[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 29/06 (2006.01)
G06F 17/30 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 200610004274.3

[43] 公开日 2006 年 9 月 20 日

[11] 公开号 CN 1835507A

[22] 申请日 2006.2.13
[21] 申请号 200610004274.3
[30] 优先权
[32] 2005.3.17 [33] EP [31] 05102110.3
[71] 申请人 国际商业机器公司
地址 美国纽约
[72] 发明人 扬尼克·赛勒特

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 李镇江

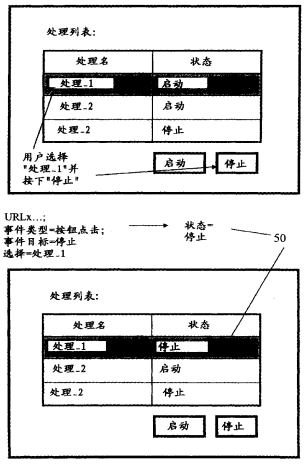
权利要求书 4 页 说明书 11 页 附图 5 页

[54] 发明名称

用于用户与 web 浏览器交互的服务器端处理的方法与系统

[57] 摘要

本发明涉及在基于 web 的客户－服务器环境中用于客户端与服务器端 web 应用程序交互的方法和系统，其中客户端 web 浏览器用作用于在浏览器应用程序框架中显示由 web 应用程序发送的 web 页面并用于将由所述应用程序框架中一个或多个用户输入动作触发的事件转换成指向所述服务器端 web 应用程序的请求的用户界面，其中例如 JavaScript 的可执行程序对象在客户端用作客户端 web 应用程序用户界面的一部分。为了改善用户界面，建议在浏览器中使用与服务器通信并且当服务器发送其响应时防止所显示文档被重新加载的附加框架。相反，只有增量信息(50)显示在该框架中。



1、一种在基于 web 的客户（12）-服务器（14）环境中用于客户端与服务器端 web 应用程序（16）交互的方法，其中客户端 web 浏览器用作用于在浏览器应用程序框架（22）中显示由 web 应用程序发送的 web 页面并用于将由所述应用程序框架（22）中一个或多个用户输入动作触发的事件转换成指向所述服务器端 web 应用程序（16）的请求的用户界面，其中客户－服务器请求－响应通信对话（2、3、4）是利用对所述用户输入动作作出响应的所述服务器端 web 应用程序(16)执行的，其特征在于:

在客户（12）与服务器（14）之间的单个对话步骤中执行步骤:

a）连同由服务器（14）生成的响应参数（50），在对用户隐藏的框架（24）中接收可执行显示编程工具（28），

b）从对用户隐藏的框架（24）中执行所述编程工具（28），其中所述执行在所述应用程序框架（22）预定位置仅部分显示所述响应参数（50）以便对其更新。

2、如权利要求 1 所述的方法，其中用户动作的集合包括利用输入感测可执行程序工具（26）进行以下的步骤:

a）监视所述客户端 web 应用程序用户界面中所述用户输入动作以便检测用户事件，

b）将所述事件的控制特征编码（2）到寻址到控制所述 web 应用程序的服务器端应用程序（16）的相应预定 URL 的扩展中，

c）指示（2）客户端仲裁框架（24）从服务器加载所述编码的 URL，

其中所述通信（2、3、4）包括步骤:

d）向所述服务小程序（16）发送（3）包括所述事件控制特征的请求，

e）接收（4）对所述请求的包括预定服务器响应的回答，该预定服务器响应又包括对应于所述用户触发事件的所述响应参数（50）和

所述第二可执行程序对象（28），及

其中所述显示步骤通过以下执行:

f）调用（5）所述第二可执行显示程序对象（28）,该对象根据web应用程序的程序逻辑修改所述应用程序框架（22）。

3、如权利要求 2 所述的方法，其中所述仲裁框架（24）向服务器（14）发送（3）所述请求。

4、如权利要求 2 所述的方法，其中连同表示期望的服务器响应必须显示在仲裁框架（24）中的附加信息，所述应用程序框架（22）向服务器（14）发送（3）所述请求。

5、一种在基于 web 的客户（12）-服务器（14）环境中用于服务器端与客户端 web 浏览器交互以便执行 web 应用程序（16）的方法，其中客户端 web 浏览器用作用于在浏览器应用程序框架（22）中显示由 web 应用程序发送的 web 页面并用于将由所述应用程序框架（22）中一个或多个用户输入动作触发的事件转换成指向所述服务器端 web 应用程序（16）的请求的用户界面，其中客户－服务器请求－响应通信对话（2、3、4）是利用对所述用户输入动作作出响应的所述服务器端 web 应用程序（16）执行的，

其特征在于步骤:

在客户（12）与服务器（14）之间的单个对话步骤中执行步骤:

a）连同由服务器（14）生成的响应参数（50），发送特定的可执行显示编程工具（28）。

6、如前面任何一项权利要求所述的方法，其中通信包括步骤:

a）从所述客户（12）浏览器的仲裁框架（24）接收（3）包括 URL 和附加 URL 扩展的请求，

b）解码（32）所述扩展，以便得出所述用户动作的事件控制特征，

c）处理（34）对应于所述得出的事件控制特征并基于服务器端 web 应用程序逻辑的事件，

d）生成（36）web 应用程序响应文档，该文档包括:

由服务器（14）生成并代表 web 应用程序逻辑对所述用户事件的响应的响应参数（50），及

专用于 web 浏览器中的执行的可执行显示程序工具（28），

e）将所述响应文档发送（4）回所述辅助编程工具（24）。

7、如权利要求 1 或权利要求 3 所述的方法，其中所述可执行程序工具（26、28）每个都是 JavaScript。

8、一种 web 服务器计算机系统，具有在基于 web 的客户（12）-服务器（14）环境中用于执行服务器端与客户端 web 浏览器的交互以便执行 web 应用程序（16）的工具，其中客户端 web 浏览器用作用于在浏览器应用程序框架（22）中显示由 web 应用程序发送的 web 页面并用于将由所述应用程序框架（22）中一个或多个用户输入动作触发的事件转换成指向所述服务器端 web 应用程序（16）的请求的用户界面，其中客户－服务器请求－响应通信对话（2、3、4）是利用对所述用户输入动作作出响应的所述服务器端 web 应用程序（16）执行的，

其特征在于事件处理功能组件（16）,该组件编程用于:

a）为了提供服务器应用程序响应数据，评价由客户请求发送的请求参数，及

b）选择特定的可执行显示编程工具，及

c）向客户（12）发送回（4）包括所述应用程序响应数据和所述可执行显示编程工具（28）的服务器响应。

9、一种在数据处理系统中执行的计算机程序，用于在基于 web 的客户（12）-服务器（14）环境中执行服务器端与客户端 web 浏览器交互，以便执行 web 应用程序（16），其中客户端 web 浏览器用作用于在浏览器应用程序框架（22）中显示由 web 应用程序发送的 web 页面并用于将由所述应用程序框架（22）中一个或多个用户输入动作触发的事件转换成指向所述服务器端 web 应用程序（16）的请求的用户界面，其中客户－服务器请求－响应通信对话（2、3、4）是利用对所述用户输入动作作出响应的所述服务器端 web 应用程序（16）执行的，

其特征在于用于在客户（12）与服务器（14）之间的单个对话步

骤中执行以下步骤的功能组件:

a）连同由服务器（14）生成的响应参数（50），发送特定的可执行显示编程工具（28）。

10、一种存储在计算机可用介质上的计算机程序产品，有形地体现了如权利要求 9 所述的计算机程序。

用于用户与web浏览器交互的服务器端处理的方法与系统

技术领域

本发明涉及计算机应用程序，尤其涉及在基于web的客户－服务器环境中用于客户端与服务器端web应用程序交互及反过来的方法与相应系统，其中客户端web浏览器用作应用程序用户界面。

背景技术

在这种环境中，传统上web浏览器用于在浏览器应用程序框架中显示由web应用程序发送的web页面及用于将由所述应用程序框架中一个或多个用户动作触发的事件转换成指向所述服务器端web应用程序的请求。由于与独立应用程序相比浏览器的功能很小，因此与由不具有互连web连接的独立应用程序运行的用户界面相比，这种web用户界面可以看作是“轻”版本。

参考图 1,涉及具有低级功能用户界面的作为“轻”客户端的 web浏览器12和存放应用程序的应用程序服务器14的现有技术web应用程序要忍受与用户交互的缺乏，还要忍受不同的编程模型。一般来说，现有技术web应用程序如下工作：根据用户动作1,例如点击链接、按下按钮、按下某个键或激活页面上的任何交互性组件，通过发送 - 见标号2 - 指向URL的HTTP请求，web浏览器12向服务器发出请求。

很可能一组要由服务器14解释的参数可以与该HTTP请求一起传递。这些参数可以包含关于用户在页面上所进行交互的信息，而且可以作为HTTP GET请求中URL编码参数或作为HTTP POST请求体中的数据传递到服务器。

这种请求由生成HTML文档并将其发送回web浏览器的服务小程序（servlet）16或JSP处理 – 见标号3。

服务小程序 16 生成的 HTML 文档一般包含链接到其它 URL 的链接或交互式性公式化元素，或者是链接到不同的服务小程序/JSP，或者是链接到具有不同参数的相同服务小程序 16。

当用户点击这些交互性元素中的一个时，浏览器就重定向到新的 URL。

新 HTTP 请求发送到服务器，该服务器生成完整的新 HTML 文档并将其发送回浏览器。新文档代替旧文档，由循环 1、2、3,通过这种处理用户输入来运行应用程序工作流。

但是，对于这种通用的现有技术方法有一些缺点:

首先，由于每次用户交互都重新加载整个文档，因此在用户交互与来自应用程序的可视响应之间有长的时间延迟: 整个文档必须通过网络发送，它必须由浏览器再次显示。因此，不可能建立有大量用户事件（例如，鼠标事件）由服务器 14 处理的高度交互性用户界面。

其次，实现这种应用程序的编程模型是复杂的。由于每次用户交互都导致新文档对当前文档的替换，因此 web 应用程序的逻辑变得复杂，因为为了在新文档中使用，关于当前会话的当前状态信息必须传递到服务器。

第三，web 应用程序的编程模型完全不同于独立应用程序的编程模型:

在独立的离线应用程序中，通过侦听来自面板元素的事件并利用相同面板其它元素中的属性改变来响应（例如: 当按钮 A 被点击时，文本 B 的颜色改变），用户界面逻辑可以在单个面板中实现。

在 web 应用程序中，每个用户触发的事件都与新文档的生成关联。

这使得没有用户界面的完全重新构造，独立应用程序就不可能移植到 web 应用程序。

发现这种缺点的原因是客户端的技术限制:

web 浏览器最初设计成显示静态内容（文本和图像）。因此，它们提供非常有限的编程能力。尽管已经开发出了许多插件来增强 web

浏览器的能力，但浏览器基本上还是只能向 web 服务器发送请求并显示返回文档的简单客户端软件。

为了弥补 web 浏览器这种能力的缺乏，到目前为止，业界已经开发出了几种特定的现有技术解决方案，包括可执行程序对象，例如作为客户端 web 应用程序用户界面一部分用在客户端的 Java Script。这种可执行代码的例子是:

首先要提到的是 JavaScript（或 ECMAScript）和动态 HTML（DHTML）：利用这种现有技术，脚本语言可以用于修改所显示文档的内容。脚本必须在由 web 浏览器加载的文档中定义。脚本的执行是由浏览器本身进行的。因此，服务器准备脚本，发送到浏览器，但只要文档一显示在浏览器上，服务器就不再对其有任何影响。

其次，Java 应用小程序：利用这种特定的现有技术，完整的 Java 程序被下载到客户端并在浏览器中执行。

但是，这些方法有显著的缺点:

JavaScript 允许修改所显示文档的“不工作”元素，而不需要重新加载整个文档，但实现这种用户界面行为的脚本必须在从服务器下载的原始文档中完全定义。一旦文档被加载，服务器就不再与用户界面交互，逻辑也不能被修改，直到加载新的文档。

后面的 Java 应用小程序是非常灵活的，可以打开与服务器另外的通信通道，但不利的是所要执行的应用程序的全部代码都必须下载到客户端并由浏览器执行。这还涉一有些显著的问题，例如由于所需的代码下载时间、危险的安全问题，还有由于与浏览器所使用的 Java 虚拟机的兼容性。

再次参考图 1,如果用户交互 1 可以本地处理 – 如果对这种交互的反应已经在与 HTML 文档一起发送的 JavaScript 中定义–，则静态嵌入在页面中的脚本可以处理它，而且处理就在这里停止。但由于上述原因，这不是一种令人满意的解决方案。

由于这些缺点，web 应用程序仍旧限定到具有低交互性的组件，即，它们通常在它们的用户界面中包括文本域、按钮、链接。例如，

利用如元素拖放的复杂鼠标交互的应用程序就很难实现。

发明内容

因此，本发明的一个目的是减轻现有技术的缺点。

本发明的这个目的是通过所包含独立权利要求中所述的特征实现的。本发明的更多有利布置和实施方式在相应的子权利要求中阐述。现在参考所附权利要求。

本发明的基本思想包括利用浏览器中与服务器通信并且当服务器发送响应时防止所显示文档被重新加载的附加框架的方法。

根据这种基本方法，公开了在基于 web 的客户－服务器环境中用于客户端与服务器端 web 应用程序交互的方法与相应系统，其中客户端 web 浏览器用作用于在浏览器应用程序框架中显示由 web 应用程序发送的 web 页面并用于将由所述应用程序框架中一个或多个用户输入动作触发的事件转换成指向所述服务器端 web 应用程序的请求的用户界面，其中客户－服务器请求－响应通信对话是利用对所述用户输入动作作出响应的所述服务器端 web 应用程序（16）逐步执行的，该方法特征在于：

在客户端与服务器之间的单个对话步骤中执行步骤：

a）连同由服务器生成的响应参数，在对用户隐藏的框架 – 下文中称为“仲裁框架”- 中接收可执行显示编程工具，

b）从对用户隐藏的框架中执行所述显示编程工具，其中所述执行在所述应用程序框架预定位置仅部分显示用于更新它的所述响应参数。

如下面所提到的，用于显示目的的“第二可执行编程工具 28”或用于输入感测目的的“第一可执行编程工具 26”的可执行程序对象可以是例如 Java Script 或 Java 应用小程序，或 VB-Script（由 MS-Internet Explorer 支持的 Visual Basic 脚本），或者任何其它合适的程序对象。

有利地，客户端“对用户隐藏的”仲裁框架是具有以下属性的框架：

它对用户完全不可视或至少看到的非常小，使其不会吸引用户的注意力。

它不对用户显示任何可视信息；它应当优选地是完全隐藏的。

代替现有技术中通常的应用程序框架本身，它向服务器发送由应用程序框架准备的 HTTP 请求并接收服务器对这种请求的回答。

它执行服务器的结果并只修改应用程序框架的内容，而不改变其余的框架图像。

在这里这种框架称为仲裁框架，因为它实现了应用程序框架与服务器之间的通信。它将应用程序框架的请求发送到服务器，接收并解释服务器的回答：因此，它在应用程序框架与服务器之间的所有通信中充当仲裁者。

仲裁框架应当有利地是对用户隐藏的，以便将该发明性方法无缝集成到通常使用的用户界面中。

本发明公开了允许捕捉 web 浏览器上的用户事件，即按键和鼠标事件、将这些事件发送到服务器、使服务器解释这些事件并估计发送回浏览器的响应的方法与相应系统。响应可以修改显示在浏览器上的一部分文档，而不需要整个文档的重新加载。

这种方法的优点是：

它允许从服务器对 UI 事件细粒度的响应，而不需要在每个事件之间重新加载文档。由于只有对文档所进行的修改而不是整个文档发送到客户端，因此客户端与服务器之间的数据交换是紧凑的，这导致更好的性能并允许更多的客户/服务器交互。这就是根据这里的措词“内容集中”通信所指的。

由此，web 应用程序的逻辑留在服务器，这节省了网络流量并提高了客户端应用程序的安全性。只有可视信息发送到浏览器。而且这还是防止盗版的一步。

web 应用程序运行在服务器上。浏览器只用于显示用户界面。因此，服务器的资源可以有效运行。

该发明性方法可以与通常的浏览器一起使用，而不需要安装特定

的插件。由于它是利用 Java 应用小程序和不同的 Java 版本给出的，因此没有不兼容的问题。

附图说明

本发明是作为例子说明的，而且不受图中图形的形状限制，其中：

图 1 是显示其基本结构和功能元素的示意性现有技术系统图，

图 2 是显示根据其优选实施方式基本结构和功能元素的示意性发明性系统图，

图 3 是说明根据本发明优选实施方式应用程序框架（上）、仲裁框架（中）和服务器逻辑（下）基本逻辑元素及相应界面的示意图，

图 4 是说明在本发明优选实施方式中运行的发明性方法的示意性控制流程图，

图 5 是说明服务器与客户端之间内容集中通信的示例示意性应用程序框架表示。

具体实施方式

现在总体上参考附图并特定参考图 2,有两种不同的框架用在浏览器:

首先，应用程序框架 22 用于显示 web 应用程序并与用户交互。显示在这种框架中的文档应当是允许捕捉用户交互并能利用 JavaScript 动态修改的文档，优选地或一般地，将使用实现文档对象模型级别 2（DOM2）规范的文档。例如 DHTML 文档或可伸缩矢量图形 - SVG 或 XUL（用户界面语言，用在 Mozilla 浏览器中），或任何基于 XML 的文档都可以可选地使用。

应当注意，选择 DOM2 是因为它代表由 W3C 联盟推荐的平台和语言不确定的界面，它使得程序和脚本能够动态访问并更新文档的内容和结构。

其次，根据这种优选实施方式，仲裁框架 24 用于与应用服务器 14 的服务小程序 16 通信。仲裁框架 24 应当保持隐藏，例如通过设置

其宽/高为零。根据本发明，为了保持用户界面简单，有利地假设用户不会与它交互，甚至不会注意到它存在。用户只与应用程序框架交互。因此，与所述仲裁框架相反，在该发明性环境下，由于在这种框架 22 中任何内容都显示给用户，因此应用程序框架 22 还可以看作是“显示”框架。

显示在应用程序框架 22 中的文档使用最小化 JavaScript 26(“第一可执行输入感测编程工具”)，JavaScript 26 检测并感测用户交互，见箭头 2,将它们编码到指向控制该应用程序的服务小程序的 URL 中并调用（再次见箭头 2）仲裁框架 24 发送-箭头 3-带该 URL 的请求。仲裁框架从服务器得到包括可执行程序方法的回答，例如由 JSCRIPT-MF（用于仲裁框架的 JavaScript）表示的 Java Script 28-见箭头 4-并执行它。通过这种执行，在服务器上估计出的新内容显示在应用程序框架，见箭头 5。因此，应用程序的工作流不需要加载新 HTML 文档就可以进行。只有用户要看的真正的相关新信息才在应用程序框架 22 中更新。

应当指出，调用仲裁框架 24 发送带以上 URL 的请求的上述步骤的可选方案是应用程序框架将该请求发送到服务器，而且该请求指示回答应当发送到仲裁框架。特别地，如果所选实现使用 HTTP-POST 与服务器通信，情况更是这样:

利用 HTTP-GET 请求（数据限制到 4Kb），流程将是如下:

首先，将参数编码到 URL 中，然后“请求”仲裁框架加载作为这个 URL 的文档。

其次，利用 HTTP-POST 请求（没有大小限制），应用程序框架一般将准备带隐藏公式的请求并通过指示响应应当转到仲裁框架来自己发送该请求。

在两种情况下，因为逻辑在每次交互时都要重新加载，所以仲裁框架应当包含尽可能少的逻辑，因此请求都是由应用程序框架自己准备的。两种方法的唯一区别是请求如何发送到服务器。

图 3 描述了应用程序框架 22 和仲裁框架 24 的结构细节。仲裁框

架的主要任务是向服务器发送请求并避免用户在其上工作的文档被替换或重新加载。应用程序框架提供准备文档的逻辑。与本发明相关的两种框架的逻辑块如下协作:

应用程序框架 22 包括感测并收集任何事件的感测块 32，即由用户通过按键或鼠标输入使用框架控制完成的事件。事件特征包括事件类型、鼠标坐标、按钮按下等。还提供了将所述事件特征变换成基于文本的按键值对的变换逻辑 34，该按键值对可以存储在 HTTP 请求中，如:

事件类型=鼠标按下，屏幕 X = 100，屏幕 Y = 150，鼠标按钮 = 1

根据本发明的优选特征，由于它不像现有技术那样工作，因此这种变换独立于页面内容，如下面接着示意性示出的:

现有技术 web 应用程序逻辑 - 典型的:

“如果点击按钮 1,则转到 URL XY”

发明性 web 应用程序逻辑 - 典型的:

“如果点击任何按钮，则将按键对值“事件类型 = 按钮点击，事件目标 = %ID_of_the_button，_where_the_event_occurred”作为参数放到要发送的请求中。这可以独立地从当前被点击的按钮应用。

因此，一个文本参数列表从应用程序框架 22 传递到仲裁框架 24，该列表精确地反映用户输入事件。

框架 24 生成包括这些文本参数的 HTTP 请求，并通过 HTTP 将其发送到其中的 URL。

应当指出，检测事件并利用事件参数编码 URL 所需的任何逻辑的实现不是在仲裁框架 24 而是在应用程序框架 22 中实现的。以这种方式，进行这种检测和编码的脚本不需要每次有事件发生时就重新加载，因为仲裁框架中的内容和脚本在每个 HTTP 请求之后都会被完全替换。

在图中，这表示为块 32 和 34 示为在应用程序框架 22 中，而 38、33、35 位于仲裁框架 24 中。

在 web 应用服务器 14（图 3 的底部），请求在块 40 接收，服务

器端参数评价逻辑在解析块 42 解析并提取参数值，并在应用程序逻辑块 44 从中得出用户输入。然后，服务器应用程序逻辑 44 处理这个用户输入并估计其响应。

提供脚本生成块 46。它生成包含可执行代码的脚本，当在块 35 中执行时，该脚本在框架 22 中实现 web 应用程序响应，而不需要从 web 服务器 14 再次加载整个框架。框架 22 新显示的部分充当更新后的框架，以便由用户注意，用户现在可以进行更多输入，该输入再次由感测逻辑 32 感测。因此，循环可以重新进入，而图 3 中所述的所有逻辑块可以再次以类似的方式使用。

另外参考图 2、4 和 5，如下描述发明性方法中的控制流:

步骤 1: 用户通过产生鼠标或按键事件与应用程序框架交互。

参考图 5 的顶部，假设 web 应用程序将一系列处理及其相应状态（启动、停止）显示在表中，并允许用户通过在表中选择它并按下表下边的两个按钮启动和停止中的一个来启动或停止处理。假设用户选择了表中的处理“处理_1”并点击按钮“停止”。

步骤 2: 一般 JavaScript 26（图 2），即前面提到的由图 3 中逻辑块 32 和 34 表示的“第一可执行输入感测编程工具 26”，嵌入到应用程序框架 22 中，收集上表中的鼠标交互并在最终停止按钮被鼠标左键点击后注册它。这种用户动作的整体理解为事件。应用程序框架 22 的所述 JavaScript 26 检测这种事件、提取其特征（即，类型、目标、按键按下等...）、将这些特征作为事件特定的按键值对编码到指向服务小程序的预定 URL 中，并命令仲裁框架 24 加载对应于这个 URL 的文档。

除了上面提到的可选方案，对于这一步，还增加以下可选方案:

代替命令仲裁框架 24 将请求发送到服务器，应用程序框架 22 可以自己发送这种请求，但要带有响应必须显示在仲裁框架 24 中的附加信息。这可以例如通过使用应用程序框架中将 HTTP POST 请求发送到服务器的隐藏公式来实现。HTML/JavaScript 语言使得有可能指定由服务器返回的文档必须显示在不同的框架中。

这里的关键点是响应不预定替换发生交互的文档，但总应当显示

在隐藏的仲裁框架中。

步骤 3：仲裁框架 24 将 HTTP 请求发送到服务小程序 16，其中这种请求包含关于在应用程序框架上检测到的用户交互的信息。即，存储为按键值对（事件类型是“按钮点击”，事件目标是按钮“停止”，表选择是“处理_1”）的特征作为参数在这种请求中传输。这意味着应当理解为是内容集中的，因为请求基本上只包含以上内容。

步骤 4：在所述 web 应用服务器上的服务小程序 16 读出在指定 URL 上进入的信息、解码这种信息并根据期望的 web 应用程序逻辑估计要返回的响应。在这种情况下，应用程序处理可以假设为包括企业数据库访问，及为了对这种用户请求生成一些响应数据，还包括一些数据处理。而且，这也可以不同地实现，例如如果相应企业工作流合适，则通过预存大量 JavaScript 和相应工作流响应数据。

可选地和依赖于程序，响应数据可以在运行时估计。服务器响应同样是内容集中的，因为服务器响应中传输的参数基本上包含非 GUI 图形的内容，在这里所估计出的响应将是在表中对应于“处理_1”的状态文本从“启动”变成“停止”。这可以是包含当文档被浏览器加载时要由浏览器执行的 JavaScript 28 的 HTML 文档。脚本包含内容。由于它可以被技术人员理解，因此这种内容只是用于更新应用程序框架 22 的增量信息，而不需要从服务器完全加载它。

步骤 5：一旦响应文档被仲裁框架 24 接收，就执行其中嵌入了 JavaScript－MF 28 的 JavaScript，即前面提到的“第一可执行显示编程工具 28”。通过增加新元素、除去现有元素或修改现有元素的属性，这种脚本修改应用程序框架中的文档。应用程序框架 22 中所显示文档的其余部分不需要重新加载。

本发明可以在硬件、软件或硬件与软件的组合中实现。根据本发明的 web 应用程序可以在一个计算机系统中以集中方式实现，或以不同元素散布在几个互连计算机系统上的分布方式实现。任何类型适于执行在此所述方法的计算机系统或其它装置都适合。硬件与软件的一般组合可以是具有计算机程序的通用计算机系统，当加载并执行时，

该计算机程序控制计算机系统使其执行在此所述的方法。

本发明还可以嵌入在计算机程序产品中，该产品包括使得能够实现在此所述方法的所有特征，而且当加载到计算机系统中时，该产品能够执行这些方法。

本环境中的计算机程序方法或计算机程序意味着一组指令的任何语言、代码或符号的任何表达，该指令要使具有信息处理能力的系统直接或在以下一个或两个之后执行特定功能：

a）转换成其它语言、代码或符号；

b）在不同的材料形式中再现。

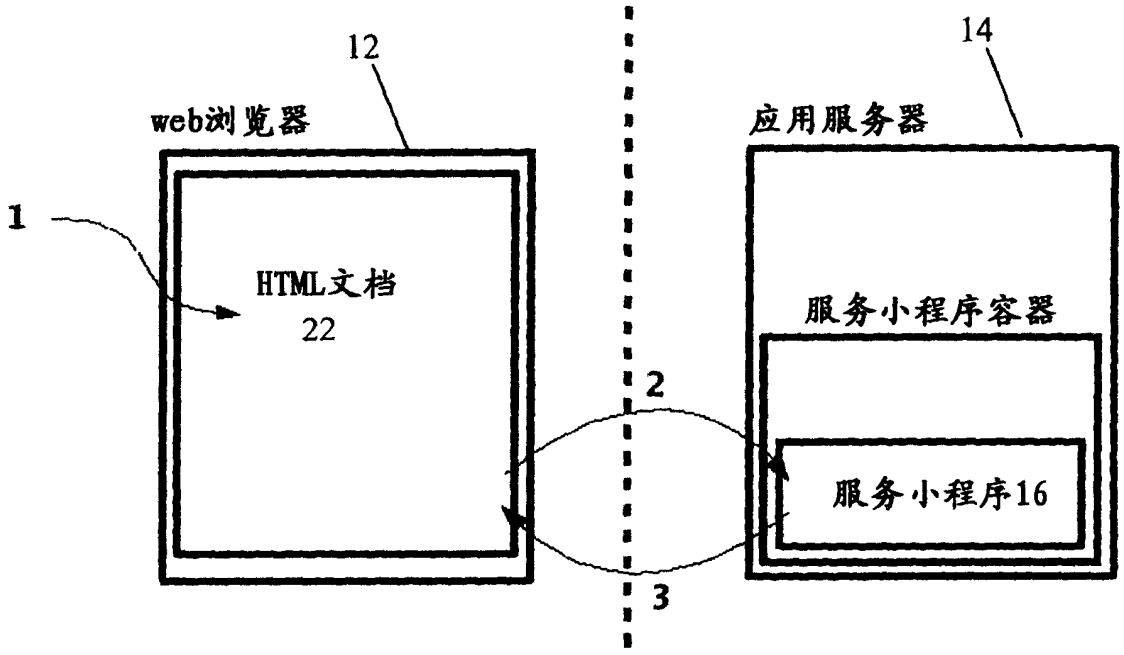


图1
现有技术

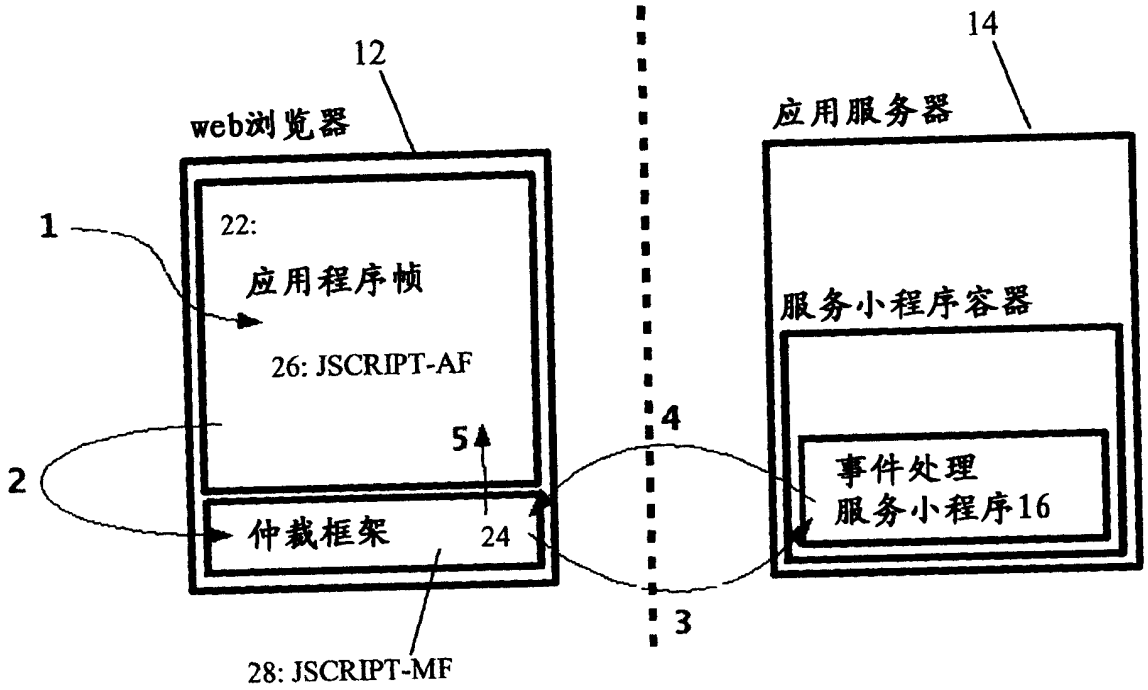

12
web浏览器
14
应用服务器
1
22:
应用程序帧
26: JSCRIPT-AF
服务小程序容器
5
4
2
事件处理
服务小程序16
仲裁框架
24
3
28: JSCRIPT-MF

图2

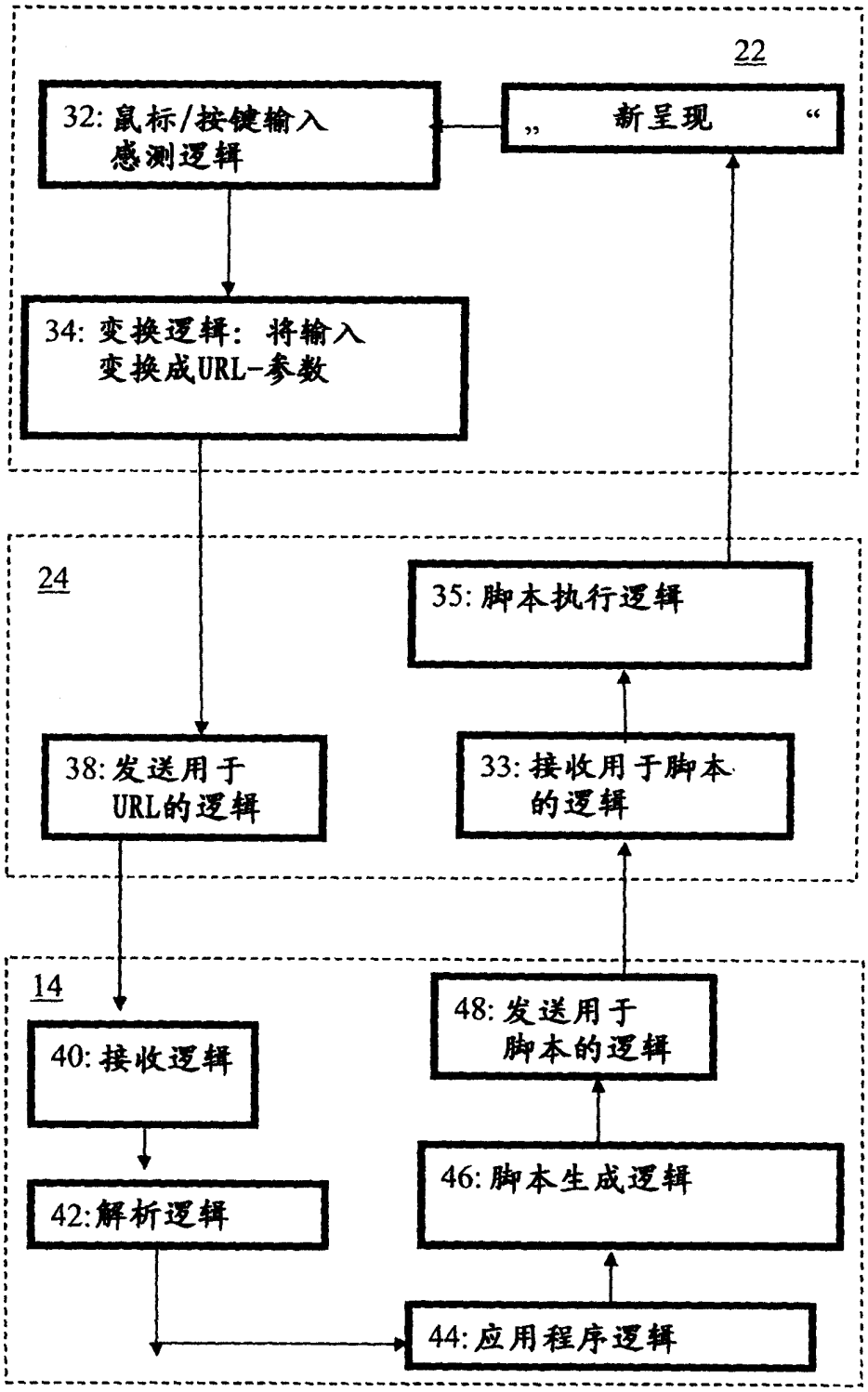

22
32: 鼠标/按键输入感测逻辑
„ 新呈现 “
34: 变换逻辑：将输入变换成URL-参数
24
35: 脚本执行逻辑
38: 发送用于URL的逻辑
33: 接收用于脚本的逻辑
14
48: 发送用于脚本的逻辑
40: 接收逻辑
46: 脚本生成逻辑
42: 解析逻辑
44: 应用程序逻辑

图 3

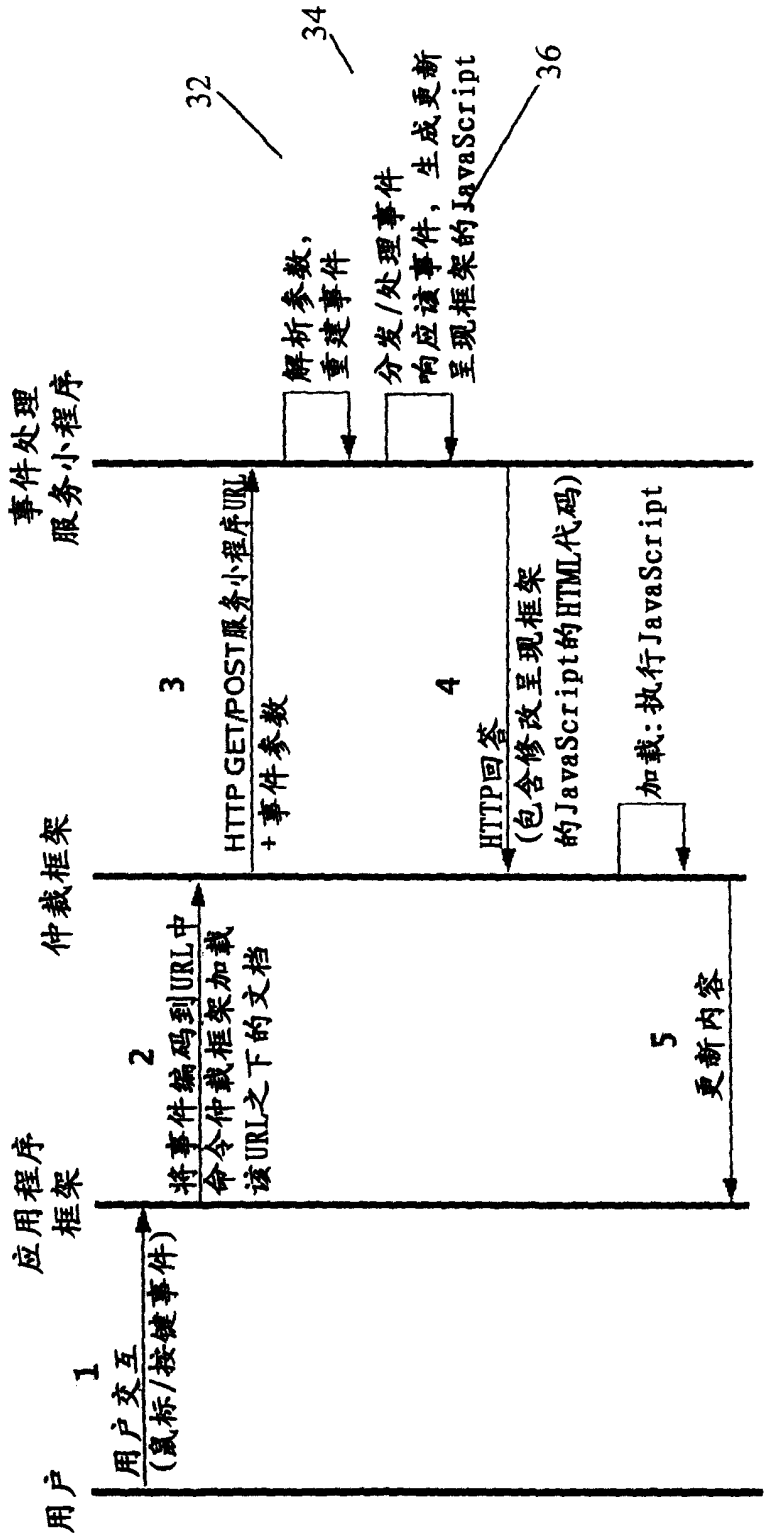

用户
应用程序框架
仲裁框架
事件处理服务小程序
1
用户交互
(鼠标/按键事件)
2
将事件编码到URL中
命令仲裁框架加载
该URL之下的文档
3
HTTP GET/POST服务小程序URL
+事件参数
解析参数,
重建事件
32
34
分发/处理事件
响应该事件,生成更新
呈现框架的JavaScript
36
4
HTTP回答
(包含修改呈现框架
的JavaScript的HTML代码)
加载:执行JavaScript
5
更新内容

图 4

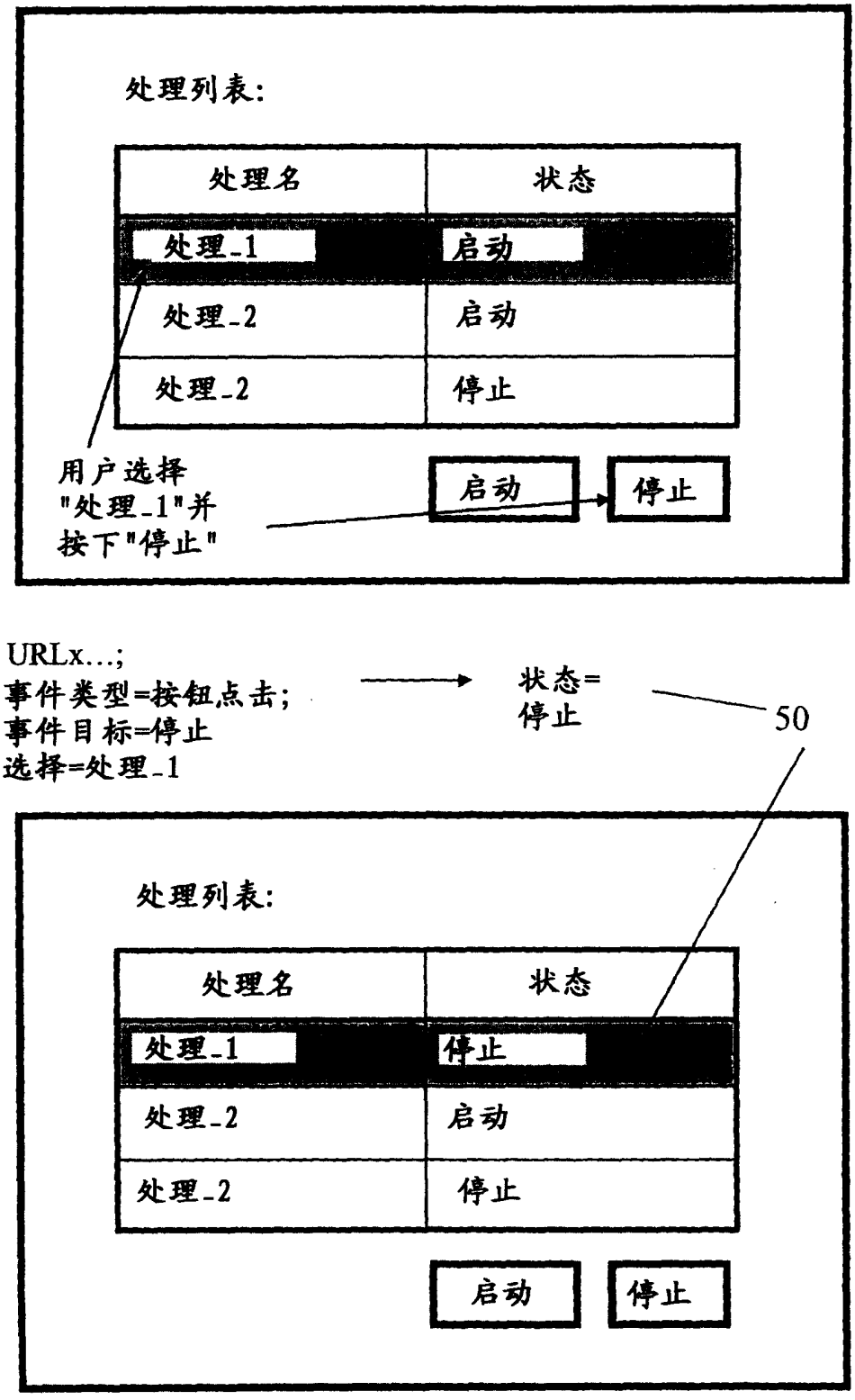

处理列表:
处理名
状态
处理_1
启动
处理_2
启动
处理_2
停止
用户选择
"处理_1"并
按下"停止"
启动
停止
URLx...;
事件类型=按钮点击;
事件目标=停止
选择=处理_1
状态=
停止
50
处理列表:
处理名
状态
处理_1
停止
处理_2
启动
处理_2
停止
启动
停止

图5