



# [12] 发明专利申请公开说明书

[21] 申请号 200410071486.4

[43] 公开日 2005年2月2日

[11] 公开号 CN 1574750A

[22] 申请日 2004.6.19

[21] 申请号 200410071486.4

[30] 优先权

[32] 2003.6.20 [33] US [31] 60/480424

[32] 2004.4.15 [33] US [31] 10/826053

[71] 申请人 西门子医疗健康服务公司

地址 美国宾夕法尼亚州

[72] 发明人 Z·方 E·马丁

[74] 专利代理机构 中国专利代理(香港)有限公司

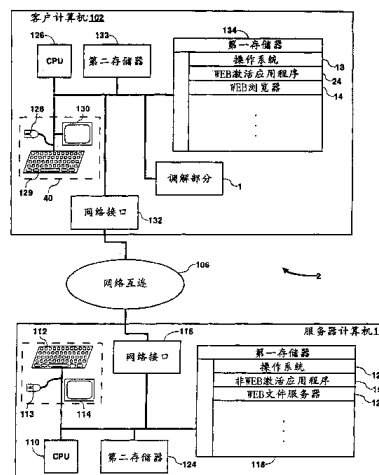
代理人 梁永

权利要求书4页 说明书11页 附图4页

[54] 发明名称 支持在网上激活应用程序和其它应用程序之间通信的系统

[57] 摘要

一种系统，它由至少一个 Web 浏览器兼容的可执行应用程序使用以启动第二可执行应用程序的动作，该系统包括：URL 处理器，它响应于用户命令而产生包含范围信息和动作请求标识符的预定特定格式的 URL 链接，该 URL 链接表示由所述第二可执行应用程序将要执行的动作。URL 链接解译器，它通过第一通信模式链接到所述 URL 处理器，用于识别所述特定格式的 URL，用于从所述特定格式的 URL 链接中提取所述上下文信息和动作请求标识符，并用于启动第一通信模式的终止。通信处理器，它连接到所述 URL 链接解译器，用于通过第二通信模式启动到第二可执行应用程序的所述上下文范围信息和动作请求标识符的通信，该第二通信模式不同于所述第一通信模式。



1. 一种由至少一个Web浏览器兼容的可执行应用程序使用以启动第二可执行应用程序的动作的系统，该系统包括：

5 URL处理器，它响应于用户命令而产生包含上下文信息和动作请求标识符的预定特定格式的URL链接，该URL链接表示由所述第二可执行应用程序将要执行的动作；

URL链接解译器，它通过第一通信模式链接到所述URL处理器，用于识别所述特定格式的URL，用于从所述特定格式的URL链接中提取所述上下文信息  
10 和动作请求标识符，并用于启动第一通信模式的终止；和

通信处理器，它连接到所述URL链接解译器，用于通过第二通信模式启动到第二可执行应用程序的所述提取的上下文信息和动作请求标识符的通信，该第二通信模式不同于所述第一通信模式。

2. 根据权利要求1的系统，其中：

15 所述第一通信模式包括（a）因特网兼容通信和（b）超文本传输协议通信中的至少一种；和

所述第二通信模式包括（i）微软Windows兼容消息通信和（ii）socket通信和（iii）COM通信中的至少一种。

20 3. 根据权利要求1的系统，其中所述第二可执行应用程序是非Web激活应用程序。

4. 根据权利要求1的系统，其中

所述用户命令是经过显示的网页被接收的；和

所述URL处理器将所述浏览器窗口对象的位置特性设置为所述特定格式的地址。

25 5. 根据权利要求1的系统，其中：

所述上下文信息包括（a）用户标识符，（b）患者标识符，（c）客户标识符，（d）来源标识符，（e）目的标识符，（f）密码，（g）计算机操作会话标识符，（h）识别第二可执行应用程序来执行所述用户命令的标识符，（i）和数据标识符中的至少一种；和

30 所述动作请求标识符，它识别（i）所述第二应用程序执行的动作的类

型, (ii) 所述用户命令的类型, (iii) 事件和 (iv) 鉴权业务中的至少一个。

6. 一种由至少一个Web浏览器兼容可执行应用程序使用以接收有关第二可执行应用程序执行的动作的状态信息的系统, 该系统包括:

接口处理器, 用于:

5 响应经过显示图像输入的用户命令以接收来自所述第二可执行应用程序的应用数据;

分析与所述显示图像有关的文件对象数据, 以找到预定程序标识符, 该标识符识别用于处理所述接收的应用数据与Web浏览器应用程序的相兼容的可执行程序; 和

10 响应来自所述接口处理器的命令, 启动执行所述识别的可执行程序, 以提供处理过的接收应用数据给所述Web浏览器应用程序。

7. 根据权利要求6的系统, 其中所述应用数据包括 (a) 命令标识符, (b) 数据和 (c) 与所述第二可执行应用执行的动作有关的状态信息中的至少一种, 其中:

15 所述图像包括显示的网页, 它被处理用于所述Web浏览器应用程序的显示; 和

所述文件对象数据包括与所述网页的文件有关对象模型。

8. 根据权利要求7的系统, 其中所述网页是由包含至少下述一个的数据来表示 (a) HTML兼容数据, (b) XML兼容数据和 (c) SGML兼容数据。

20 9. 根据权利要求6的系统, 还包括连接到所述接口处理器的通信处理器, 用于与所述第二可执行应用程序通信以获得所述应用数据, 其中所述通信处理器通过下述方式中至少一个方式与所述第二可执行应用程序相通信, (i) 微软Windows兼容消息通信和 (ii) socket通信和 (iii) COM通信。

25 10. 根据权利要求9的系统, 其中所述通信处理器建立与所述第二可执行应用程序的非轮询通信, 所述第二可执行应用程序响应下述至少一项内容将所述状态信息提供给所述通信处理器, (i) 所述第二可执行应用程序的命令和 (ii) 来自所述通信处理器的请求。

30 11. 根据权利要求7的系统, 其中所述Web浏览器应用程序启动所述接收的状态信息的显示, 其中所述接收的状态信息包括下述信息中至少一个 (a) 识别第二可执行应用程序的任务执行状态的标识符, (b) 状态说明和 (c) 文本

消息。

12. 一种由第一Web浏览器兼容的可执行应用程序使用以启动第二可执行应用程序的动作, 并获得有关所述动作的状态信息的系统, 该系统包括:

URL处理器, 它响应于经过显示浏览图像输入的用户命令而产生包含上下文信息和动作请求标识符的预定特定格式的URL链接, 该URL链接表示由所述  
5 第二可执行应用程序执行的动作;

URL链接解译器, 它通过第一通信模式连接到所述URL处理器, 用于识别所述特定格式的URL链接, 用于从所述特定格式的URL链接中提取所述上下文信息和动作请求标识符, 并用于启动第一通信模式的终止; 和

10 通信处理器, 它连接到所述URL链接解译器, 用于通过第二通信模式启动到第二可执行应用程序的所述上下文信息和动作请求标识符的通信, 该第二通信模式不同于所述第一通信模式; 和

接口处理器, 用于:

15 响应所述用户命令以接收与所述第二可执行应用程序执行的所述动作相关的状态信息;

分析与所述浏览图像有关的文件对象数据, 以找到预定程序标识符, 该标识符识别用于处理所述接收的应用数据与Web浏览器应用程序的相兼容的可执行程序; 和

20 响应来自所述接口处理器的命令, 启动执行所述识别的可执行程序, 以供处理过的接收状态信息给所述Web浏览器应用程序。

13. 一种由至少一个Web浏览器兼容的可执行应用程序使用以启动第二可执行应用程序的动作的方法, 包括步骤:

响应于用户命令产生包含上下文信息和动作请求标识符的预定特定格式的URL链接, 该URL链接表示所述第二可执行应用程序将要执行的动作;

25 启动所述URL链接在第一通信模式下的通信;

识别所述特定格式的链接;

从所述特定格式的链接中提取所述上下文信息和动作请求标识符;

启动第一通信模式的终止; 和

30 通过第二通信模式启动到第二可执行应用程序的所述上下文信息和动作请求标识符的通信, 该第二通信模式不同于所述第一通信模式。

14. 一种由至少一个Web浏览器兼容的可执行应用程序使用以接收与第二可执行应用程序执行的动作有关的状态信息的方法，包括步骤：

响应于通过显示图像输入的用户命令来接收与所述第二可执行应用程序执行的动作相关的应用数据；

- 5       分析与所述浏览器图像有关的文件对象数据，以找到预定程序标识符，该标识符识别用于处理所述接收的应用数据与Web浏览器应用程序的相兼容的可执行程序；和

启动执行所述识别的可执行程序，以提供处理过的接收应用数据给所述Web浏览器应用程序。

- 10       15. 根据权利要求14的方法，还包括步骤：

响应于接收的应用数据更新本地存储单元，还包括下述至少一个步骤：

请求数据；

执行程序；和

- 15       由于所述Web浏览器兼容的可执行应用程序的状态改变，响应来自Web激活应用程序的命令。

## 支持在网上激活应用程序和 其它应用程序之间通信的系统

5

### 技术领域

本发明通常涉及万维网客户机应用程序的网络互用性的领域，更特别的是涉及一个允许Web激活应用程序来请求其它应用程序和接收来自其它应用程序的状态更新的系统，尤其是对来自传统、非Web激活的软件应用程序的状态更新。

10

### 背景技术

万维网（WWW）是将许多Web服务器计算机链接到用户计算机的协议系统，它构成了因特网的重要部分。Web服务器计算机存储Web格式的文件，Web激活应用程序，还可以包括传统的计算机程序和各种其它的非Web应用程序。Web服务器可以经调节后允许经因特网连接到服务器的Web客户计算机来访问传统的计算机程序和应用程序。服务器计算机通过给每个文件和应用程序分配一个称为统一资源定位符（URL）的唯一地址，来允许访问上述文件和应用程序。用户通过向Web服务器发送URL来请求文件和/或应用程序，Web服务器给请求用户返回所请求的文件或应用程序。

15

有时被称为“胖客户”（“fat clients”）的传统非Web计算机应用程序典型的是通过人可读源代码的部分来定义。该源代码用一种高级编程语言来编写，该高级编程语言可以被编译或伪编译成机器可读码。计算机系统将源代码编译成可执行应用程序。可执行应用程序通常安装在执行机器可读码的计算机系统中。

20

Web计算机应用程序另一方面是由称为“瘦客户”（“thin clients”）的Web浏览器计算机程序来执行的。Web格式文件包括显示页面图像的指令。“瘦客户”主要依靠由Web浏览器应用程序实现的图形用户接口（GUI）来接收和解译Web格式文件中的指令，并根据那些指令在窗口中显示图像。这些显示指令包含很少的或不包含可执行码。当存在可执行码时它通常是脚本语言的形式，scripting语言是作为显示指令的一部分被动态传递到客户计算机，并由Web客户

25

30

计算机中运行的Web浏览器容器进行解译。Web浏览器容器限制了存在于客户计算机中的任何代码的功能范围。在某些操作过程中，Web客户计算机可能需要具有从传统胖客户应用程序中请求动作的能力和接收与那些请求有关的状态信息的能力。

- 5 现有系统都受到限制，仅提供在Web客户和胖客户之间的连接和通信方面的有限功能。例如，它们可以为Web客户提供向胖客户发送请求的能力，可以使Web客户具有接收来自传统胖应用程序的状态更新的能力。但是这种能力是需要Web服务器到Web客户的轮询来实现。也就是说，Web客户向Web服务器发送一个状态请求，Web服务器将该请求发送到胖客户。胖客户响应该请求将
- 10 状态数据发送给Web服务器，Web服务器将该数据发送到Web客户。当论询状态信息时，在Web服务器和Web客户之间包含的内部通信对Web服务器和Web客户的性能以及计算机网络的使用率来说都是一种负担。

一种用于减少在执行轮询状态时所需的内部通信的方法是在Web客户计算机中设立可执行码，以提供在Web浏览器和胖客户之间的直接通信。然而，在

15 Web客户计算机中导入非Web码会引起安全性和代码分配的问题，这些问题并不仅仅存在于Web兼容应用程序和业务中。通常，Web兼容应用程序不需要在客户计算机中进行代码分配或安装。为了使Web客户的用户能够获得服务或经过因特网访问应用程序，用户只需要简单导航到业务或应用程序所在的URL。但是，如果在理想业务或应用程序的Web客户中需要非Web码，例如ActiveX控制，

20 用户就需要在访问该业务或应用程序之前安装该代码。非Web码可以引起客户计算机易受到未经许可进入的攻击。未经许可的用户可以借助于由不老练客户使用者所安装的非web码，或者借助于未经许可的用户所熟知的一些其它攻击方法从计算机系统中提取敏感信息。

#### 发明内容

- 25 根据本发明的原理，提供一种系统，它由至少一个Web浏览器兼容可执行应用程序使用以启动第二个可执行应用程序的动作。所述系统包含下述部分：URL处理器，它响应用户命令产生预定特殊格式的URL链接，该预定特殊格式包含上下文信息和动作请求标识符。该URL链接表示由第二可执行应用程序所执行的动作。URL链接解译器，它经过第一通信模式连接到所述URL处理器，
- 30 识别所述特殊格式的URL链接，从特殊格式URL链接中提取上下文信息和动作

请求标识符，启动第一通信模式的终止。通信处理器，它连接到所述URL链接解译器，通过不同于所述第一通信模式的第二通信模式，启动到第二可执行应用程序的所述提取的上下文信息和动作请求标识符的通信。

#### 附图说明

5 图1是依据本发明的原理在Web和非Web激活应用程序之间提供通信的计算机网络的方框图；

图2是显示图1中所示的本发明结构的方框图；

图3 是由图2中所示的本发明执行操作顺序的流程图；

10 图4是显示本发明所执行的基于Web的应用程序与非Web激活应用程序之间的交互作用的流程图。

#### 具体实施方式

参照图1，叙述了一个包括至少一个客户计算机102的计算机网络2，客户计算机102通过网络互连106互连到至少一个服务器计算机104。在本发明中网络互连106典型的是一些因特网部件，尽管它也可以是局域网或广域网的部件，或者是在单个计算机中的程序之间建立的通信信道。客户计算机102包  
15 括中央处理单元（CPU）126，包含有诸如鼠标128，键盘129和显示器或监视器130等装置的用户接口。客户计算机102还包含第一存储器134，诸如磁盘驱动器的第二存储器133，以及网络接口132。网络接口132允许客户计算机102与服务器计算机104通过网络互连106进行通信。

20 客户计算机102的第一存储器134存储有操作系统13和网络或Web浏览器14，这两者可以从第二存储器133中检索或加载。在本发明中，Web浏览器14是有JavaScript能力的程序。JavaScript可以从由加利福尼亚的Sun Microsystems of Mountain View开发的Java编程语言中得到。典型的Web浏览器14在商业上可用的形式是例如来自加利福尼亚的JavaSoft of Cupertino的HotJava，可用于来自加利福尼亚的Netscape Communications of Mountain View的Netscape Navigator，以及  
25 可用于来自华盛顿的Microsoft Corporation of Redmond的Internet Explorer。

第一存储器134还存储有Web文件和Web激活应用程序24，所述Web激活应用程序可以从服务器计算机104中检索到。Web浏览器14和操作系统13都由CPU126运行。操作系统13响应从鼠标128和键盘129中接收的用户命令来协调  
30 和控制Web浏览器14的功能。

同样的，服务器计算机104包含CPU110，输入装置112和113，监视器或显示装置114，网络接口116，第一存储器118和第二存储器124。网络接口116允许服务器计算机104与客户计算机102通过网络互连106进行通信。服务器计算机104的存储器118存储了操作系统120，Web文件服务器122，和一个或多个非Web激活应用程序150。在本文的应用中，非web激活应用程序150是一个不依赖于它所操作的Web浏览器技术的可执行应用程序。服务器和客户机的结构、维护和操作都是本领域普通技术人员所熟知的内容。

客户计算机102的Web浏览器14包括有控制核心，用于调节CPU126依据Web地址（例如统一资源定位符（URL）参数）从服务器计算机104中检索网页和包含在URL参数中的其它数据。网页可以由包含下面至少一个数据来表示，(a) HTML兼容数据，(b) XML兼容数据和 (c) SGML兼容数据。检索到的网页可以包括Web激活应用程序24。Web浏览器14也可以执行将非Web激活应用程序150从服务器计算机104下载到客户计算机102的命令。然而，如上所述，这将引起在客户计算机102中的安全问题。

客户计算机102包括调解部分1，它能使客户计算机102产生对服务器计算机104上的非Web激活应用程序150的动作请求，和接收来自服务器计算机104上的非Web激活应用程序150的状态更新。由于非Web激活应用程序150并不是由客户计算机来传送和执行的，因此在客户102中不使用非Web技术。调解部分1不需要Web服务器104的请求或轮询来获得来自非Web应用程序150的状态信息。这个优点消除了包含在使用诸如Web客户接口132内例如Active X码的某些非Web应用程序中的安全问题。调解部分1也减轻了在实现Web客户14和非Web应用程序150之间通信时的分配代码的负担。

本文使用的可执行程序包括在例如响应用户命令或输入时用于实现预定功能的代码或机器可读指令，所述预定功能包括操作系统、保健信息系统或其它信息处理系统的功能。本文使用的可执行程序是用于执行一个或多个特殊处理的代码段（机器可读指令），子程序或其他特殊的代码段或可执行应用程序部分，它可以包括执行对接收到的输入参数（或响应于接收到的输入参数）的操作，并提供结果输出参数。呼叫程序是响应于接收到的命令或指令能够执行其他程序的程序。本文使用的处理器是执行任务的装置和 / 或机器可读指令集合。如本文中所述的内容，处理器包括硬件、固件和 / 或软件中的任一个或其

组合。处理器根据操作、分析、修改、转换或发送由可执行程序或信息装置使用的信息进行动作，和 / 或根据通过路由到输出装置的信息进行动作。处理器可以使用或包括例如控制器或微处理器的能力。本文使用的对象包括数据组，可执行指令或这两者的组合或可执行程序。

5 图2是叙述根据本发明实现的软件结构的方框图。软件可以在工作站的环境中或分布的环境中实现。参照图2，传统的非Web激活应用程序150包括应用逻辑3，它指导传统应用程序150的操作。传统应用程序150的应用逻辑3与协调代理4的协调部分1相互作用。类似的方式，Web客户14包括应用逻辑7，它指导Web浏览器和Web浏览器接口逻辑132的操作。Web客户14的应用逻辑7与Web  
10 浏览器接口逻辑132相互作用，反过来Web浏览器接口逻辑132可以与外部机构相互作用，诸如通过发送URL请求和接收响应Web网页和Web应用程序的Web服务器。在典型的工作站环境中，Web客户14接口逻辑132与Web客户主应用2的接口逻辑5相互作用。Web客户主应用2的接口逻辑5也与协调代理4的协调部分1相互作用。

15 在Web客户14内，网页和相关的Web应用程序可以以周知的方式组织成对象的分级结构。每个这种对象被命名，通过通常称为特性的相关数据来定义，并由通常称为方法的子程序来控制。此外，可以响应于事件来控制对象，或者例如响应于可以由用户产生的对该事件特性的变化来产生事件。例如，网页内的全部对象是“window”。“window”对象可以包含其它对象，诸如“  
20 document”。“window”对象还包括多个特性，所述特性包含具有表示URL数据的“location”特性，该URL是定义网页和Web激活应用程序的指令源。“window”对象还可以包括用户控制对象，诸如“buttons”。当用户激活“button”对象时，例如通过使用鼠标在按钮的显示图像上点击时，产生一个“单击”事件，这是可以导致其它对象变化的过程。

25 Web客户主应用2还包括IWebBrowser 2控制8 Active X 控制。IWebBrowser 2控制8 Active X 控制是微软公司的产品。Active X控制是组件对象模型（COM）对象，它可以被嵌入到微软窗口平台运行的其它应用程序中去。IWebBrowser 2控制8 Active X 控制可以响应由对象产生的事件。另外，当对象的身份被Web客户14本身获知时，它们通常不能由其它应用程序所获知和使用。IWebBrowser  
30 2控制8允许外部的应用程序来确定在Web客户14中定义的对象身份。

存在于Web客户14内部的Web客户接口逻辑132仅仅利用基于Web的技术，诸如欧洲计算机制造协会（ECMA）Script 语言ECMA-262，也称为JavaScript。JavaScript程序是客户端应用程序，它典型的是可以随着定义网页的数据一起被发送到Web浏览器14的小程序，用来执行动画制作，以及使用客户计算机102的计算资源来执行计算或其它直接发送的任务，而不需要向服务器104发送请求来进行更多处理。

图3叙述了本发明执行的操作顺序。起初，可供Web客户14访问使用的传统应用程序150发送一个指示其可用性9的通知给协调代理4。当Web客户主应用2（图2）被启动时，Web客户主应用接口逻辑5执行客户注册功能26，接着启动查询11以确定传统应用程序150的可用性。响应12，调解代理4返回给Web客户主应用接口逻辑5一可用的传统应用程序150的列表。

接着，在来自特定传统应用程序150的动作是预期的情况下当Web客户14在其操作期间到达某点时，Web客户14启动查询10以确定预期的传统应用程序150是否可用。查询10被发送到Web接口逻辑132，接着作为请求13被发送到Web客户主应用接口逻辑5。Web客户主应用接口逻辑5返回给Web客户机接口逻辑132一个消息12，它表示预期的传统应用程序是否可用。该消息12接着被发送给Web客户14。

如果由可用性返回消息12所表示的请求的应用程序150可用时，例如按钮等动作触发控制25以及用于表示动作触发控制15代表的动作的数据就会显示在Web客户14的网页上。如果找不到所请求的应用程序150，就不显示动作触发25或将它显示为禁止（变灰）模式。

如果用户通过鼠标128或键盘129激活动作触发控制25，然后Web客户14执行多个处理功能，这包含根据Web客户的当前状态来集合必要的上下文信息。Web客户14也产生对传统应用程序150的请求，通过将传统应用程序150的标识符、预期动作、集合的上下文信息传送给Web客户接口逻辑132来执行预期的动作。请求15随后作为消息16被发送给Web客户主应用接口逻辑5，接着作为消息17被发送给作为请求17的调解代理4，最后作为消息18被发送给所请求的传统应用程序150。

响应于对动作18的请求，传统应用程序150在间隔19期间执行或尝试执行所请求的活动。当活动被执行时，应用程序150异步地产生并发送一个或多个

中间状态消息20给调解代理4，尽管在某些情况下，传统应用程序可以不返回任何中间状态消息。这些状态消息可以包括一个或多个：用于识别所述第二可执行应用程序的任务执行状态的标识符；状态说明；和 / 或包含例如文本形式的状态信息的文本消息。产生的状态消息20被发送到调解代理4，Web客户主应用接口逻辑5，Web客户接口逻辑132并最后被发送到Web客户14。Web客户14可以在表示来自传统应用程序150的中间状态消息20的显示网页上显示消息21。在完成请求动作19之后，或由于误差情况引起动作的中断之后，传统应用程序150可以通过向调解代理4发送表示最后状态的消息22来返回最后状态。最后状态消息22随后被发送到Web客户主应用接口逻辑5，Web客户接口逻辑132并最终发送到Web客户14。在所请求的动作已经完成或中断时，Web客户14可以显示最后状态的指示23，也可以将显示的网页改变到其它的状态。

通过由IWebBrowser2 控制8（图2）提供的能力应用就可以实现从Web客户接口逻辑132到Web客户主应用接口逻辑5的通信。具有IWebBrowser2 控制8的Web客户主应用2的实现或集成允许Web客户主应用2从Web浏览器14内部捕获事件，诸如WWW（万维网）的站点和画面导航事件。例如，一个“OnBeforeNavigate2”的事件通知Web激活应用程序：Web浏览器14将要导航到一个新的URL。在Web浏览器14导航到那个URL之前，在用户访问（例如“点击”）新URL链接的任何时候，通过IWebBrowser2 控制8访问“OnBeforeNavigate2”的事件允许通知给Web客户主应用2。在Web浏览器14导航到那个URL之前，Web客户主应用2可以接着分析目标URL和所有传递的数据。Web客户主应用2具有取消已提出的导航，允许提出的导航继续和 / 或执行其它处理的选择权。

每当导航被请求到新的URL时，请求该导航的对象识别就可以用于Web激活应用24（图1）。Web激活应用24最初建立一个画面对象，它是不可见的或隐蔽的超文本标记语言 / JavaScript（html/jsp）画面。如上所述，如果预期的传统应用程序150可用，那么Web浏览器将诸如按钮用户控制放置在显示的网页上。如果用户希望传统应用程序150执行动作，用户就激活该用户控制。作为响应，Web激活应用24调节隐藏的画面对象来请求导航到预定的URL，它包括表示预期的传统应用程序150的数据、预期的动作和从网页内的其它对象中收集的上下文信息。该URL在例如正常URL请求通信的第一通信模式中通过Web客户接口逻辑132借助于IWebBrowser2 控制8被传送到Web客户主应用2。这就

产生了一个“OnBeforeNavigate2”的事件。

在接收到来自IWebBrowser2 控制8的“OnBeforeNavigate2”事件时，Web客户主应用接口逻辑5判断预期的URL是否来自隐藏的画面。如果是这样，Web客户主应用接口逻辑5就保存该URL和相关数据并取消导航。Web客户主应用接口逻辑5可以将存在于导航URL内的参数数据传递到Web客户主应用接口逻辑5。这个传递可以在第二通信模式下执行，例如通过功能呼叫参数的传送。

通过由微软IWebBrowser2 控制8（图2）提供的新的能力应用也可以实现从Web客户主应用接口逻辑5到Web客户接口逻辑132的通信。IWebBrowser2 控制8最初要获得表示Web激活应用24组件对象模型（COM）的数据，例如所有对象和它们之间关系的编辑。从该数据中可以判断Web激活应用24的数据是否包括画面。如果包括，就产生称为计数器的画面和它们的标识的列表。这个计数器允许访问单独文件形式的每个画面。反过来，该文件可以包含JavaScript对象。更特别的，隐藏的html/jsp画面包括一个特定JavaScript，当访问该特定JavaScript时其允许数据以功能呼叫参数形式被传送到Web激活应用24。

在Web客户主应用2中的循环程序被用于分析在原始源Web浏览器14文件内的画面。当设置隐藏的html/jsp画面时，就在画面文件中设置一个特定JavaScript功能的标识符。这通过首先执行包含在该画面内文件中的IDispatch功能来完成。IDispatch功能可以将正整数标识数字与指定命名的对象相联系。接着使用GetIDsOfNames功能来设置标识数据，该标识数据被称为在不可见html/jsp画面中特定的JavaScript功能。现在可以采用预期的参数来执行特定的JavaScript功能。使用这种技术，Web客户主应用接口逻辑5可以直接访问存在于隐藏的html/jsp内的JavaScript功能，以便将数据传送给Web浏览器14。接着，Web客户14响应于接收到的数据更新显示的网页。例如，可以接收和显示传统应用程序150运行的当前状态。

图4是叙述Web客户14如何与传统应用程序150相互作用的流程图400。在步骤402，Web客户14通过用户接口40的监视器130显示菜单、列表或一些其他图像，这允许用户随着动作控制（例如标记“拨号”的按键）选择某些上下文信息（例如，从客户列表中选择客户姓名和他们的电话号码）。在更通常的情况下，上下文信息可以包括如下数据，诸如用户标识符、患者标识符、客户标识符、来源标识符、目的标识符、密码、计算机操作会话标识符，和用于识

别实现用户命令的第二可执行应用程序的标识符，或数据标识符和 / 或任何其他必须供应给传统应用程序150以允许它正确执行其功能的数据。

用户选择希望的上下文信息（例如希望的客户和电话号码）并激活动作控制（例如“拨号”按钮），从而选择当前的上下文和动作。作为响应，Web浏览器14呼叫用于“拨号”按钮的JavaScript“点击”事件处理器。“点击”事件处理器收集当前选择的上下文（例如客户和电话号码），并使用当前选择的上下文参数（客户和电话号码）和请求动作（拨号）来启动在Web客户接口逻辑132中的“请求行动” JavaScript方法或功能。

在步骤404，在Web客户接口逻辑132内的“请求行动”功能担当URL处理器，它将当前的动作和上下文（客户和电话号码）连同特定标识符字符串一起编码成特定格式的URL42，例如表现为：

```
specialcode?cust="joe"&phone="3483748"&action="dial.
```

该特定标识符字符串可以表示由第二应用程序执行的动作的类型，用户命令的类型和 / 或事件或鉴权业务。

“请求动作”方法接下来使用JavaScript将“窗口”对象的“位置”特性设置为特定格式的URL。在步骤406这启动到新URL的导航，并从而触发如上所述在Web客户主应用接口逻辑5中的导航事件的接收者。导航事件接收者是例如在微软IWebBrowser2 ActiveX 控制中的“OnBeforeNavigate2”事件处理器。可替换的，导航事件接收者可以是在Mozilla 插入浏览控制（Mozilla浏览器可以从加利福尼亚，Mountain View，Mozilla Foundation中获得）中的“onStartURIOPEN”事件处理器。导航事件接收者检测特定格式的URL。Web客户主应用接口逻辑5担当URL解译器，从请求中提取上下文（客户姓名和电话号码）和请求的动作，并取消到该URL的导航。提取的上下文和动作数据被发送到Web客户主应用2。

在步骤408，Web客户主应用2担当通信处理器。它停止第一通信模式（例如正常http URL 请求）并发送请求44到传统应用程序150，以使用接收的上下文（客户姓名和电话号码）执行所请求的动作（拨号）。本文使用的请求包括例如启动动作的消息，这些动作诸如是打开或显示图像，检查报告、执行程序 and / 或批准图像等等。通过使用第二通信模式，例如所有合适的标准通信机构（COM，Windows 消息，socket通信等）将请求44发送到传统应用程序150。

在到传统应用程序150的请求44中，主应用14也请求通过执行应用程序150产生和发送状态信息。如本文所述的内容，状态信息包括下述中的一个或多个相关的信息：正在检索的数据或信息，开始或中断程序的指示，完成多步骤程序中一个步骤的指示和 / 或成功完成程序的指示。在步骤410，传统应用程序150执行所请求的动作，并在过渡基础上和 / 或在最后状态时提供状态信息。状态信息或请求可以是在传统应用程序150到Web客户14的任一传输方向，或者从Web客户14到传统应用程序150的任一传输方向。

在执行请求动作的传统应用程序150产生状态信息（“拨号电话”、“呼叫中断”、“呼叫完成”等）时，传统应用程序150使用第二通信模式，例如所有合适的标准通信机构（COM，Windows 消息，socket通信等）将状态信息作为消息45返回给Web客户主应用2，Web客户主应用2使用如上所述的IWebBrowser2控制8来检索当前“窗口”对象的文件对象模型。接着，Web客户主应用2使用GetIDsOfNames方法通过文件对象模式进行导航来设置特定的JavaScript功能，以获得特定JavaScript方法的鉴别参数。现在采用表示接收到的状态信息46的参数来命名特定的JavaScript功能。这允许Web客户主应用接口逻辑5接收功能参数形式的状态数据并将其传送到Web客户接口逻辑132，而不需要Web客户14进行论询。Web客户14可以改变显示图像以反映接收的状态。

尽管是参照如上所述的特定图表和实例对本发明进行描述的，但是在不脱离本发明基本原理的情况下可以在本系统中加入多种改进或特征。例如，本发明的第二实施例还可以包括一个事件处理器程序，用来识别来自Web浏览器14的指示网页将被加载的事件标识符。该程序接着还使用用户可定义的逻辑来分析Web地址，它规定了来自网页的附加数据和命令（以URL参数的格式）。如果代码逻辑识别出预定命令或数据，第二程序就处理来自Web浏览器14的命令或数据。

本发明的第三实施例可以使用基于例如JavaScript实现的用户可定义逻辑的程序，以便完成更多的下述功能。首先，系统可以启动包含事务逻辑的其它程序，以便从本地应用程序或 / 和从网页中提供信息。其次，系统可以直接在应用程序中显示信息。而且，系统可以启动程序以命令本地非Web应用程序的功能，例如发送请求给用户接口应用。

可以使用附加的程序来进一步命令与Web无关的本地功能。这种程序可以

---

是，例如由于与Web应用相关的状态或配置的改变而更新本地存储器。这种程序也可以请求数据，执行程序 and 任务，和 / 或响应来自Web应用的命令。另一个可实行的程序可以将数据或命令传递到Web浏览器14，以显示在网页上或由来自启动非Web的应用程序中的Web激活应用程序来处理，和 / 或返回关于数据或命令是否被Web激活应用程序所接受或拒绝的结果。

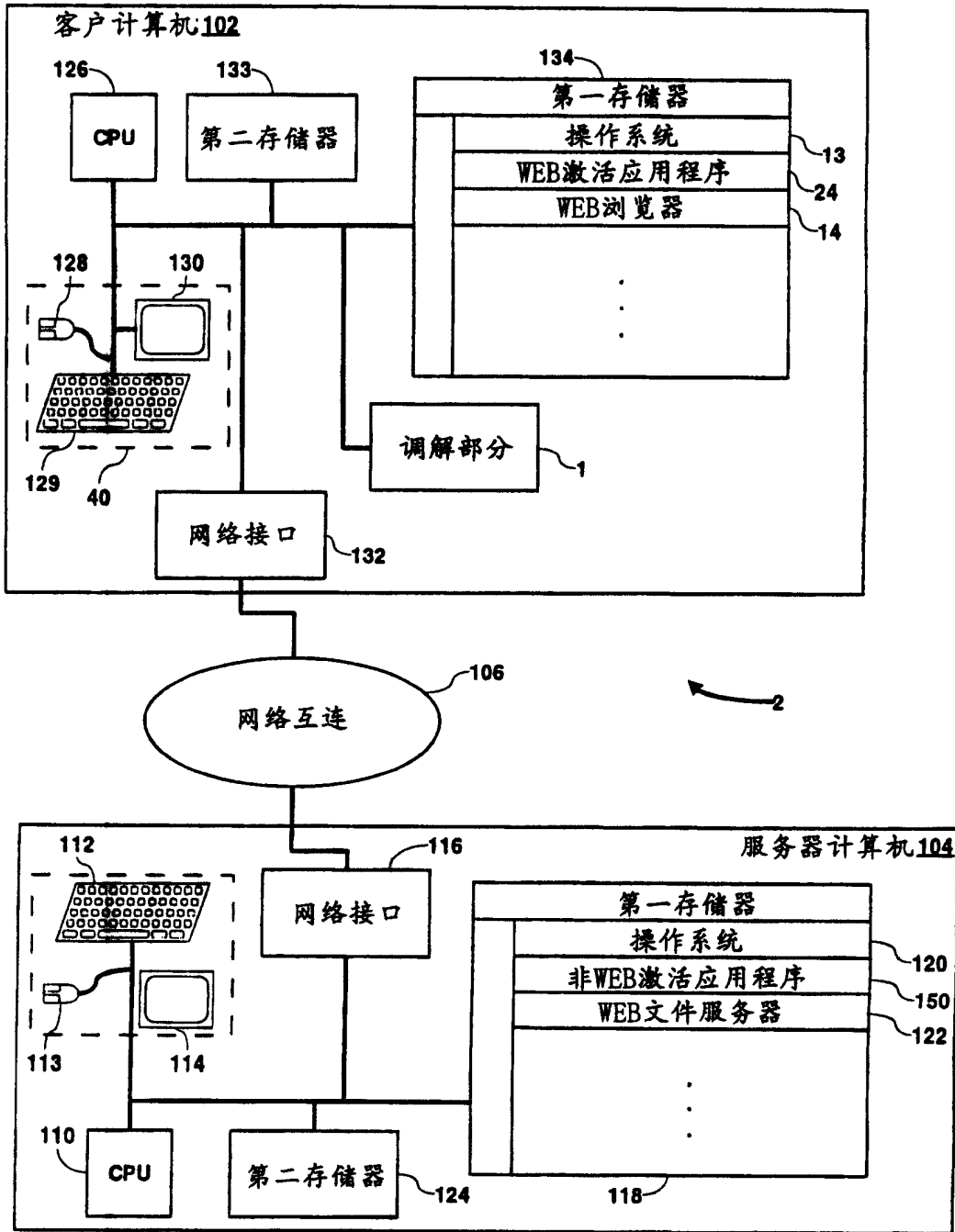


图 1

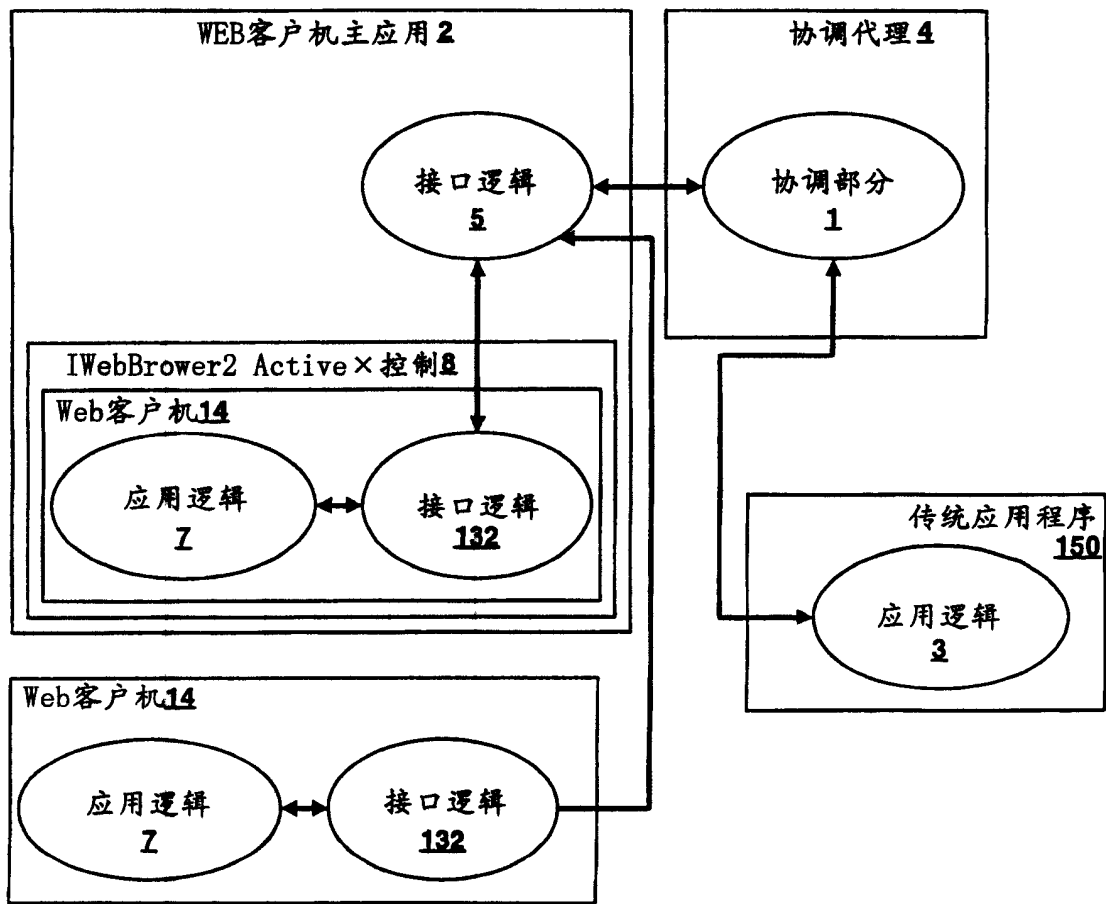


图 2

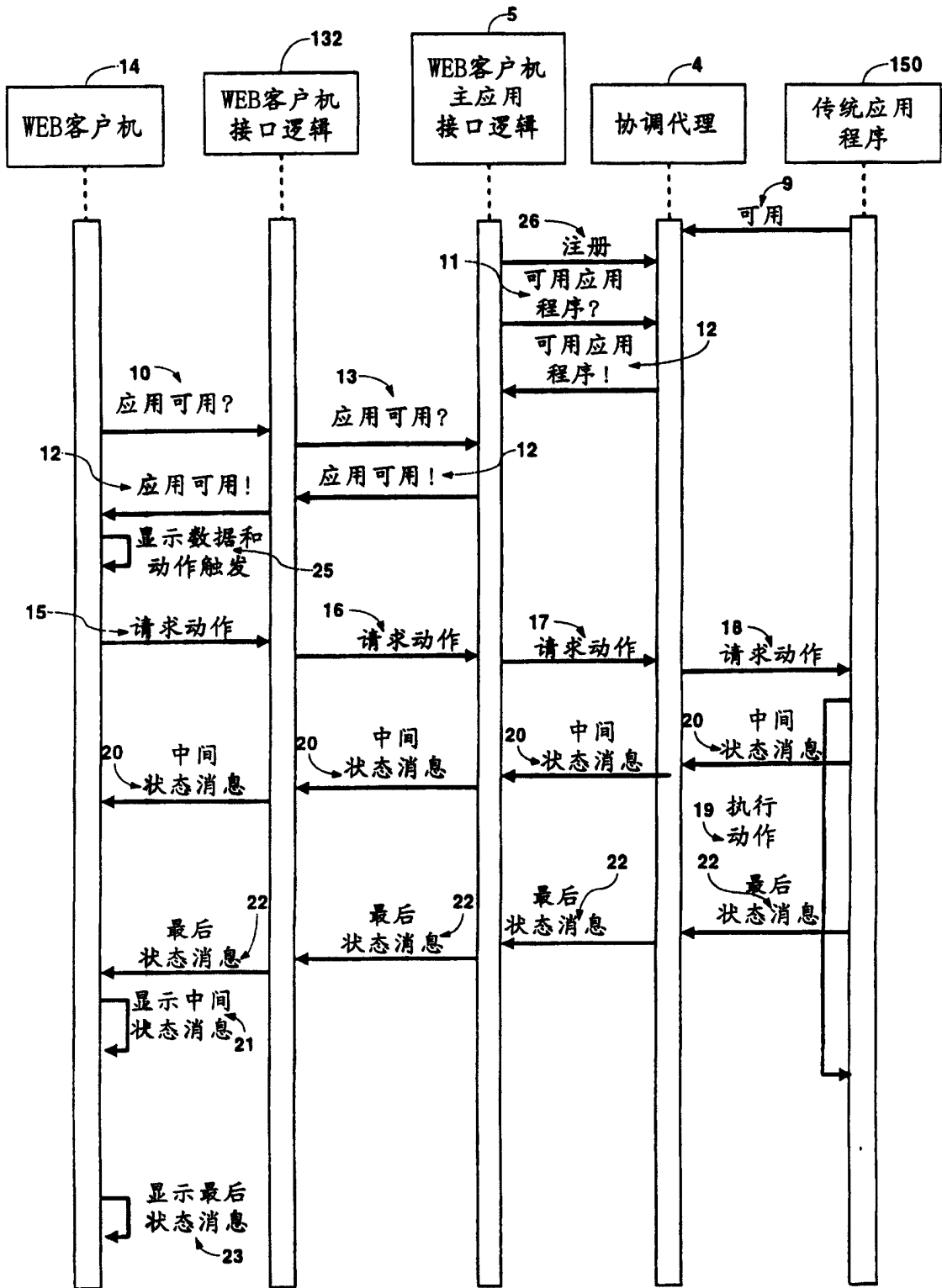


图 3

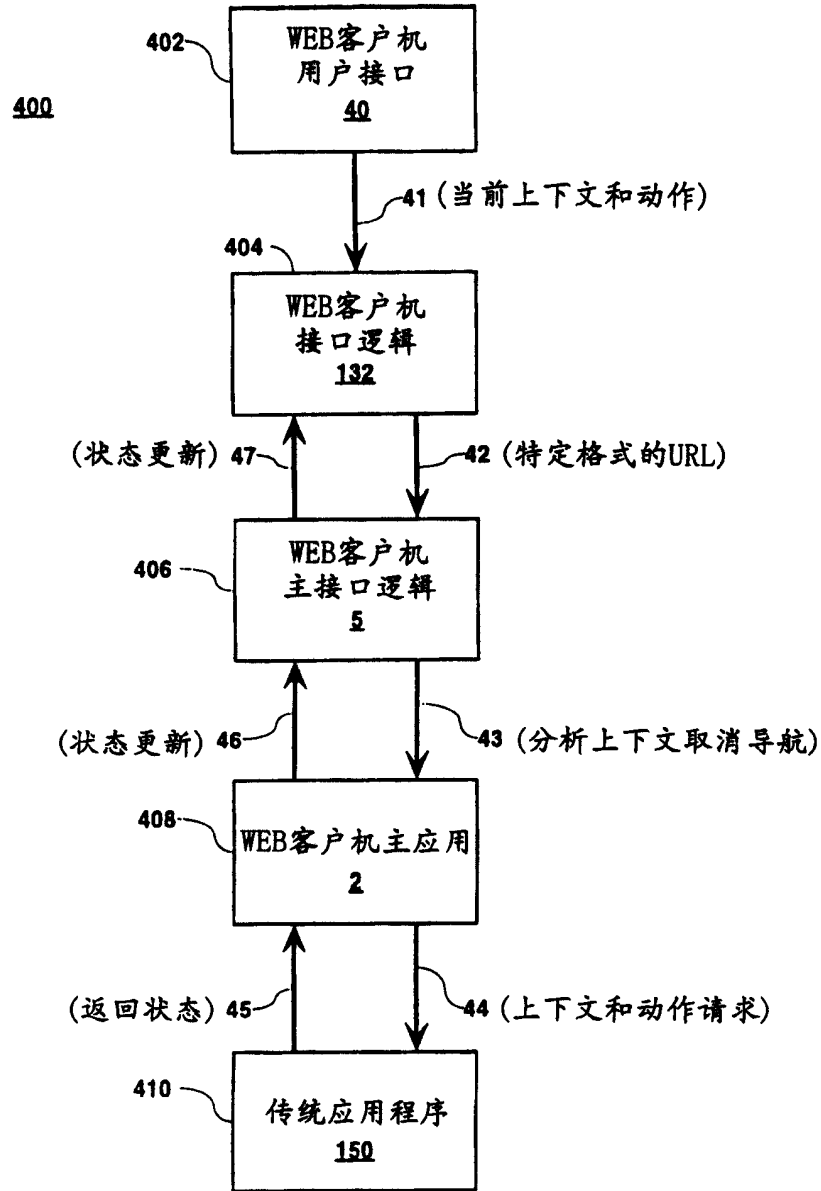


图 4