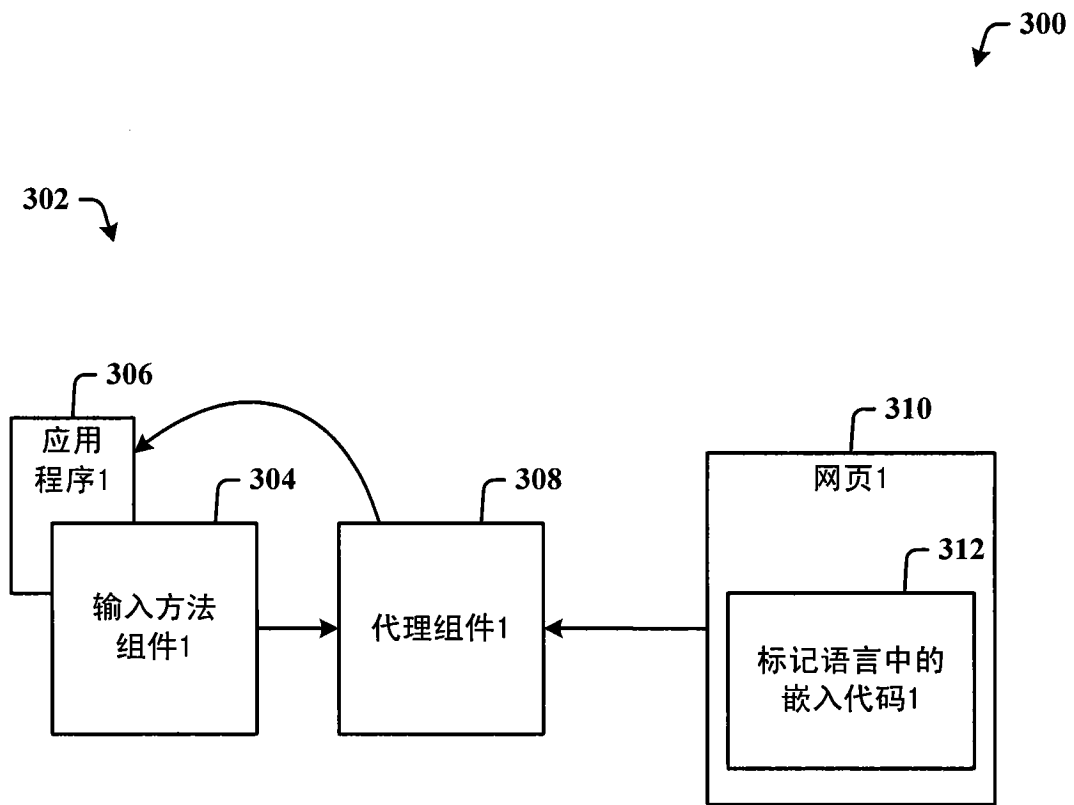


图 3

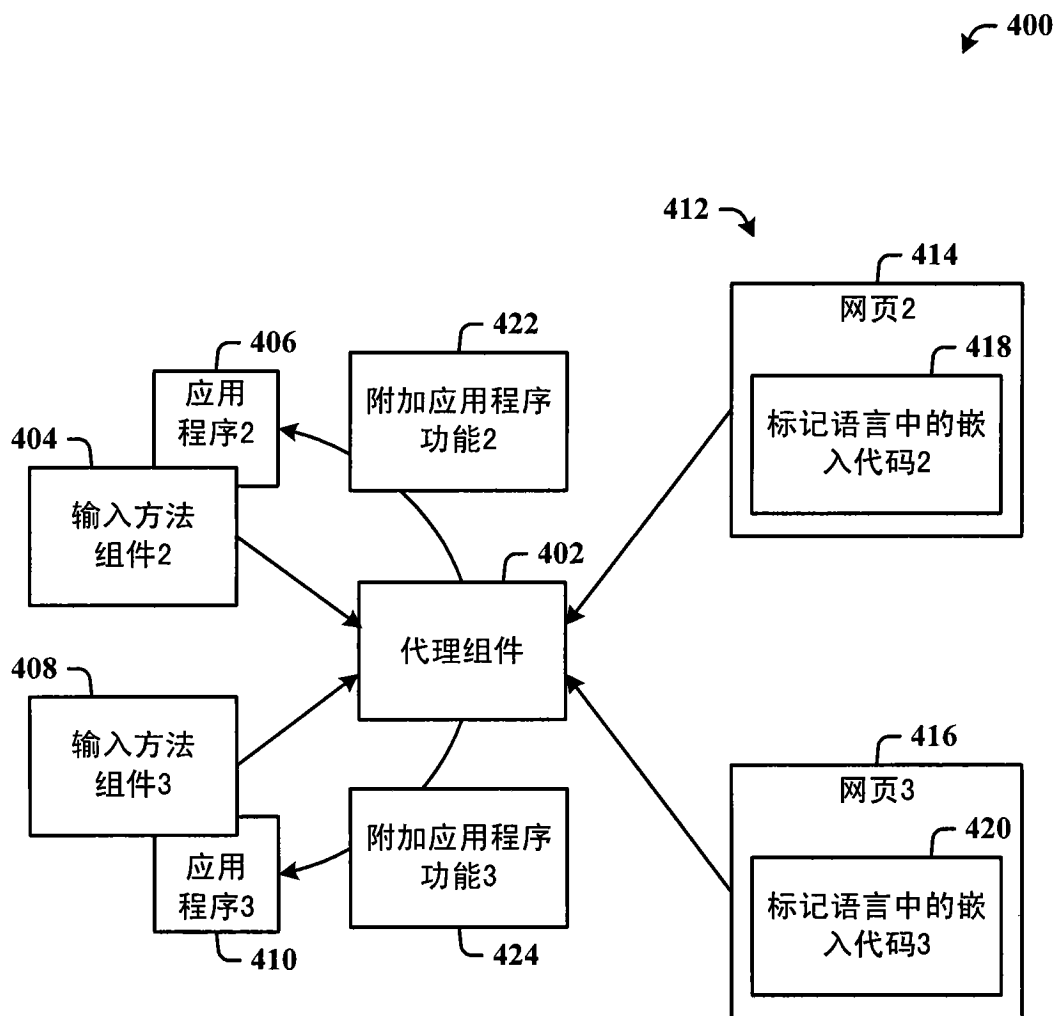


图 4

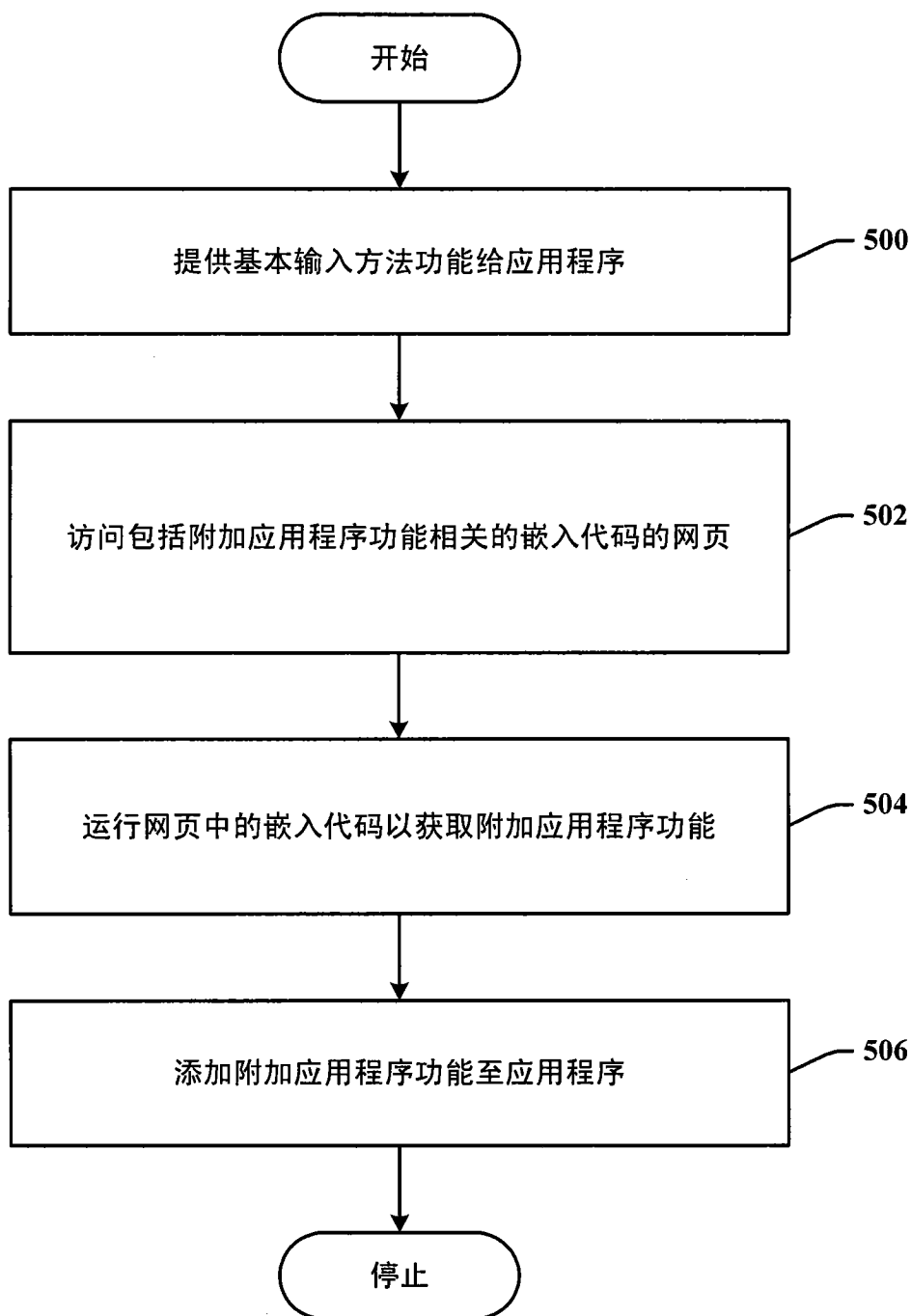


图 5

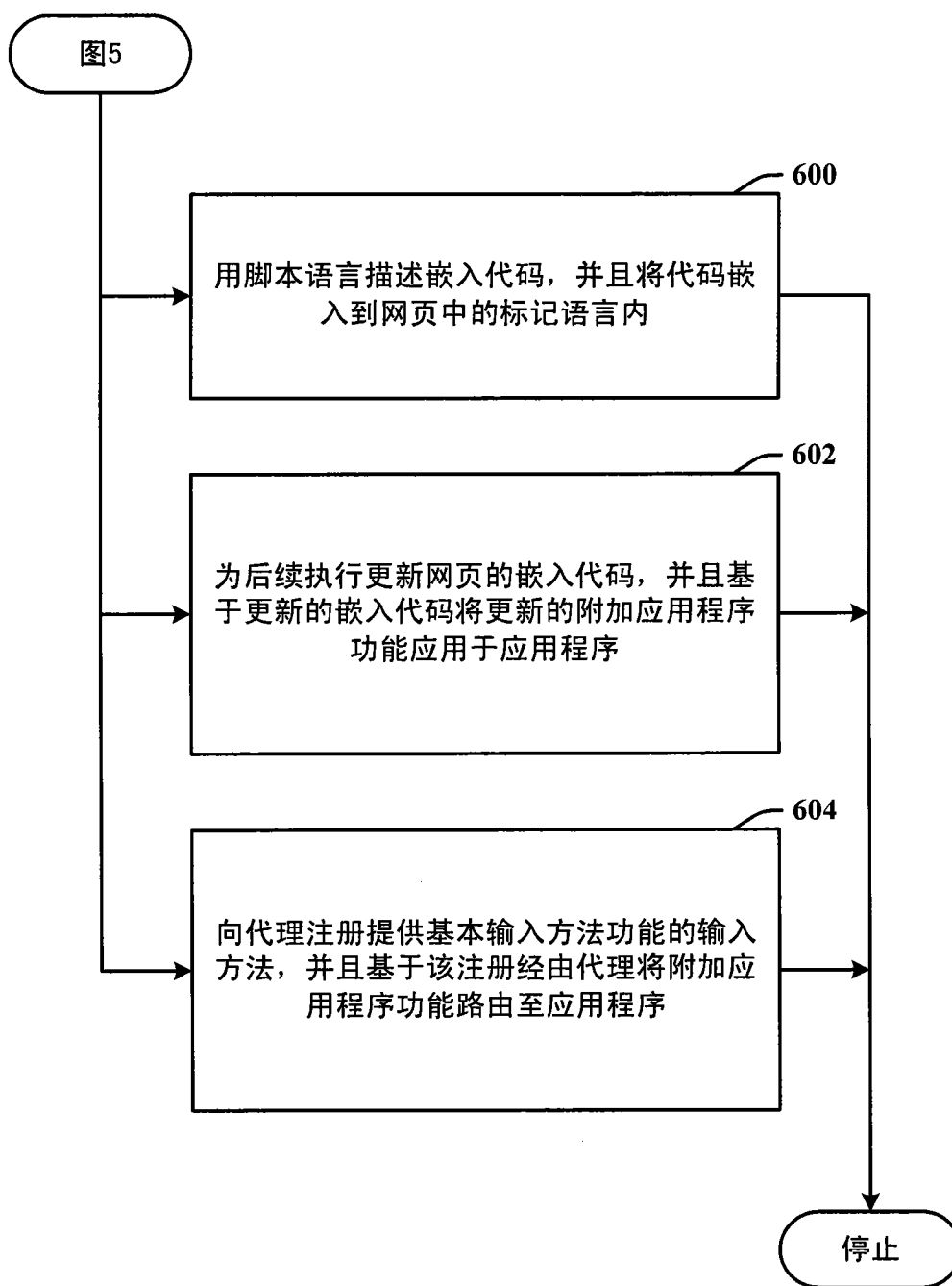


图 6

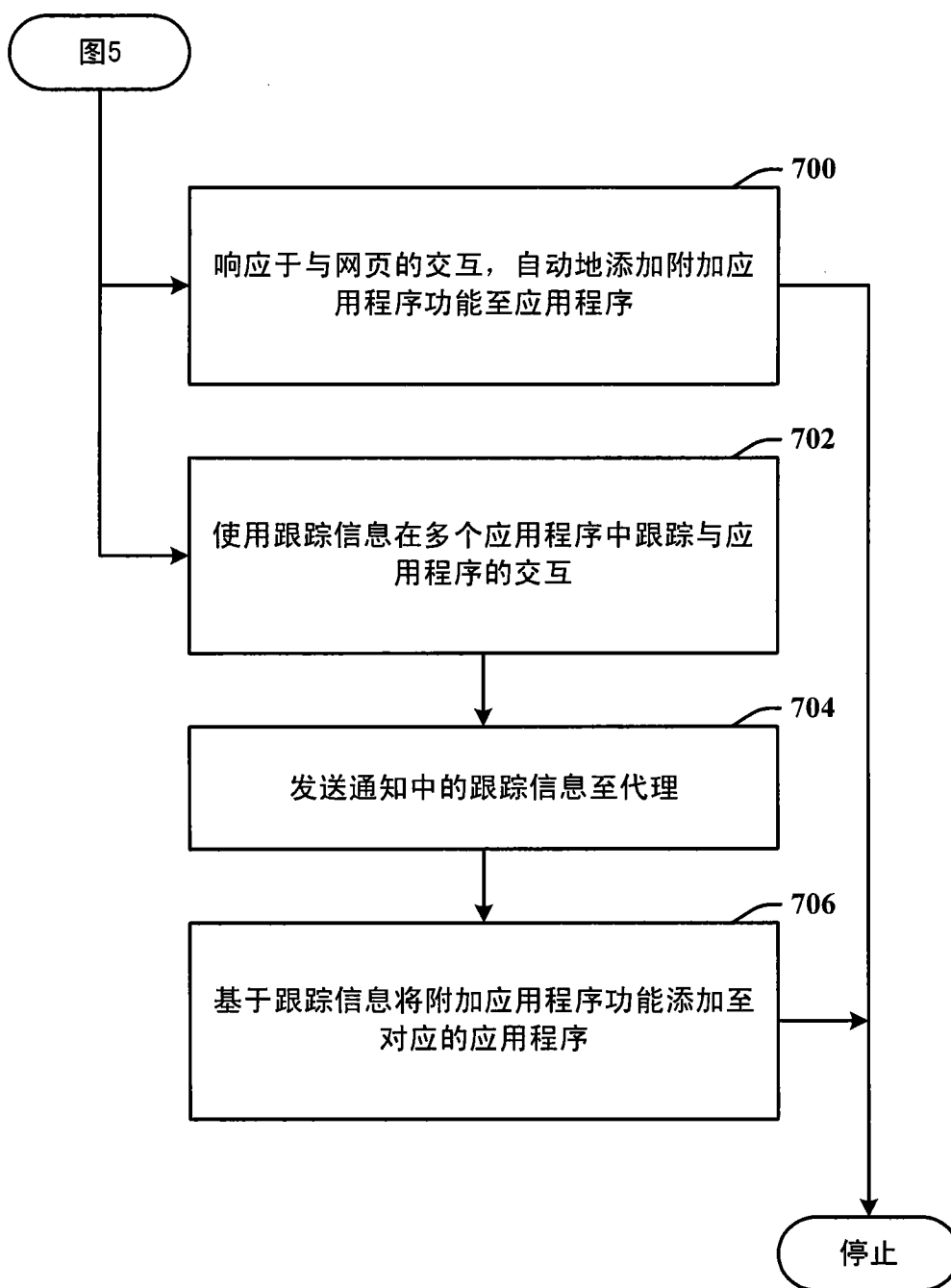


图 7

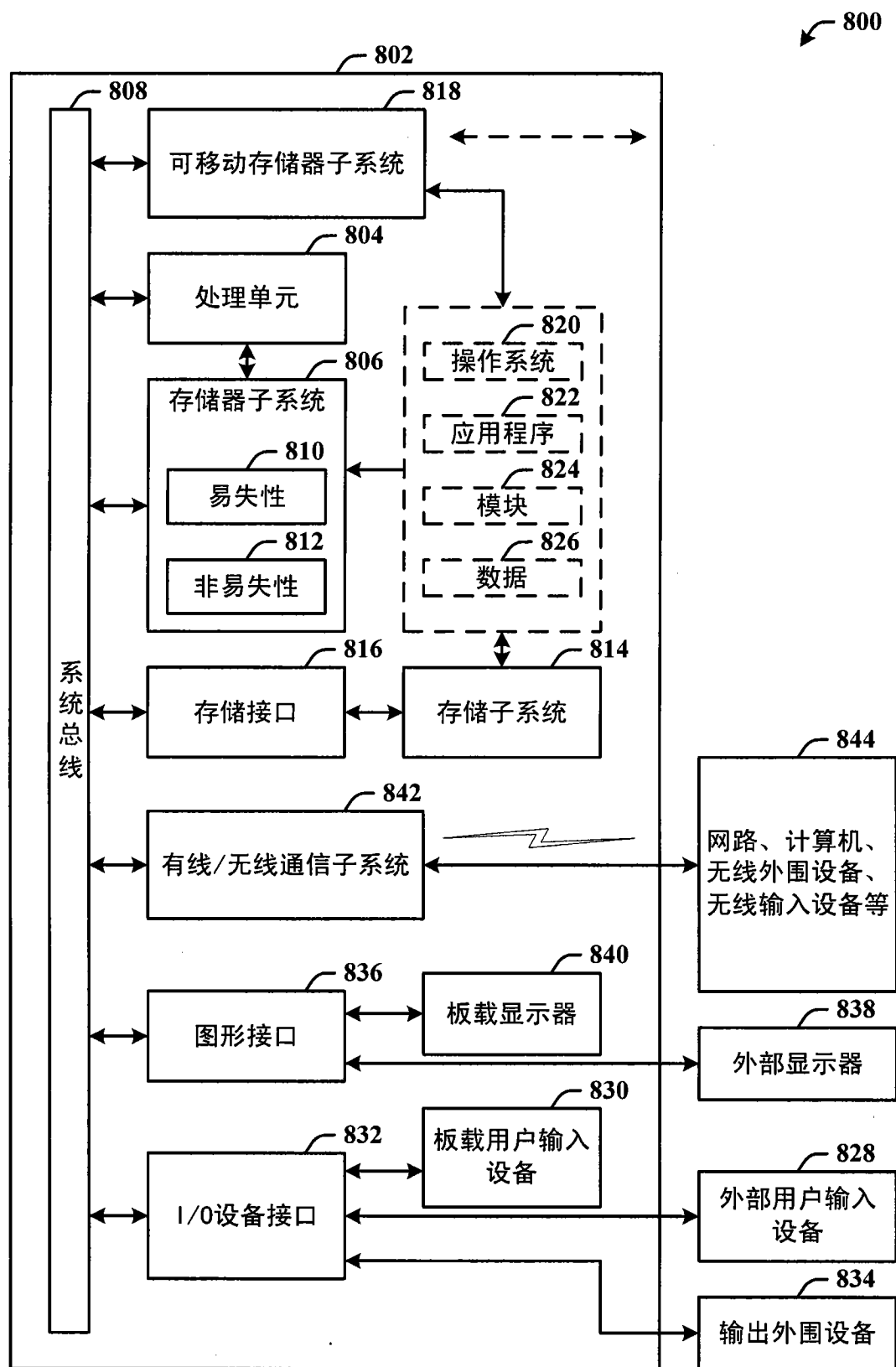


图 8

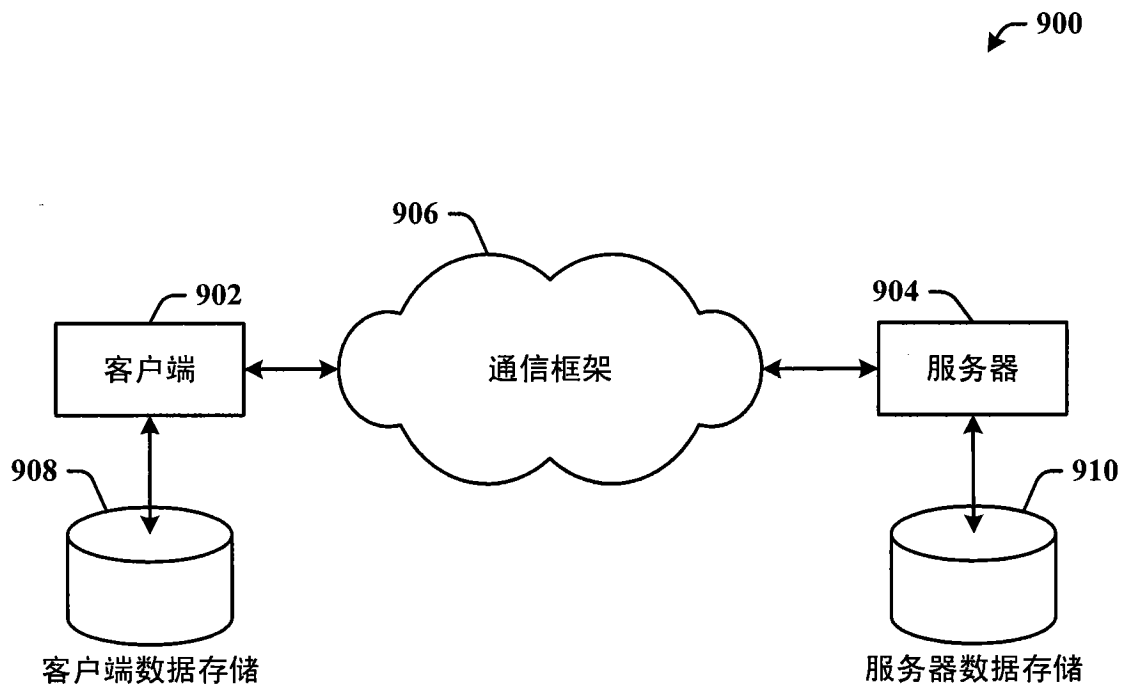


图 9